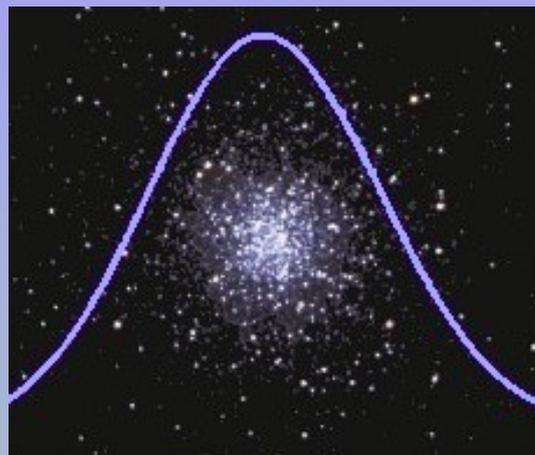


*The 4th IIA-PennState Astrostatistics School*  
*22<sup>nd</sup> - 29<sup>th</sup> July, 2013*



*Vainu Bappu Observatory,  
Indian Institute of Astrophysics, Kavalur*



*Lecture Notes*





## **Co-ordinators**

*Prajval Shastri (IIA)*

*Sabyasachi Chatterjee (IIA)*

*Jogesh Babu (Pennsylvania State University)*

## **Resource Persons**

*Jogesh Babu (Pennsylvania State University)*

*Sushama Bendre ((Indian Statistical Institute, Tezpur )*

*Arnab Chakraborty (Indian Statistical Institute, Kolkata)*

*Thriyambakam Krishnan (Mu Sigma Business Solutions, Bangalore)*

*Bhamidi V Rao (Chennai Mathematical Institute, Chennai)*

*Rahul Roy (Indian Statistical Institute, New Delhi)*

*Deepayan Sarkar (Indian Statistical Institute, New Delhi)*

## **Organsing Committee**

*Harish Bhatt*

*Sabyasachi Chatterjee*

*Mousumi Das*

*Preeti Kharb*

*Anbazhagan Poobalan*

*Prajval Shastri (IIA)*

*Arun Surya*

*Sivarani Thirupathi*

### **Credits:**

*Front Cover photograph (also used in this frontispiece & webpages): A. Ramachandran, Anbazhagan Poobalan.*

*Back Cover: (Top) Image of NGC 4590 observed with the Vainu Bappu Telescope in February, 2001 by S. Ambika & the VBT team. (Bottom) Gauss with a telescope, at the Goettingen Observatory. His studies on the positions of celestial objects were of seminal importance in such fundamental discoveries like the normal distribution of errors and the method of least squares (Image scanned by the American Institute of Physics). Back cover produced by Firoza Sutaria.*

**Acknowledgements:** *We would like to thank Dr P. Kumaresan, Raja Iyengar, S. Rajendran, S.B. Ramesh, S. Dhananjaya, S. Fayaz, Yogesh Jogi, Estrella Jimenez, John Hodgson and all the volunteers and staff of the Vainu Bappu Observatory.*



# Contents

<b>1</b>	<b>PROBABILITY</b> <i>Notes by Rahul Roy, Bhamidi V Rao &amp; Rajeeva Karandikar</i>	<b>1</b>
<b>2</b>	<b>INTRODUCTION TO R</b> <i>Notes by Arnab Chakraborty</i>	<b>19</b>
<b>3</b>	<b>DESCRIPTIVE STATISTICS &amp; GRAPHING WITH R</b> <i>Notes by Arnab Chakraborty</i>	<b>35</b>
<b>4</b>	<b>ESTIMATION, CONFIDENCE INTERVALS &amp; HYPOTHESIS TESTING</b> <i>Notes by Donald Richards &amp; Bhamidi V Rao</i>	<b>49</b>
<b>5</b>	<b>CORRELATION &amp; REGRESSION</b> <i>Notes by Rajeeva Karandikar</i>	<b>65</b>
<b>6</b>	<b>REGRESSION WITH R</b> <i>Notes by Arnab Chakraborty</i>	<b>83</b>
<b>7</b>	<b>MAXIMUM LIKELIHOOD ESTIMATION</b> <i>Notes by Donald Richards &amp; Bhamidi V Rao</i>	<b>97</b>
<b>8</b>	<b>TESTING AND ESTIMATION IN R</b> <i>Notes by Arnab Chakraborty</i>	<b>111</b>
<b>9</b>	<b>TRUNCATION &amp; CENSORING</b> <i>Notes by Jogesh Babu</i>	<b>129</b>
<b>10</b>	<b>NON-PARAMETRIC STATISTICS</b> <i>Notes by Sushama Bendre</i>	<b>135</b>
<b>11</b>	<b>NON-PARAMETRIC STATISTICS WITH R</b> <i>Notes by Arnab Chakraborty</i>	<b>161</b>
<b>12</b>	<b>ANALYSIS OF DATACUBES</b> <b>NOTES BY JOGESH BABU</b>	<b>173</b>
<b>13</b>	<b>BAYESIAN ANALYSIS</b>	

	<i>Notes by Mohan Delampady</i>	<b>187</b>
<b>14</b>	BAYESIAN ANALYSIS <i>Notes by Tom Loredo</i>	<b>221</b>
<b>15</b>	JACKKNIFE & BOOTSTRAP <i>Notes by Jogesh Babu</i>	<b>253</b>
<b>16</b>	MODEL SELECTION & EVALUATION, GOODNESS-OF-FIT <i>Notes by Rajeeva Karandikar &amp; Jogesh Babu</i>	<b>265</b>
<b>17</b>	SIMULATION AND BOOTSTRAPPING WITH R <i>Notes by Arnab Chakraborty</i>	<b>279</b>
<b>18</b>	EM ALGORITHM <i>Notes by Thriyambakam Krishnan</i>	<b>293</b>
<b>19</b>	MULTIVARIATE ANALYSIS <i>Notes by Thriyambakam Krishnan</i>	<b>307</b>
<b>20</b>	PRINCIPAL COMPONENT ANALYSIS IN R <i>Notes by Arnab Chakraborty</i>	<b>329</b>
<b>21</b>	CLUSTER ANALYSIS <i>Notes by Thriyambakam Krishnan &amp; Jia Li</i>	<b>337</b>
<b>22</b>	CLUSTER ANALYSIS IN R <i>Notes by Arnab Chakraborty</i>	<b>359</b>
<b>23</b>	MCMC <i>Notes by Arnab Chakraborty</i>	<b>369</b>
<b>24</b>	MCMC <i>Notes by Murali Haran</i>	<b>379</b>
<b>25</b>	MCMC <i>Notes by Tom Loredo</i>	<b>383</b>
<b>26</b>	TIME SERIES ANALYSIS <i>Notes by Arnab Chakraborty</i>	<b>395</b>
<b>27</b>	TIME SERIES ANALYSIS <i>Notes by Eric Feigelson</i>	<b>407</b>
<b>28</b>	APPENDIX A: DISTRIBUTIONS	<b>415</b>
<b>29</b>	APPENDIX B: JARGON	<b>423</b>

# Chapter 1

## PROBABILITY

*Notes by Rahul Roy, Bhamidi V Rao & Rajeeva Karandikar*

## Do Random phenomena exist in Nature?

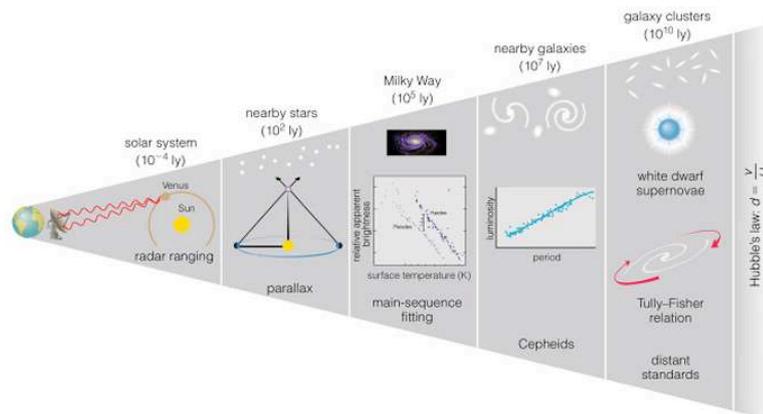
Which way a coin tossed in air will fall may be completely determined by laws of physics. The only problem in figuring out the trajectory and hence the face of the coin when it is on ground is that we have to measure too many parameters, e.g. angular momentum of rotation, force at the time of toss, wind pressure at various instants during the rotation of the coin, etc.!

Which way an electron will spin is also not known and so modelling it will require incorporating a **random** structure. But we cannot exclude the possibility that sometime in future, someone will come up with a theory that will explain the spin.

Thus we often come across events whose outcome is uncertain. The uncertainty could be because of our inability to observe accurately all the inputs required to compute the outcome.

It may be too expensive or even counterproductive to observe all the inputs. The uncertainty could be due to the current level of understanding of the phenomenon. The uncertainty could be on account of the outcome depending on choices made by a group of people at a future time - such as outcome of an election yet to be held.

**Cosmic distance ladder** The objects in the solar system were measured quite accurately by ancient Greeks and Babylonians using geometric and trigonometric methods.



see also [www.math.ucla.edu/~tao/preprints/Slides/Cosmic%20Distance%20Ladder.ppt](http://www.math.ucla.edu/~tao/preprints/Slides/Cosmic%20Distance%20Ladder.ppt)

The distance to the stars in the second lot are found by ideas of parallax, calculating the angular deviation over 6 months. First done by the mathematician

Friedrich Bessel. The error here is of the order of 100 light years.

---

The distances of the moderately far stars are obtained by a combination of their apparent brightness and the distance to the nearby stars. This method works for stars upto 300,000 light years and the error is significantly more.

The distance to the next and final lot of stars is obtained by plotting the oscillations of their brightness. This method works for stars upto 13,000,000 light years!

---

At every step of the distance ladder, errors and uncertainties creep in. Each step inherits all the problems of the ones below, and also the errors intrinsic to each step tend to get larger for the more distant objects; thus the spectacular precision at the base of the ladder degenerates into much greater uncertainty at the very top.

---

So we need to understand **UNCERTAINTY**.

And the only way of understanding a notion scientifically is to provide a structure to the notion.

A structure rich enough to lend itself to quantification.

---



The structure needed to understand a coin toss is intuitive.

We assign a probability  $1/2$  to the outcome **HEAD** and a probability  $1/2$  to the outcome **TAIL** of appearing.

---



Similarly for each of the outcomes **1,2,3,4,5,6** of the throw of a dice we assign a probability  $1/6$  of appearing.



Similarly for each of the outcomes  $000001, \dots, 999999$  of a lottery ticket we assign a probability  $1/999999$  of being the winning ticket.

Of course, we could obtain the structure of the uncertainty in a coin toss from the example of throwing a dice.

In particular if we declare as **HEAD** when the outcome of a throw of a dice is an even number, and if we declare as **TAIL** when the outcome of a throw of a dice is an odd number, then we have the same structure as that we had from a coin toss.

More generally, associated with any experiment we have an outcome space  $\Omega$  consisting of outcomes  $\{o_1, o_2, \dots, o_m\}$ .

Coin Toss –  $\Omega = \{H, T\}$

Dice –  $\Omega = \{1, 2, 3, 4, 5, 6\}$

Lottery –  $\Omega = \{1, \dots, 999999\}$

Each outcome is assigned a probability

Coin Toss –  $p_H = 1/2, p_T = 1/2$

Dice –  $p_i = 1/6$  for  $i = 1, \dots, 6$

Lottery –  $p_i = 1/999999$  for  $i = 1, \dots, 999999$

More generally, for an experiment with an outcome space  $\Omega = \{o_1, o_2, \dots, o_m\}$ , we assign a probability  $p_i$  to the outcome  $o_i$  for every  $i$  in such a way that the probabilities add up to 1.

The set  $\Omega = \{o_1, o_2, \dots, o_m\}$  is called a sample space.

A subset  $E \subseteq \Omega$  is called an event.

We may be gambling with dice, so we could have a situation like

<i>outcome</i>	1	2	3	4	5	6
<i>money amount</i>	-8	2	0	4	-2	4

Our interest in the outcome is only *vis-á-vis* its association with the monetary amount.

---

So we are interested in a mapping (i.e. a function) of the outcome space  $\Omega$  to the reals  $\mathbb{R}$

Such functions are called random variables.

The probabilistic properties of these random variables can be read out from the probabilities assigned to the outcomes of the underlying outcome space.

---

The probability that you win 4 rupees, i.e.  $P\{X = 4\}$  means you want to find that the number 4 or the number 6 came out on the dice, i.e.  $P\{4, 6\}$ . Thus  $P\{\omega : X(\omega) = 4\} = P\{4, 6\} = (1/6) + (1/6) = 1/3$ .

Similarly the probability that you do not lose any money is the probability of the event that either 2, 3, 4 or 6 came out on the dice, and this probability is  $(1/6) + (1/6) + (1/6) + (1/6) = 2/3$ .

---

### What are we doing?

Recall our assignment of probabilities  $P(o_i) = p_i$  on the outcome space  $\Omega = \{o_1, o_2, \dots, o_m\}$ .

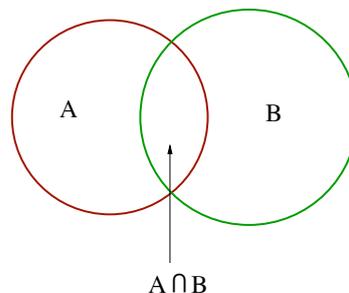
For an event  $E = \{o_{i_1}, o_{i_2}, \dots, o_{i_k}\}$ , we define

$$P(E) = p_{i_1} + p_{i_2} + \dots + p_{i_k}.$$

Easy to check that if  $A, B$  are mutually disjoint, i.e.  $A \cap B = \phi$  then

$$P(A \cup B) = P(A) + P(B)$$


---



More generally, we can check that for any two events  $A, B$

$$P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

Similarly, for three events  $A, B, C$

$$\begin{aligned} P(A \cup B \cup C) &= P(A) + P(B) + P(C) \\ &\quad - P(A \cap B) - P(A \cap C) - P(B \cap C) \\ &\quad + P(A \cap B \cap C) \end{aligned}$$

This has a generalization to  $n$  events.

How do we assign the probabilities  $p_i$  to the elementary outcomes?

The simplest case is when due to inherent symmetries present, we can model all the elementary events (*i.e.* outcomes) as being **equally likely**.

vspace2mm

When an experiment results in  $m$  equally likely outcomes  $o_1, o_2, \dots, o_m$ , the probability of an event  $A$  is

$$P(A) = \frac{\#A}{m}$$

*i.e.* the ratio of the number of favourable outcomes to the total number of outcomes.

**Example: Toss a coin three times**

$\Omega = \{HHH,$ $HHT, HTH, THH,$ $HTT, THT, TTH,$ $TTT\}$ $p(***) = 1/8$	<b>No. of Heads in 3 tosses</b> $\Omega = \{0, 1, 2, 3\}$ $\longleftarrow$ 3 Heads $\longleftarrow$ 2 Heads $\longleftarrow$ 1 Head $\longleftarrow$ 0 Heads $p(0) = 1/8, p(1) = 3/8,$ $p(2) = 3/8, p(3) = 1/8$
--	--

Note we could have done the calculations in the red part without even associating it with the blue sample space etc.

**Conditional probability** Let  $X$  be the number which appears on the throw of a dice.

Each of the six outcomes is equally likely, but suppose I take a peek and tell you that  $X$  is an even number.

**Question:** What is the probability that the outcome belongs to  $\{1, 2, 3\}$ ?

---

**Given** the information I conveyed, the six outcomes are no longer equally likely. Instead, the outcome is one of  $\{2, 4, 6\}$  – each being equally likely.

So with the information you have, the probability that the outcome belongs to  $\{1, 2, 3\}$  equals  $1/3$ .

---

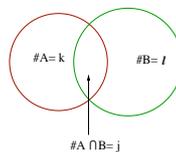
Consider an experiment with  $m$  equally likely outcomes and let  $A$  and  $B$  be two events.

**Given the information that  $B$  has happened**, what is the probability that  $A$  occurs?

This probability is called the **conditional probability of  $A$  given  $B$**

and written as  $P(A | B)$ .

---



Let  $\#A = k$ ,  $\#B = l$ ,  $\#(A \cap B) = j$ .

**Given that  $B$  has happened**, the new probability assignment gives a probability  $1/l$  to each of the outcomes in  $B$ .

---

Out of these  $l$  outcomes of  $B$ ,  $\#(A \cap B) = j$  outcomes also belong to  $A$ .

Hence  $P(A | B) = j/l$ .  
Noting that  $P(A \cap B) = j/m$  and  $P(B) = l/m$ , it follows that

$$P(A | B) = \frac{P(A \cap B)}{P(B)}.$$


---

In general when  $A, B$  are events such that  $P(B) > 0$ , the conditional probability of  $A$  given that  $B$  has occurred  $P(A | B)$  is defined by

$$P(A | B) = \frac{P(A \cap B)}{P(B)}$$

This leads to the **Multiplicative law of probability**

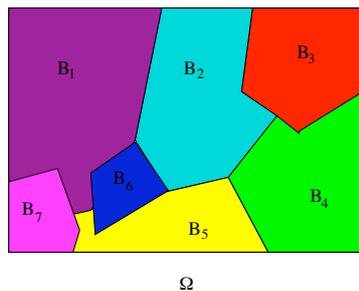
$$P(A \cap B) = P(A | B)P(B)$$

This has a generalization to  $n$  events:

$$\begin{aligned} &P(A_1 \cap A_2 \cap \dots \cap A_n) \\ &= P(A_n | A_1, \dots, A_{n-1}) \\ &\quad \times P(A_{n-1} | A_1, \dots, A_{n-2}) \\ &\quad \times \dots \times \\ &\quad \times P(A_2 | A_1)P(A_1) \end{aligned}$$

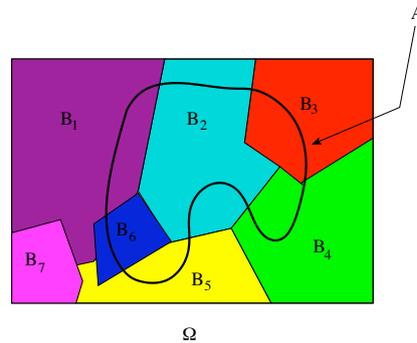
### The Law of Total Probability

Let  $B_1, \dots, B_k$  be a partition of the sample space  $\Omega$



### The Law of Total Probability

Let  $B_1, \dots, B_k$  be a partition of the sample space  $\Omega$  and  $A$  an event



Then

$$P(A) = P(A \cap B_1) + \cdots + P(A \cap B_k)$$

Also we know

$$P(A \cap B_i) = P(A|B_i)P(B_i)$$

so we get the Law of Total Probability

$$P(A) = P(A|B_1)P(B_1) + \cdots + P(A|B_k)P(B_k)$$

### Example

Suppose a bag has 6 one rupee coins, exactly one of which is a **Sholay coin**, i.e. both sides are **HEAD**. A coin is picked at random and tossed 4 times, and each toss yielded a **HEAD**.

Two questions which may be asked here are

- (i) what is the probability of the occurrence of  $A = \{\text{all four tosses yielded HEADS}\}$ ?
- (ii) **given** that  $A$  occurred, what is the probability that the coin picked was the **Sholay coin**?

The first question is easily answered by the laws of total probability. Let

$B_1$  = coin picked was a regular coin

$B_2$  = coin picked was a **Sholay coin**

Then

$$\begin{aligned} P(A) &= P(A | B_1)P(B_1) + P(A | B_2)P(B_2) \\ &= \left(\frac{1}{2}\right)^4 \frac{5}{6} + \frac{1}{6} \\ &= \frac{21}{96} = \frac{7}{32} \end{aligned}$$

For the second question we need to find

$$\begin{aligned}
 P(B_2 | A) &= \frac{P(B_2 \cap A)}{P(A)} \\
 &= \frac{P(A | B_2)P(B_2)}{P(A)} \\
 &= \frac{1/6}{7/32} \\
 &= \frac{16}{21}
 \end{aligned}$$

The previous example is atypical of the situation where we perform scientific experiments and make observations. On the basis of the observations we have to infer what was the theoretical process involved in the experiment to obtain the given observation. Occassionally we may have some (though not complete) information of the process, in which case we can use this information to help in our inference.

In particular, in the example we had the **prior** information that there was exactly one **Sholay coin** among the six coins.

Suppose we have observed that  $A$  occurred.

Let  $B_1, \dots, B_m$  be all possible scenarios under which  $A$  may occur, i.e.  $B_1, \dots, B_m$  is a partition of the sample space. To quantify our suspicion that  $B_i$  was the cause for the occurrence of  $A$ , we would like to obtain  $P(B_i | A)$ .

Bayes' formula or Bayes' theorem is the prescription to obtain this quantity. The theorem is very easy to establish and is the basis of **Bayesian Inference**.

### Bayes' Theorem:

If  $B_1, B_2, \dots, B_m$  is a partition of the sample space, then

$$\begin{aligned}
 P(B_i | A) &= \frac{P(A | B_i)P(B_i)}{P(A)} \\
 &= \frac{P(A | B_i)P(B_i)}{\sum_{j=1}^m P(A | B_j)P(B_j)}
 \end{aligned}$$

Suppose that  $A, B$  are events such that

$$P(A | B) = P(A)P(B).$$

Then we get

$$P(A | B) = P(A).$$

i.e. the knowledge that  $B$  has occurred has not altered the probability of  $A$ .

In this case,  $A$  and  $B$  are said to be **independent** events.

---

Let  $X, Y, Z$  be random variables each taking finitely many values. Then  $X, Y, Z$  are said to be independent if

$$P(X = i, Y = j, Z = k) = P(X = i)P(Y = j)P(Z = k)$$

for all possible values  $i, j, k$  of  $X, Y, Z$  respectively.

This can be generalized to finitely many random variables.

---

## Expectation of a random variable

Let  $X$  be a random variable taking values  $x_1, x_2, \dots, x_n$ . The expected value  $\mu$  of  $X$  (also called the **mean** of  $X$ ) denoted by  $E(X)$  is defined by

$$\mu = E(X) = \sum_{i=1}^n x_i P(X = x_i).$$

The **variance** of a random variable is defined by

$$\sigma^2 = \text{Var}(X) = E\{(X - \mu)^2\}.$$


---

## Example

Let  $X$  be a random variable

taking values

+1 or -1

with prob. 1/2 each

$\mu = E(X) = 0$

and

$\sigma^2 = \text{Var}(X) = 1$

taking values

+10 or -10

with prob. 1/2 each

$\mu = E(X) = 0$

and

$\sigma^2 = \text{Var}(X) = 100$

The variance of a random variable describes the spread of the values taken by the random variable.

---

## Notation

We will denote by CAPITAL letters the random variables, and by small letters the values taken by the random variables.

Thus  $X, Y, Z$  will stand for random variables, while  $x, y, z$  will stand for the values attained by the random variables  $X, Y, Z$  respectively.

---

### Examples of random variables

Consider  $n$  independent trials where the probability of success in each trial is  $p$  and let  $X$  denote the total number of successes, then

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

for  $k = 0, 1, \dots, n$ ,  $0 \leq p \leq 1$ . This is known as Binomial distribution, written as  $X \sim B(n, p)$ .

$$E(X) = np \text{ and } Var(X) = np(1-p).$$


---

Consider a random variable  $X$  such that

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

for  $k = 0, 1, 2, \dots$  and  $\lambda > 0$ . This is known as Poisson distribution. Here  $E(X) = \lambda$  and  $Var(X) = \lambda$ .

---

If  $X$  has Binomial distribution  $B(n, p)$  with large  $n$  and small  $p$ , then  $X$  can be approximated by a Poisson random variable  $Y$  with parameter  $\lambda = np$ , i.e.

$$P(X \leq a) \approx P(Y \leq a)$$


---

In order to consider random variables that may take any real number or any number in an interval as its value, we need to extend our notion of sample space and events. One difficulty is that we can no longer define probabilities for all subsets of the sample space. We will only note here that the class of events - namely the sets for which the probabilities are defined is large enough.

---

We also need to add an axiom called **Countable additivity axiom**: If  $A_1, A_2, \dots, A_k, \dots$  are pairwise mutually exclusive events then

$$P(\cup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i).$$

A real valued function  $X$  on a sample space  $\Omega$  is said to be a random variable if for all real numbers  $a$ , the set  $\{\omega : X(\omega) \leq a\}$  is an event.

For a random variable  $X$ , the function  $F$  defined by

$$F(x) = P(X \leq x)$$

is called its distribution function. If there exists a function  $f$  such that

$$F(x) = \int_{-\infty}^x f(t) dt$$

then  $f$  is called the density of  $X$ .

Examples of densities:

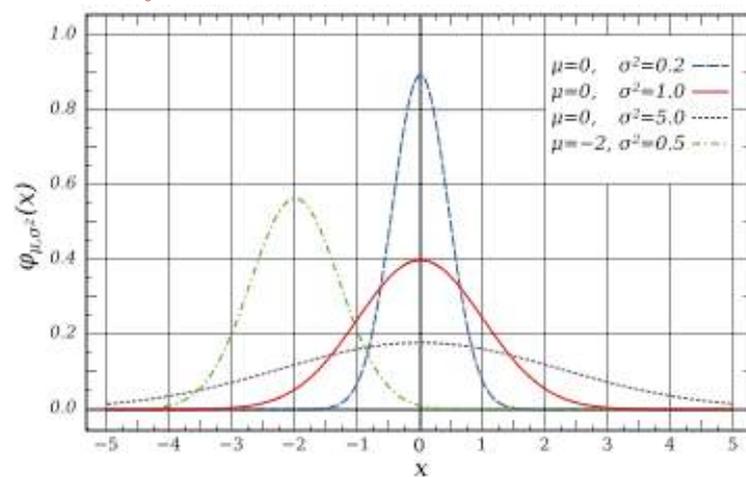
$$f(x) = \begin{cases} \lambda \exp(-\lambda x) & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

This is called exponential density.

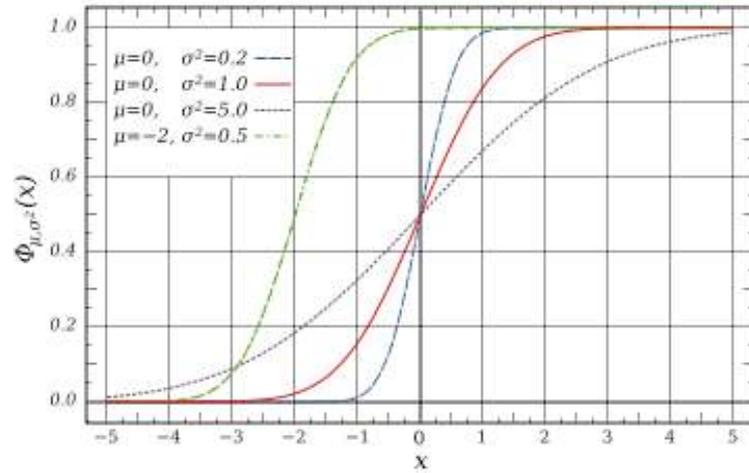
$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

This is the Normal density.

### Normal density function



### Normal density function



A very common density function encountered in astronomy is the globular cluster luminosity function GCLF.

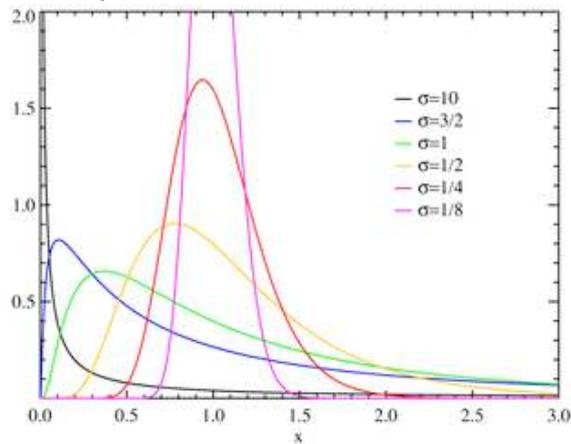
A globular cluster (GC) is a collection of  $10^4$ - $10^6$  ancient stars concentrated into a tight spherical structure structurally distinct from the field population of stars.

The distribution of GC luminosities (i.e. the collective brightness of all of its stars) is known as the globular cluster luminosity function (GCLF).

The shape of this function is said to be **lognormal** i.e.

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left\{-\frac{(\ln(x) - \mu)^2}{2\sigma^2}\right\} \text{ for } x > 0.$$

### Lognormal density function



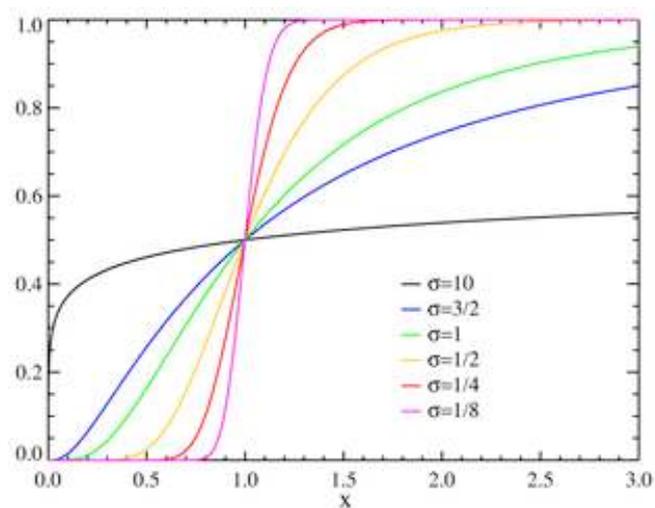
---

The distribution function of this random variable is difficult to compute explicitly.

It may be shown that if  $X$  is a normal random variable, then  $e^X$  has a log-normal distribution.

---

### Lognormal density function




---

For a random variable  $X$  with density  $f$ , the expected value of  $X$ , where  $g$  is a function is defined by

$$E[g(X)] = \int_{-\infty}^{\infty} g(x)f(x)dx.$$

---

For a random variable  $X$  with Normal density

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

$$E(X) = \mu$$

$$Var(X) = E[(X - \mu)^2] = \sigma^2.$$

We write  $X \sim N(\mu, \sigma^2)$ .

---

If  $X \sim N(0, 1)$  (i.e. the mean is 0 and variance is 1) then we call  $X$  a **standard normal random variable** and denote its density function and distribution function as

$$\begin{aligned}\phi(z) &= \frac{1}{\sqrt{2\pi}} e^{(-z^2/2)} \\ \Phi(x) &= \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{(-z^2/2)} dz\end{aligned}$$

The values of  $\Phi(x)$  and those of  $F(x)$  for other standard distributions are available in various computer spreadsheets.

---

For a random variable  $Y$  with lognormal density

$$\begin{aligned}f(x) &= \frac{1}{x\sigma\sqrt{2\pi}} \exp\left\{-\frac{(\ln(x) - \mu)^2}{2\sigma^2}\right\} \text{ for } x > 0. \\ E(X) &= e^{\mu + (\sigma^2/2)} \\ \text{Var}(X) &= E[(X - \mu)^2] = (e^{\sigma^2} - 1)e^{2\mu + \sigma^2}.\end{aligned}$$


---

When an experiment is conducted many times, e.g. a coin is tossed a hundred times, we are generating a sequence of random variables. Such a sequence is called a sequence of **i.i.d.** (independent identically distributed) random variables.

Suppose we gamble on the toss of a coin as follows – if **HEAD** appears then you give me 1 Rupee and if **TAIL** appears then you give me  $-1$  Rupee.

---

So if we play  $n$  round of this game we have generated **i.i.d. sequence of random variables**  $X_1, \dots, X_n$  where each  $X_i$  satisfies

$$X_i = \begin{cases} +1 & \text{with prob. } 1/2 \\ -1 & \text{with prob. } 1/2 \end{cases}$$

Now

$$S_n = X_1 + X_2 + \dots + X_n$$

represents my gain after playing  $n$  rounds of this game.

---

Suppose we play the game  $n$  times and observe my gains/losses

OBSERVATION	Probability
$S_{10} \leq -2$ i.e. I lost at least 6 out of 10	<b>CHANCE</b> 0.38 moderate
$S_{100} \leq -20$ i.e. I lost at least 60 out of 100	0.03 unlikely
$S_{1000} \leq -200$ i.e. I lost at least 600 out of 1000	$1.36^{-10}$ impossible

---

OBSERVATION	PROPORTION	Probability
$ S_{10}  \leq 1$	$\frac{ S_{10} }{10} \leq 0.1$	0.25
$ S_{100}  \leq 8$	$\frac{ S_{100} }{100} \leq 0.08$	0.56
$ S_{1000}  \leq 40$	$\frac{ S_{1000} }{1000} \leq 0.04$	0.8

---

## Law of Large Numbers

Suppose  $X_1, X_2, \dots$  is a sequence of *i.i.d.* random variables with  $E(|X_1|) < \infty$ . Then

$$\bar{X}_n = \sum_{i=1}^n \frac{X_i}{n}$$

converges to  $\mu = E(X_1)$ : *i.e.* for all  $\epsilon > 0$

$$P(|\bar{X}_n - \mu| > \epsilon) \longrightarrow 0.$$


---

Event	Probability	Normal
$\sqrt{1000} \frac{S_{1000}}{1000} \leq 0$	0.5126	$\Phi(0) = 0.5$
$\sqrt{1000} \frac{S_{1000}}{1000} \leq 1$	0.85	$\Phi(1) = 0.84$
$\sqrt{1000} \frac{ S_{1000} }{1000} \leq 1.64$	0.95	$\Phi(1.64) = 0.95$
$\sqrt{1000} \frac{ S_{1000} }{1000} \leq 1.96$	0.977	$\Phi(1.96) = 0.975$

---

For the sequence of *i.i.d.* random variables  $X_1, X_2, \dots$  with

$$X_i = \begin{cases} +1 & \text{with prob. } 1/2 \\ -1 & \text{with prob. } 1/2 \end{cases}$$

we have for  $\bar{X}_n = S_n/n$ ,

$$P\left\{\sqrt{n}\left(\frac{\bar{X}_n - \mu}{\sigma}\right) \leq x\right\} \longrightarrow \Phi(x)$$

### Central Limit Theorem

Suppose  $X_1, X_2, \dots$  is a sequence of *i.i.d.* random variables with  $E(|X_1|^2) < \infty$ . Let  $\mu = E(X_1)$  and  $\sigma^2 = E[(X_1 - \mu)^2]$ . Let

$$\bar{X}_n = \sum_{i=1}^n \frac{X_i}{n}.$$

Then

$$P\left\{\sqrt{n}\left(\frac{\bar{X}_n - \mu}{\sigma}\right) \leq x\right\} \longrightarrow \Phi(x)$$

$X \sim \text{Binomial}(n, p)$ ,  $n$  large. Then

$$P(X \leq a)$$

can be approximated by

$$\Phi\left(\sqrt{n}\left(\frac{a-np}{\sqrt{p(1-p)}}\right)\right)$$

## Chapter 2

# INTRODUCTION TO $\mathbb{R}$

*Notes by Arnab Chakraborty*

# Introduction to R

## Hello, R!

R is a free statistical software. It has many uses including

1. performing simple calculations (like a very powerful pocket calculator)
2. making plots (graphs, diagrams etc),
3. analysing data using ready-made statistical tools (*e.g.*, regression),
4. and above all it is a powerful programming language.

We shall acquaint ourselves with the basics of R in this tutorial.

## Starting R

First you must have R installed in your computer. Then typically you have to hunt for an

icon like  and double click on it. If everything goes well, you should see a window pop up containing something like this.

```
R : Copyright 2005, The R Foundation for Statistical Computing  
Version 2.1.1 (2005-06-20), ISBN 3-900051-07-0
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.
```

```
Natural language support but running in an English locale
```

```
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.
```

```
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for a HTML browser interface to help.  
Type 'q()' to quit R.
```

```
>
```

The `>` is the R *prompt*. You have to type commands in front of this prompt and press the **Enter** key on your keyboard.

## Simple arithmetics

R may be used like a simple calculator. Type the following command in front of the prompt and hit **Enter**.

```
2 + 3
```

Ignore the [1] in the output for the time being. We shall learn its meaning later. Now try

```
2 / 3
```

What about the following? Wait! Don't type these all over again.

```
2 * 3
2 - 3
```



Just hit the **↑** key of your keyboard to *replay* the last line. Now use the **←** and **→** cursor keys and the **Delete** key to make the necessary changes.

**Exercise:** What does R say to the following?

```
2/0
```

Guess what is going to be the result of

```
2/Inf
```

and of

```
Inf/Inf
```

So now you know about three different types of 'numbers' that R can handle: ordinary numbers, infinities, NaN (Not a Number).

## Variables

R can work with variables. For example

```
x = 4
```

assigns the value 4 to the variable  $x$ . This assignment occurs silently, so you do not see any visible effect immediately. To see the value of  $x$  type

```
x
```

This is a very important thing to remember:



To see the value of the variable just type its name and hit **Enter**.

Let us create a new variable

```
y = -4
```

**Exercise:** Try the following.

```
x + y
x - 2*y
x^2 + 1/y
```

The caret (^) in the last line denotes power.

**Exercise:** What happens if you type the following?

```
z-2*x
```

and what about the next line

```
X + Y
```

Well, I should have told you already: R is case sensitive!

**Exercise:** Can you explain the effect of this?

```
x = 2*x
```

## Standard functions

R knows most of the standard functions.

**Exercise:** Try

```
sin(x)
cos(0)
sin(pi) #pi is a built-in constant
tan(pi/2)
```



The part of a line after # is called a **comment**. It is meant for *U*, the UserR who use R! R does not care about comments.

**Exercise:** While you are in the mood of using R as a calculator you may also try

```
exp(1)
log(3)
log(-3)
log(0)
log(x-y)
```

What is the base of the logarithm?

## Getting help

R has many many features and it is impossible to keep all its nuances in one's head. So R has an efficient online help system. The next exercise introduces you to this.

**Exercise:** Suppose that you desperately need logarithm to the base 10. You want to know if R has a ready-made function to compute that. So type

```
?log
```

A new window (the help window) will pop up. Do you find what you need?



Always look up the help of anything that does not seem clear. The technique is to type a question mark followed by the name of the thing you are interested in. All words written like **this** in this tutorial have online help.

Sometimes, you may not know the exact name of the function that you are interested in. Then you can try the **help.search** function.

**Exercise:** Can R compute the Gamma function? As your first effort try

```
Gamma(2)
```

Oops! Apparently this is not the Gamma function you are looking for. So try

```
help.search("Gamma")
```

This will list all the topics that involve Gamma. After some deliberation you can see that "Special Functions of Mathematics" matches your need most closely. So type

```
?Special
```

Got the information you needed?

Searching for functions with names known only approximately is often frustrating.



Sometimes it is easier to google the internet than perform **help.search!**

## Functions

We can type

```
sin(1)
```

to get the value  $\sin 1$ . Here **sin** is a standard built-in function. R allows us to create new functions of our own. For example, suppose that some computation requires you to find the value of

$$f(x) = x/(1-x)$$

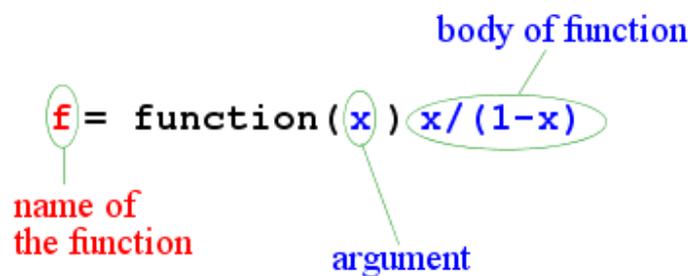
repeatedly. Then we can write function to do this as follows.

```
f = function(x) x/(1-x)
```

Now you may type

```
f(2)
y = 4
f(y)
f(2*y)
```

Here `f` is the name of the function. It can be any name of your choice (as long as it does not conflict with names already existing in R).



### Anatomy of an R function

A couple of points are in order here. First, the choice of the name depends completely on you. Second, the name of the argument is also a matter of personal choice. But you must use the same name also inside the body of the function. It is also possible to write functions of more than one variable.

**Exercise:** Try out the following.

```
g = function(x,y) (x+2*y)/3
g(1,2)
g(2,1)
```

**Exercise:** Write a function with name `myfun` that computes  $x+2*y/3$ . Use it to compute  $2+2*3/3$ .

### Vectors

So far R appears little more than a sophisticated calculator. But unlike most calculators it can handle **vectors**, which are basically lists of numbers.

```
x = c(1,2,4,56)
x
```

The **C** function is for concatenating numbers (or variables) into vectors.

**Exercise:** Try

```
y = c(x, c(-1,5), x)
length(x)
length(y)
```

There are useful methods to create long vectors whose elements are in arithmetic progression:

```
x = 1:20
4:-10
```

If the common difference is not 1 or -1 then we can use the **seq** function

```
y=seq(2,5,0.3)
y
```

**Exercise:** Try the following

```
1:100
```

Do you see the meaning of the numbers inside the square brackets?

**Exercise:** How to create the following vector in R?

```
1, 1.1, 1.2, , ... 1.9, 2, 5, 5.2, 5.4, ... 9.8, 10
```

Hint: First make the two parts separately, and then concatenate them.

### Working with vectors

Now that we know how to create vectors in R, it is time to use them. There are basically three different types of functions to handle vectors.

1. those that work entrywise
2. those that summarise a vector into a few numbers (like finds the sum of all the numbers)
3. others

**Exercise:** Most operations that work with numbers act entrywise when applied to vectors. Try this.

```
x = 1:5
x^2
x+1
2*x
sin(x)
exp(sqrt(x))
```

It is very easy to add/subtract/multiply/divide two vectors entry by entry.

**Exercise:**

```
x = c(1,2,-3,0)
y = c(0,3,4,0)
x+y
x*y
x/y
2*x-3*y
```

Next we meet some functions that summarises a vector into one or two numbers.

**Exercise:** Try the following and guess the meanings of commands.

```
val = c(2,1,-4,4,56,-4,2)
sum(val)
mean(val)
min(val)
max(val)
range(val)
```

**Exercise:** Guess the outcome of

```
which.min(val)
which.max(val)
```

Check your guess with the online help.

### Extracting parts of a vector

If  $x$  is vector of length 3 then its entries may be accessed as  $x[1]$ ,  $x[2]$  and  $x[3]$ .

```
x = c(2,4,-1)
x[1]
x[2]+x[3]
i = 3
x[i]
x[i-1]
x[4]
```

Note that the counting starts from 1 and proceeds left-to-right. The quantity inside the square brackets is called the **subscript** or **index**.



It is also possible to access multiple entries of a vector by using a subscript that is itself a vector.

```
x = 3:10
x[1:4]
x[c(2,4,1)]
```

**Exercise:** What is the effect of the following?

```
x = c(10,3,4,1)
ind = c(3,2,4,1) #a permutation of 1,2,3,4
x[ind]
```

This technique is often useful to rearrange a vector.

**Exercise:** Try the following to find how R interprets **negative subscripts**.

```
x = 3:10
x
x[-1]
x[-c(1,3)]
```

Subscripting allows us to find one or more entries in a vector if we know the position(s) in the vector. There is a different (and very useful) form of subscripting that allows us to extract entries with some given property.

```
x = c(100,2,200,4)
x[x>50]
```

The second line extracts all the entries in `x` that exceed 50. There are some nifty things that we can achieve using this kind of subscripting. To find the sum of all entries exceeding 50 we can use

```
sum(x[x>50])
```

How does this work? If you type

```
x>50
```

you will get a vector of **TRUE**s and **FALSE**s. A **TRUE** stands for a case where the entry exceeds 50. When such a True-False vector is used as the subscript only the entries corresponding to the **TRUE**s are retained. Even that is not all. Internally a **TRUE** is basically a 1, while a **FALSE** is a 0. So if you type

```
sum(x>50)
```

you will get the number of entries exceeding 50.



The number of entries satisfying some given property (like "less than 4") may be found easily like

```
sum(x<4)
```

**Exercise:** If

```
val = c(1,30,10,24,24,30,10,45)
```

then what will be the result of the following?

```
sum(val >= 10 & val <= 40)
sum(val > 40 | val < 10) # | means "OR"
sum(val == 30) # == means "equal to"
sum(val != 24) # != means "not equal to"
```

Be careful with `==`. It is different from `=`. The former means comparing for equality, while the latter means assignment of a value to a variable.

**Exercise:** What does

```
mean(x>50)
```

denote?

**Exercise:** Try and interpret the results of the following.

```
x = c(100, 2, 200, 4)
sum(x>=4)
mean(x!=2)
x==100
```

## Sorting

```
x = c(2, 3, 4, 5, 3, 1)
y = sort(x)
y #sorted
x #unchanged
```

**Exercise:** Look up the help of the `sort` function to find out how to sort in *decreasing* order.

Sometimes we need to order one vector *according to* another vector.

```
x = c(2, 3, 4, 5, 3, 1)
y = c(3, 4, 1, 3, 8, 9)
ord = order(x)
ord
```

Notice that `ord[1]` is the *position* of the smallest number, `ord[2]` is the position of the next smallest number, and so on.

```
x[ord] #same as sort(x)
y[ord] #y sorted according to x
```

## Matrices

R has no direct way to create an arbitrary matrix. You have to first list all the entries of the matrix as a single vector (an  $m$  by  $n$  matrix will need a vector of length  $mn$ ) and then fold the vector into a matrix. To create

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

we first list the entries *column by column* to get 1, 3, 2, 4.

To create the matrix in R:

```
A = matrix(c(1, 3, 2, 4), nrow=2)
A
```

The `nrow=2` command tells R that the matrix has 2 rows (then R can compute the number of columns by dividing the length of the vector by `nrow`.) You could have also typed:

```
A <- matrix(c(1,3,2,4),ncol=2) #<- is same as =
A
```

to get the same effect. Notice that R folds a vector into a matrix *column by column*. Sometimes, however, we may need to fold *row by row* :

```
A = matrix(c(1,3,2,4),nrow=2,byrow=T)
```

The **T** is same as **TRUE**.

**Exercise:** Matrix operations in R are more or less straight forward. Try the following.

```
A = matrix(c(1,3,2,4),ncol=2)
B = matrix(2:7,nrow=2)
C = matrix(5:2,ncol=2)
dim(B) #dimension
nrow(B)
ncol(B)
A+C
A-C
A%%C #matrix multiplication
A*C #entrywise multiplication
A%%B
t(B)
```

Subscripting a matrix is done much like subscripting a vector, except that for a matrix we need two subscripts. To see the (1,2)-th entry (*i.e.*, the entry in row 1 and column 2) of A type

```
A[1,2]
```

**Exercise:** Try out the following commands to find what they do.

```
A[1,]
B[1,c(2,3)]
B[,-1]
```

### Working with rows and columns

Consider the following.

```
A = matrix(c(1,3,2,4),ncol=2)
sin(A)
```

Here the **sin** function applies *entrywise*. Now suppose that we want to find the sum of each column. So we want to apply the sum function *columnwise*. We achieve this by using the **apply** function like this:

```
apply(A, 2, sum)
```

The 2 above means *columnwise*. If we need to find the *rowwise* means we can use

```
apply(A, 1, mean)
```

## Lists

Vectors and matrices in R are two ways to work with a collection of objects. **Lists** provide a third method. Unlike a vector or a matrix a list can hold different kinds of objects. Thus, one entry in a list may be a number, while the next is a matrix, while a third is a character string (like "Hello R!"). Lists are useful to store different pieces of information about some common entity. The following list, for example, stores details about a student.

```
x = list(name="Rama", nationality="Indian", height=5.5,
marks=c(95,45,80))
```

We can now extract the different fields of `x` as

```
names(x)
x$name
x$hei #abbrevs are OK
x$marks
x$m[2]
x$na #oops!
```

In the coming tutorials we shall never need to make a list ourselves. But the statistical functions of R usually return the result in the form of lists. So we must know how to unpack a list using the `$` symbol as above.



To see the online help about symbols like `$` type



```
?"$"
```



Notice the double quotes surrounding the symbol.



Let us see an example of this. Suppose we want to write a function that finds the length, total and mean of a vector. Since the function is returning three different pieces of information we should use lists as follows.

```
f = function(x) list(len=length(x), total=sum(x), mean=mean(x))
```

Now we can use it like this:

```
dat = 1:10
result = f(dat)
names(result)
result$len
result$tot
```

```
result$mean
```

## Doing statistics with R

Now that we know R to some extent it is time to put our knowledge to perform some statistics using R. There are basically three ways to do this.

1. Doing elementary statistical summarisation or plotting of data
2. Using R as a calculator to compute some formula obtained from some statistics text.
3. Using the sophisticated statistical tools built into R.

In this first tutorial we shall content ourselves with the first of these three. But first we need to get our data set inside R.

### Loading a data set into R

We shall consider part of a data set given in

Distance to the Large Magellanic Cloud: The RR Lyrae Stars Gisella Clementini, Raffaele Gratton, Angela Bragaglia, Eugenio Carretta, Luca Di Fabrizio, and Marcella Maio *Astronomical Journal* 125, 1309-1329 (2003).

We have slightly doctored the data file to make it compatible with R. The file is called `LMC.dat` and resides in some folder `F:\astro`, say. The data set has two columns with the headings `Method`, `Dist` and `Err`. Here are the first few lines of the file:

<code>Method</code>	<code>Dist</code>	<code>Err</code>
<code>"Cepheids: trig. paral."</code>	<code>18.70</code>	<code>0.16</code>
<code>"Cepheids: MS fitting"</code>	<code>18.55</code>	<code>0.06</code>
<code>"Cepheids: B-W"</code>	<code>18.55</code>	<code>0.10</code>

There are various ways to load the data set. One is to use

```
LMC = read.table("F:/astro/LMC.dat", header=T)
```

Note the use of forward slash (/) even if you are working in Windows. Also the `header=T` tells that the first line of the data file gives the names of the columns. Here we have used the *absolute path* of the data file. In Unix the absolute path starts with a forward slash (/).

```
dim(LMC)
names(LMC)
LMC
```

This object `LMC` is like a matrix (more precisely it is called a **data frame**). Each column stores the values of one variable, and each row stores a case. Its main difference with a matrix is that different columns can hold different types of data (for example, the `Method` column stores character strings, while the other two columns hold numbers). Otherwise, a data frame is really like a matrix. We can find the mean of the `Dist` variable like this

```
mean(LMC[,2])
mean(LMC[, "Dist"])
```

Note that each column of the `LMC` matrix is a variable, so it is tempting to write

```
mean(Dist)
```

but this will not work, since `Dist` is inside `LMC`. We can "bring it out" by the command

```
attach(LMC)
```

Now the command

```
mean(Dist)
```

works perfectly. All the values of the `Dist` variable are different measurements of the same distance. So it is only natural to use the average as an estimate of the true distance. But the `Err` variable tells us that not all the measurements are equally reliable. So a better estimate might be a weighted mean, where the weights are inversely proportional to the errors. We can use R as a calculator to directly implement this formula:

```
sum(Dist/Err)/sum(1/Err)
```

or you may want to be a bit more explicit

```
wt = 1/Err
sum(Dist*wt)/sum(wt)
```

Actually there is a smarter way than both of these.

```
weighted.mean(Dist, 1/Err)
```

## Script files

So far we are using R *interactively* where we type commands at the prompt and the R executes a line before we type the next line. But sometimes we may want to submit many lines of commands to R at a single go. Then we need to use scripts.



Use script files to save frequently used command sequences. Script files are also useful for replaying an analysis at a later date.

A script file in R is a text file containing R commands (much as you would type them at the prompt). As an example, open a text editor (*e.g.*, notepad in Windows, or gedit in Linux). Avoid fancy editors like MSWord. Create a file called, say, `test.r` containing the following lines.

```
x = seq(0,10,0.1)
y = sin(x)
plot(x,y,ty="l") #guess what this line does!
```

Save the file in some folder (say `F:/astro`). In order to make R execute this script type

```
source("F:/astro/test.r")
```

If your script has any mistake in it then R will produce error messages at this point. Otherwise, it will execute your script.

The variables `x` and `y` created inside the command file are available for use from the prompt now. For example, you can check the value of `x` by simply typing its name at the prompt.

```
x
```

Commands inside a script file are executed pretty much like commands typed at the prompt. One important difference is that in order to **print** the value of a variable `x` on the screen you have to write

```
print(x)
```

Merely writing

```
x
```

on a line by itself will not do inside a script file.



Printing results of the intermediate steps using **print** from inside a script file is a good way to debug R scripts.



## Chapter 3

# DESCRIPTIVE STATISTICS & GRAPHING WITH R

*Notes by Arnab Chakraborty*

# Descriptive Statistics and Graphing with R

In this tutorial we shall learn to perform simple statistical analysis and plotting of data with R. The parts involving astronomical information are based on the notes by Prof. David Hunter.

## Getting astronomical data

The astronomical community has a vast complex of on-line databases. Many databases are hosted by data centres such as the [Centre des Donnees astronomiques de Strasbourg \(CDS\)](#), the [NASA/IPAC Extragalactic Database \(NED\)](#), and the [Astrophysics Data System \(ADS\)](#). The Virtual Observatory (VO) is developing new flexible tools for accessing, mining and combining datasets at distributed locations; see the Web sites for the [international](#), [European](#), and [U.S.](#) VO for information on recent developments. The [VO Web Services](#), [Summer Schools](#), and [Core Applications](#) provide helpful entries into these new capabilities.

We initially treat here only input of tabular data such as catalogs of astronomical sources. We give two examples of interactive acquisition of tabular data. One of the multivariate tabular datasets used here is a dataset of stars observed with the European Space Agency's Hipparcos satellite during the 1990s. It gives a table with 9 columns and 2719 rows giving Hipparcos stars lying between 40 and 50 parsecs from the Sun. The dataset was acquired using CDS's [Vizie Catalogue Service](#) as follows:

- In Web browser, go to [http://vizier.u-strasbg.fr/viz-bin/VizieR?-source=I/239/hip\\_main](http://vizier.u-strasbg.fr/viz-bin/VizieR?-source=I/239/hip_main)
- Set Max Entries to 9999, Output layout ASCII table
- Remove "Compute r" and "Compute Position" buttons
- Set parallax constraint "20 .. 25" to gives stars between 40 and 50 pc
- Retrieve 9 properties: HIP, Vmag, RA(ICRS), DE(ICRS), Plx, pmRA, pmDE, e\_Plx, and B-V
- Submit Query
- Use ASCII editor to trim header to one line with variable names
- Trim trailer
- Indicate missing values by NA.
- Save ASCII file on disk for ingestion into R

## Reading the data into R

Let us assume that the data set is in

```
F:\astro\HIP.dat
```

We have already learned how to use the absolute path `F:\astro\HIP.dat` to load the data set into R. Now we shall learn a two step process that is usually easier. First navigate to the correct folder/directory

```
setwd("F:/astro") #notice the forward slash
getwd() #just to make sure
```

The function `setwd` means "set working directory".

Now load the data set

```
hip = read.table("HIP.dat", header=T)
```

 The advantage of using **setwd** is that you have to type the name of the folder/directory only once. All files (data/script) in that folder can then be referred to by just their names.

After the loading is complete we should make sure that things are as they should be. So we check the size of the data set, the variable names.

```
dim(hip)
names(hip)
```

Let us take a look at the first 3 rows of the data set.

```
hip[1:3,]
```

**Exercise:** What command should you use to see the first 2 columns?

There is a variable called RA in the data set. It corresponds to column 3 in the data set. To see its values you may use either

```
hip[,3]
```

or

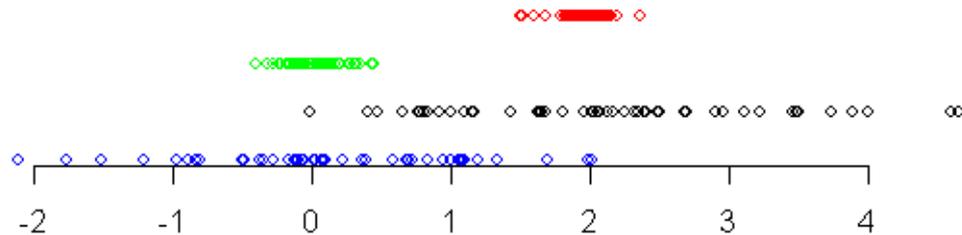
```
hip[,"RA"]
```

Incidentally, the first column is just an index variable without any statistical value. So let us get rid of it.

```
hip = hip[,-1]
```

## Summarising data

The following diagram shows four different data sets along a number line.



### Four data sets shown along a number line

Notice that the points in the red data set (topmost) and the black data set (third from top) are more or less around the same centre point (approximately 2). The other two data sets are more or less around the value 0. We say that the red and black data sets have the same **central tendency**, while the other data sets have a different central tendency.

Again, notice that the points in the red and blue data sets (the topmost two) are tightly packed, while the other two data sets have larger spreads. We say that the bottom two data sets have larger **dispersion** than the top two.

### Central tendency

When summarising a data set we are primarily interested in learning about its central tendency and dispersion. The central tendency may be obtained by either the **mean** or **median**. The median is the most central value of a variable. To find these for *all* the variables in our data set we **apply** the **mean** and **median** function on the columns.

```
apply(hip, 2, mean)
```

Have you noticed the mean of the last variable? It is **NA** or "Not Available", which is hardly surprising since not all the values for that variable were present in the original data set. We shall learn later how to deal with missing values (**NA**s).

**Exercise:** Find the **median** of all the variables.

### Dispersion

Possibly the simplest (but not the best) way to get an idea of the dispersion of a data set is to compute the min and max. R has the functions **min** and **max** for this purpose.

```
apply(hip, 2, min)
apply(hip, 2, max)
```

In fact, we could have applied the **range** function to find both min and max in a single line.

```
apply(hip, 2, range)
```

The most popular way to find the dispersion of a data set is by using the **variance** (or its positive square root, the **standard deviation**). The formula is

$$\text{variance} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2,$$

where  $\bar{x}$  is the mean of the data. The function **var** and **sd** compute the variance and standard deviation.

```
var(hip[, "RA"])
sd(hip[, "RA"])
```

Another popular measure of dispersion is the **median absolute deviation** (or MAD) is proportional to the median of the absolute distances of the values from the median. It is given by the following formula.

$$MAD = 1.4826 \times \text{median}\{|x_1 - \text{median}|, \dots, |x_n - \text{median}|\}.$$

The constant of proportionality happens to be the magic number 1.4826 for some technical reason. For example, if we have just 3 values 1, 2 and 4, then

- the median is 2,
- absolute deviations from median are  $|1-2|=1$ ,  $|2-2|=0$  and  $|4-2|=2$ ,
- median of the absolute deviations is 1,
- $MAD = 1.4826$ .

The function **mad** computes this.

```
mad(c(1, 2, 4))
```

For example,

```
mad(hip[, 1])
```

We want to compute both the median and MAD using one function. So we write

```
f = function(x) c(median(x), mad(x))
f(hip[, 1])
```

**Exercise:** What will be the result of the following?

```
apply(hip, 2, f)
```

There is yet another way to measure the dispersion of a data set. This requires the concept of a **quantile**. Some examinations report the grade of a student in the form of **percentiles**. A 90-percentile student is one whose grade is exceeded by 10% of all the students. The quantile is the same concept except that it talks about proportions instead of percentages. Thus, the 90-th percentile is 0.90-th quantile.

**Exercise:** The median of a data set is the most central value. In other words, exactly half of the data set exceeds the median. So for what value of  $p$  is the median the  $p$ -th

quantile?

The R function **quantile** (not surprisingly!) computes quantiles.

```
quantile(hip[,1],0.10)
quantile(hip[,1],0.50)
median(hip[,1])
```

The 0.25-th and 0.75-th quantiles are called the **first quartile** and the **third quartile**, respectively.

**Exercise:** What is the **second quartile**?

```
quantile(hip[,1],c(0.25,0.50,0.75))
```

The difference between first and third quartiles is another measure of the dispersion of a data set, and is called the **InterQuartile Range (IQR)**. There is function called **summary** that computes quite a few of the summary statistics.

```
summary(hip)
```

**Exercise:** Look up the online help for the functions **cov** and **cor** to find out what they do. Use them to find the covariance and correlation between **RA** and **pmRA**.

## Handling missing values

So far we have ignored the **NA** problem completely. The next exercise shows that this is not always possible in R.

**Exercise:** The function **var** computes the variance. Try **applying** it to the columns of our data set.

**NA** denotes missing data in R. It is like a different kind of number in R (just like Inf, or NaN). Any mathematics with **NA** produces only **NA**

```
NA + 2
NA - NA
```

The function **is.na** checks for presence of **NA**s in a vector or matrix.

```
x = c(1,2,NA)
is.na(x)
any(is.na(x))
```

The function **any** reports TRUE if there is at least one TRUE in its argument vector. The **any** and **is.na** combination is very useful. So let us make a function out of them.

```
hasNA = function(x) any(is.na(x))
```

**Exercise:** What is the consequence of this?

```
apply(hip, 2, hasNA)
```

This exercise shows that only the last variable has **NA**s in it. So naturally the following commands

```
min(B.V)
max(B.V)
mean(B.V)
```

all return **NA**. But often we want to apply the function on only the non-**NA**s. If this is what we want to do all the time then we can omit the **NA** from the data set itself in the first place. This is done by the **na.omit** function

```
hip1 = na.omit(hip)
dim(hip)
dim(hip1)
```

This function takes a very drastic measure: it simply wipes out all rows with at least one **NA** in it.

```
apply(hip, 2, mean)
apply(hip1, 2, mean)
```

Notice how the other means have also changed. Of course, you may want to change only the **B.V** variable. Then you need

```
B.V1 = na.omit(hip[, "B.V"])
```

**Exercise:** Compute the variances of all the columns of `hip1` using **apply**.

There is another way to ignore the **NA**s without omitting them from the original data set.

```
mean(hip[, "B.V"], na.rm=T)
var(hip[, "B.V"], na.rm=T)
```

Here `na.rm` is an argument that specifies whether **NA**s should be **removed**. By setting it equal to **T** (or **TRUE**) we are asking the function to remove all the obnoxious **NA**s.

You can use this inside **apply** as well

```
apply(hip, 2, var, na.rm=T)
```

## Attaching a data set

A data set in R is basically a matrix where each column denotes a variable. The `hip` data set, for example, has 8 variables (after removing the first column) whose names are obtained as

```
names(hip)
```

To access the **RA** variable we may use

```
hip[, "RA"] # too much to type
```

or

```
hip[,3] # requires remembering the column number
```

Fortunately, R allows a third mechanism to access the individual variables in a data set that is often easier. Here you have to first **attach** the data set

```
attach(hip)
```

This unpacks the data set and makes its columns accessible by name. For example, you can now type

```
RA # instead of hip[,"RA"]
mean(RA)
hasNA(RA)
```

We can of course still write

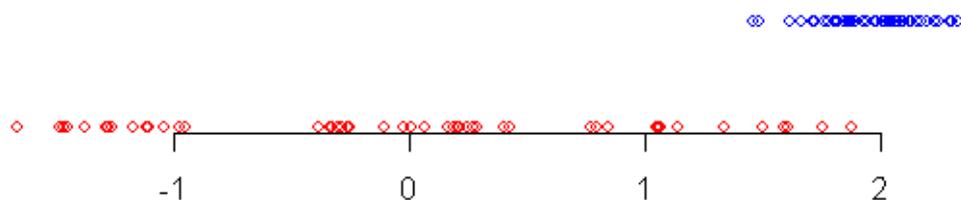
```
hip[,"RA"]
```

## Making plots

Graphical representations of data are a great way to get a ``feel" about a data set, and R has a plethora of plotting functions.

### Boxplots

Consider the two data sets shown along a number line.

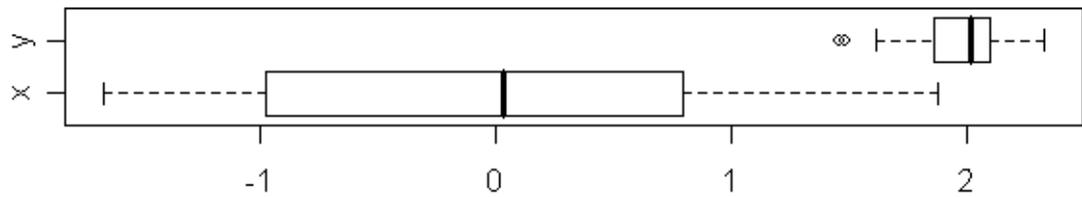


### Two data sets

When we look at the data sets for the first time our eyes pick up the following details:

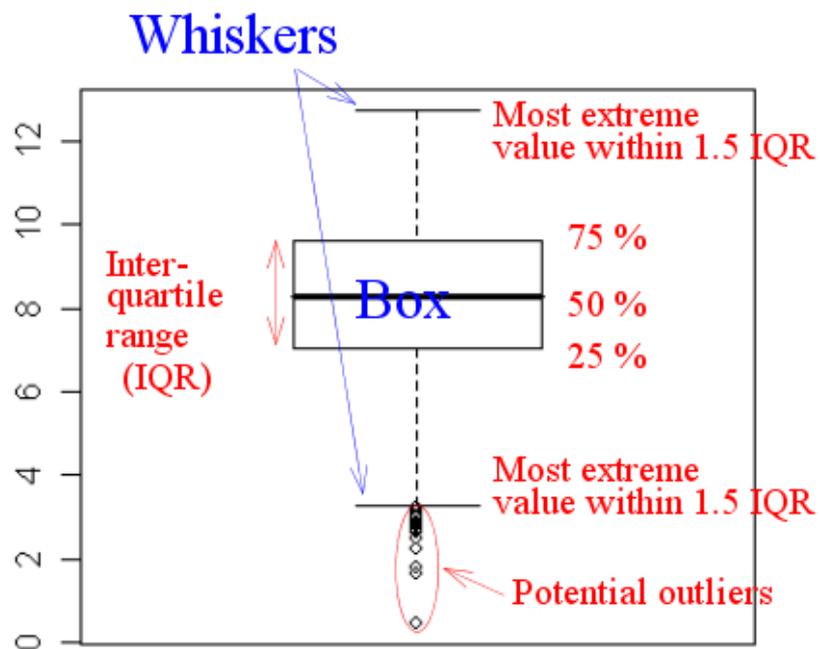
- the blue data set (topmost) has smaller spread than the red one
- the central tendency of blue data set is more to the right than the red one
- there are some blue points somewhat away from the bulk of the data.

In other words, our eye notices where the bulk of the data is, and is also attracted by points that are away from the bulk. The boxplot is a graphical way to show precisely these aspects.



### Boxplots for the two data sets

It requires some knowledge to interpret a boxplot (often called a box-and-whiskers plot). The following diagram might help.



### An annotated boxplot

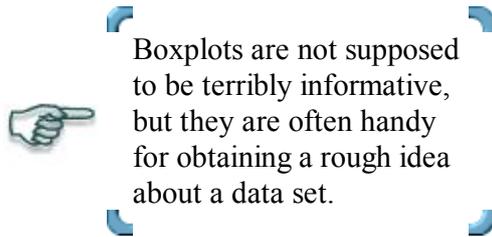
Let us use the `boxplot` function on our data set.

```
boxplot(Vmag)
```

Boxplots are usually more informative when more than one variable are plotted side by side.

```
boxplot(hip)
```

The size of the box roughly gives an idea about the spread of the data.



## Scatterplots

Next let us make a **scatterplot**.

```
plot(RA, DE)
```

This produces a **scatterplot**, where each pair of values is shown as a point. R allows a lot of control on the appearance of the plot. See the effect of the following.

```
plot(RA, DE, xlab="Right ascension", ylab="Declination",
      main="RA and DE from Hipparcos data")
```

You may change the colour and point type.

```
plot(RA, DE, pch=".", col="red")
```

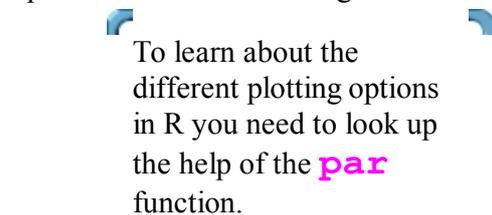
Sometimes it is important to change the colours of *some* points. Suppose that we want to colour red all the points with *DE* exceeding 0. Then the **ifelse** function comes handy.

```
cols = ifelse(DE>0, "red", "black")
cols
```

This means "*cols* is red if *DE*>0, else it is black".

```
plot(RA, DE, col=cols)
```

You may similarly use a vector for *pch* so that different points are shown differently. There are many other plotting options that you can learn using the online help. We shall explain some of these during these tutorials as and when needed.



```
?par
```

It has a long list of options. Before attempting to make your first publication-quality graph with R you should better go through this list.

**Exercise:** Make a scatterplot of *RA* and *pmRA*. Do you see any pattern?

Instead of making all such plots separately for different pairs of variables we can make a **scatterplot matrix**

```
plot(hip, pch=".")
```

## Histograms

Histograms show how densely or sparsely the values of a variable lie at different points.

```
hist(B.V)
```

The histogram shows that the maximum concentration of values occurs from 0.5 to 1. The vertical axis shows the number of values. A bar of height 600, standing on the range 0.4 to 0.6, for example, means there are 600 values in that range. Some people, however, want to scale the vertical axis so that the total area of all the rectangles is 1. Then the area of each rectangle denotes the probability of its range.

```
hist(B.V, prob=T)
```

## Multiple plots

Sometimes we want more than one plot in a single page (mainly to facilitate comparison and printing). The way to achieve this in R is rather weird. Suppose that we want 4 plots laid out as a 2 by 2 matrix in a page. Then we need to write

```
oldpar = par(mfrow=c(2,2))
```

The **par** function sets graphics options that determines how subsequent plots should be made.



The **par** function controls the global graphics set up. All future plots will be affected by this function. Everytime it is called the old set up is returned by the function. It is a good idea to save this old set up (as we have in a variable called `oldpar`) so that we can restore the old set up later.

Here `mfrow` means **multi-frame row-wise**. The vector `c(2,2)` tells R to use a 2 by 2 layout. Now let us make 4 plots. These will be added to the screen row by row.

```
x = seq(0,1,0.1)
plot(x, sin(x), ty="l")
hist(RA)
plot(DE, pmDE)
boxplot(Vmag)
```

To restore the original "one plot per page" set up use

```
par(oldpar)
```

## Adding to existing plots

Sometimes we want to add something (line, point etc) to an existing plot. Then the functions **abline**, **lines** and **points** are useful.

```
plot(RA, DE)
abline(a=-3.95, b=0.219)
```

This adds the line  $y = a + bx$  to the plot. Also try

```
abline(h=0.15)
abline(v=18.5)
```

To add curved lines to a plot we use the **lines** function.

```
x = seq(0, 10, 0.1)
plot(x, sin(x), ty="l")
lines(x, cos(x), col="red")
```

We can add new points to a plot using the **points** function.

```
points(x, (sin(x)+cos(x))/2, col="blue")
```

There are more things that you can add to a plot. See, for example, the online help for the **text** and **rect** functions.

## Extracting the Hyades stars

Sometimes we have to work with only a subset of the entire data. We shall illustrate this next by selecting only the Hyades stars from the data set. To do this we shall use the facts★ that the Main Sequence Hyades stars have

- RA in the range (50,100)
- DE in the range (0,25)
- pmRA in the range (90,130)
- pmDE in the range (-60,-10)
- e\_Plx <5
- Vmag >4 OR B.V <0.2 (this eliminates 4 red giants)

★This are borrowed from Prof Hunter's notes, where he uses astronomy knowledge to obtain these conditions by making suitable plots. The interested reader is encouraged to look into his notes for details.

Let us see how we apply these conditions one by one. First, we shall **attach** the data set so that we may access each variable by its name.

```
attach(hip)
```

Next we shall apply the conditions as filters.

```
filter1 = (RA>50 & RA<100 & DE>0 & DE<25)
filter2 = (pmRA>90 & pmRA<130 & pmDE>-60 & pmDE< -10)
filter3 = filter1 & filter2 & e_Plx<5
HyadFilter = filter3 & (Vmag>4 | B.V <0.2)
```

The **&** denotes (as expected) logical AND while the vertical bar **|** denotes logical OR.

We are going to need this filter in the later tutorials. So it is a good idea to save these

lines in a script file called, say, `hyad.r`.

By the way, the filters are all just vectors of **TRUE**s and **FALSE**s. The entry for a star is **TRUE** if and only if it is a Hyades star.

Now we shall apply the filter to the data set. This produces a new (filtered) data set which we have called `hyades`. Finally we **attach** this data set.

```
hyades = hip[HyadFilter,]  
attach(hyades)
```

You'll see a bunch of warning messages when you **attach** the filtered data set. This is because the old (unfiltered) variables are now being superceded by the new (filtered) variables of the same name.

R always issues a warning whenever a variable from a new data set clashes with some existing variable of the same name. This prevents the user from accidentally changing a variable. In our case, however, we did it deliberately. So we can ignore the warning.

All subsequent command will work with only Hyades stars.

```
dim(hyades)  
plot(Vmag, B.V)
```

We shall often work with the Hyades stars in the later tutorials. So let us save in a script file `hyad.r` the commands to extract the Hyades star.



## Chapter 4

# ESTIMATION, CONFIDENCE INTERVALS & HYPOTHESIS TESTING

*Notes by Donald Richards & Bhamidi V Rao*

1. A problem.

Van den Bergh (1985, ApJ 297, p. 361) considered the luminosity function (LF) for globular clusters in various galaxies.

V-d-B's conclusion: The LF for clusters in the Milky Way is adequately described by a normal distribution (see the graph on the next page).

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x - \mu)^2}{2\sigma^2}\right].$$

$\mu$  ( $\equiv M_0$ ) is the mean visual absolute magnitude and  $\sigma$  is the standard deviation of visual absolute magnitude. Magnitudes are log variables (a log-normal distribution). This appears to be one of the few normal distributions in astronomy.

Statistical Problems:

1. On the basis of collected data, estimate the numbers  $\mu$  and  $\sigma$ . Also, derive a plausible range of values for each of them; etc.
2. V-d-B concludes that the LF is "adequately described" by a normal distribution. How can we quantify the plausibility of this conclusion?

In this lecture, we shall mainly introduce the relevant statistical concepts and vocabulary.

BERGH

Vol. 297

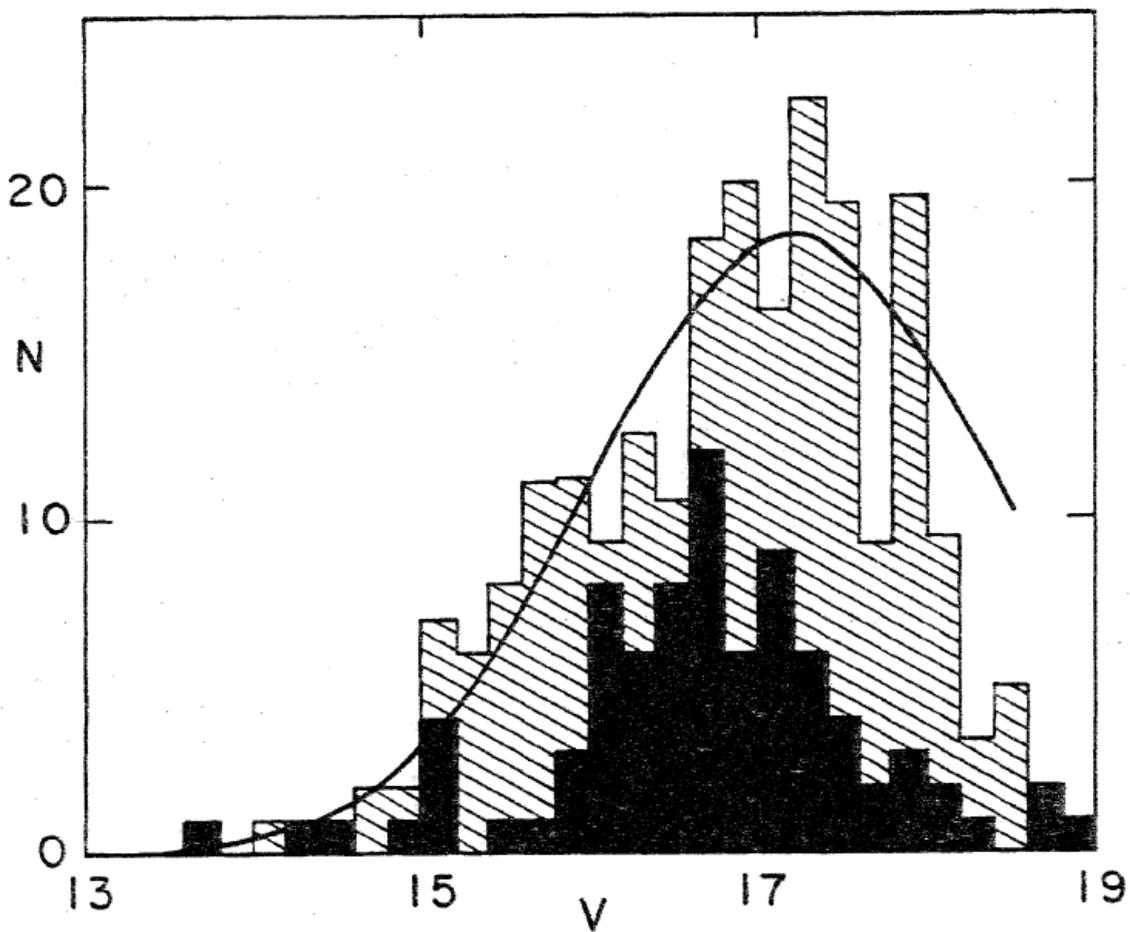


FIG. 1.—Luminosity function of clusters with  $0.70 \leq B-V < 1.00$  in M31. Smooth curve is a Gaussian with  $V(\text{max}) = 17.2$  and  $\sigma = 1.2$  mag. Lower histogram shows the luminosity function of the halo of M31 derived by Racine and Shara (1979).

## 2. Some terminology.

**Population:** This term is used in two different contexts. If you are studying the luminosity function of globular clusters, then the globular clusters constitute the population and you select a sample from this population and make measurements to draw conclusions. Second context, and this is how we use, is the following. You want to study a particular attribute  $X$ , like the luminosity. You make a probabilistic model for this attribute. For example you may want to say that the possible values of  $X$  follow a particular density,  $f(x)$ . Then this model is called the population. Thus, normal population means that the attribute under study obeys density

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[ -\frac{(x - \mu)^2}{2\sigma^2} \right],$$

for some numbers  $\mu$  and  $\sigma > 0$ .

Remember, this means that the chances of your observation falling between, say, 4 and 10, is the area under the above curve  $f(x)$  between these two values  $x = 4$  and  $x = 10$ . In practice, this means the proportion of observations that lie in the interval  $(4, 10)$  equals this area (approximately). Only when we prescribe the values of  $\mu$  and  $\sigma$ , the model is completely specified. Otherwise, it is a class of models for the attribute.

The function  $f(x)$  is called the **probability density function (p.d.f.)** of  $X$ . A **statistical model** is a choice of p.d.f. for  $X$ . We wish to choose a model which “adequately describes” data collected on  $X$ . A **parameter** is a number that appears in the choice of the density, which is to be determined from observations. For example,  $\mu$  and  $\sigma$  are parameters for the p.d.f. of the LF for globular clusters. **parameter space** is the set of permissible values of the parameters. In the above normal model, the parameter space is  $\Theta = \{(\mu, \sigma) : -\infty < \mu < \infty, \sigma > 0\}$ . Thus, for example,  $\sigma$  can not be negative.

A **random sample** means mutually independent random variables  $X_1, \dots, X_n$ ; which all have the same distribution as  $X$ . Here  $n$  is called the size of the sample. In practice, this amounts to data values  $X_1, \dots, X_n$  which are *fully representative* of the population.

On the one hand, I said that  $X_1$  is a random variable and then I said

that this amounts to the data point  $X_1$ . Before confusion sets in, we need to clarify this. We pick a cluster at random and measure its luminosity  $X_1$ . Since we picked at random, before we made the observation,  $X_1$  could have been any number. Of course, the chances that  $X_1$  lies in an interval is given by the corresponding area under the above curve. Thus  $X_1$  is a random variable. When we made the actual observation, we ended up with a number. In other words, we have *one realization* of this random variable. When we pick another cluster at random and make observation, you may no longer obtain the same number. It will be  $X_2$ , another realization of the random variable. Since we pick this cluster independently, this is realization of an independent random variable.

In general, Roman letters are used to represent data, and Greek letters are used to represent parameters. For example  $\theta$ ,  $\mu$ ,  $\sigma$  are parameters where as  $X_1, \dots, X_n$ , are data. A **statistic** is a number computed from the observations, that is, from the random sample  $X_1, \dots, X_n$ . Here are two examples. Sample mean defined as  $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$  and sample variance defined as  $S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$  are statistics. Note that these are random variables. When you compute this quantity at your data points, that is, for your observations, then you get *one realization* of the statistic. In general, a statistic could be any function  $Y = u(X_1, \dots, X_n)$  of the observations.

The **sampling distribution** of a statistic is the probability distribution of the statistic. For example, if  $X_1, \dots, X_n$  is sample from the normal population described above, and the sample mean  $\bar{X}$  is the statistic, then the sampling distribution of this statistic is also normal (it can be shown), but with parameters  $\mu$  and  $\sigma/\sqrt{n}$ .

An **estimator** or **estimate** for a parameter is a statistic. Thus estimator is nothing but a number computed from the sample. But when you say estimator, you should also say the parameter for which this is proposed as an estimator. For example  $\bar{X}$  is an estimator for  $\mu$  and  $S^2$  is an estimator for  $\sigma^2$ . These are also called **point estimators**, simply because based on observations, they provide us with *one* number as a possible value for the corresponding parameter.

### 3. Estimation.

As mentioned earlier, unless the parameters are explained, the model is not fully specified. Having proposed a class of models for the attribute under study, how do we estimate the parameters, to fully specify the model. For example, in modeling the LF, the proposal was that a normal model fits the data. But which normal model?

How do we construct estimates and how do we know a good estimate from a bad one. There are several methods for constructing estimates for the parameters.

Judicious guessing, the method of Maximum Likelihood (use a model that maximizes the chance of coming up with your data), the method of Moments (use a model that makes sample moments agree with population moments, at least the first few), method of Minimum  $\chi^2$  (use a model that reduces discrepancy with what you observed and what is expected to be observed according to the model), Bayesian methods (use some prior knowledge as to what values of the parameter are most likely and what values are less likely), Decision-theoretic methods etc.

There are several criteria proposed for estimators. Unbiased estimator (long run average of the estimate equals the true parameter value), Consistent estimator (with more and more observations, the estimate gets nearer the true parameter value), Efficient estimator (has small variability), etc. Keep in mind that an estimator is a random variable, because it depends on the observations and the observations are, in turn, realizations of random variables.

An estimator  $Y$  for a parameter  $\theta$  is **unbiased** if  $E(Y) = \theta$ . Intuitively,  $Y$  is unbiased if its long-term average value is equal to  $\theta$ . In the above normal population model,  $\bar{X}$  is an unbiased estimator of  $\mu$ . This is because,  $E(X_i) = \mu$  for each  $i$  and hence  $E(\bar{X}) = \mu$ . Also  $S^2$  is an unbiased estimator of  $\sigma^2$ . On the other hand, if you put  $Y$  as the largest of the observations, then  $Y$  is NOT an unbiased estimator. It appears that, after all, this maximum is one of the observations and each observation has expected value  $\mu$ , so  $Y$  must have the same property. But it is not so. Indeed, if the sample size is at least two, then  $E(Y) > \mu$ .

An estimator is consistent if it gets closer and closer to the parameter value as the sample size increases. One way of stating this is to say that the chances of it differing from the parameter, by a preassigned quantity, become smaller and smaller, no matter what the preassigned quantity is. More precisely, an estimator  $Y$ , for a parameter  $\theta$ , is **consistent** if for any  $\epsilon > 0$ , we have  $P(|Y - \theta| \geq \epsilon) \rightarrow 0$  as  $n \rightarrow \infty$ . You should remember that the estimator  $Y$  depends on the sample size  $n$ . Actually, we should have written  $Y_n$  for the estimator based on a sample of size  $n$ . In the above normal population model  $\bar{X}$  is a consistent estimator of  $\mu$ . This is because, given any  $\epsilon > 0$ ,

$$P(|\bar{X} - \mu| \geq \epsilon) \leq \frac{\text{var}(\bar{X})}{\epsilon^2} = \frac{\sigma^2}{\epsilon^2} \frac{1}{n} \rightarrow 0.$$

Here we have used the Chebyshev's inequality.

This argument shows that for any model, with  $\mu$  denoting the population mean,  $\bar{X}$  is a consistent estimator of  $\mu$ . This depends only on the fact that the variance of  $\bar{X}$  is  $\sigma^2/n$  where  $\sigma^2$  is population variance, assumed to be finite. In fact the same argument gives us a general fact: If  $Y_n$  is an unbiased estimator (based on a sample of size  $n$ ) of  $\theta$  and if  $\text{var}(Y_n) \rightarrow 0$ , then  $Y_n$  is a consistent estimator of  $\theta$ .

If  $Y$  is unbiased estimator of  $\theta$ , then, of course,  $E(Y - \theta)^2$  is nothing but the variance of  $Y$ . However, if  $Y$  is not unbiased, then this is no longer the variance of  $Y$ . This quantity  $E(Y - \theta)^2$  is called the **Mean Square Error (MSE)**. An estimator is said to have **minimum mean square error** if this quantity is the least possible. When the estimator is unbiased, then mean square error being its variance, an unbiased estimator with minimum mean square error is called **Minimum Variance Unbiased Estimator (MVUE)**.  $\bar{X}$  has minimum variance among all (unbiased) estimators which are linear combinations of  $X_1, \dots, X_n$ .

#### 4. Confidence intervals.

Point estimators are not always perfect. We wish to quantify the accuracy of the estimator. One way to measure the accuracy is to see its variance. The smaller the variance, the better it is. But there is a fundamentally different

method of looking at the problem of estimation. Instead of saying that a number is an estimator of the parameter  $\mu$ , why not prescribe an interval and quantify by saying that the parameter lies in this interval with a certain probability which is high. This leads to the notion of confidence intervals.

Let us start with our normal example for LF. We know that  $\bar{X}$  is an unbiased estimator of  $\mu$ , its variance is  $\sigma^2/n$  and in fact  $\bar{X} \sim N(\mu, \frac{\sigma^2}{n})$ . As a result,

$$\frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim N(0, 1).$$

If  $Z \sim N(0, 1)$ , then  $P(-1.96 < Z < 1.96) = 0.95$ , so that

$$P\left(-1.96 < \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} < 1.96\right) = 0.95.$$

The above inequality can be restated as

$$P\left(\bar{X} - 1.96\frac{\sigma}{\sqrt{n}} < \mu < \bar{X} + 1.96\frac{\sigma}{\sqrt{n}}\right) = 0.95.$$

The probability that the interval

$$\left(\bar{X} - 1.96\frac{\sigma}{\sqrt{n}}, \bar{X} + 1.96\frac{\sigma}{\sqrt{n}}\right)$$

“captures”  $\mu$  is 0.95. This interval is called a 95% **confidence interval** for  $\mu$ . It is a plausible range of values for  $\mu$  together with a quantifiable measure of its plausibility.

A confidence interval is a *random* interval; it changes as the collected data changes. This explains why we say “a 95% confidence interval” rather than “the 95% confidence interval”. We chose the “cutoff limits”  $\pm 1.96$  symmetrically around 0 to minimize the length of the confidence interval. “cutoff limits” are also called “percentage points”.

Example (devised from van den Bergh, 1985):  $n = 145$  Galactic globular clusters.  $\bar{x} = -7.11$  mag. Let us take  $\sigma = 1.35$  mag (this is an assumption, we actually do not know  $\sigma$ ). Let  $M_0$  be the population mean visual absolute magnitude. A 95% confidence interval for  $M_0$  is

$$\left(\bar{x} - 1.96\frac{\sigma}{\sqrt{n}}, \bar{x} + 1.96\frac{\sigma}{\sqrt{n}}\right) = \left(-7.11 - 1.96\frac{1.35}{\sqrt{145}}, -7.11 + 1.96\frac{1.35}{\sqrt{145}}\right).$$

Thus,  $(-7.11 \mp 0.22)$  is a plausible range of values for  $M_0$ .

Warning: Don't bet your life that your 95% confidence interval has captured  $\mu$ ! There is a chance (5%) of it not capturing. Should we derive intervals with higher levels of confidence, 96%, 98%, 99%? Return to the tables of the  $N(0, 1)$  distribution and observe that  $P(-2.33 < Z < 2.33) = 0.98$ . Repeat the earlier arguments. Assuming that  $\sigma$  is known,

$$P\left(-2.33 < \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} < 2.33\right) = 0.98.$$

leading to a 98% confidence interval,

$$\left(\bar{X} - 2.33 \frac{\sigma}{\sqrt{n}}, \bar{X} + 2.33 \frac{\sigma}{\sqrt{n}}\right).$$

If  $\sigma$  is unknown then the method outlined above for getting confidence intervals does not work. A basic principle in statistics is: *Replace any unknown parameter with a good estimator*. Consider the LF data problem. We have a random sample  $X_1, \dots, X_n$  drawn from  $N(\mu, \sigma^2)$ . We want to construct confidence interval for  $\mu$  using the statistic  $\frac{\bar{X} - \mu}{S/\sqrt{n}}$ . To repeat the above method, we need the sampling distribution of this statistic. It is not normally distributed.

**The  $t$ -distribution:** If  $X_1, \dots, X_n$  is a random sample drawn from  $N(\mu, \sigma^2)$  then the statistic

$$T = \frac{\bar{X} - \mu}{S/\sqrt{n}}$$

has a  $t$ -distribution with  $n - 1$  degrees of freedom. Once this is granted, we construct confidence intervals as before. Suppose that  $n = 16$ , then see the tables of the  $t$ -distribution with 15 degrees of freedom.

$$P(-2.131 < T_{15} < 2.131) = 0.95.$$

Therefore

$$P\left(-2.131 < \frac{\bar{X} - \mu}{S/\sqrt{n}} < 2.131\right) = 0.95.$$

Thus, a 95% confidence interval for  $\mu$  is

$$\left(\bar{X} - 2.131 \frac{S}{\sqrt{n}}, \bar{X} + 2.131 \frac{S}{\sqrt{n}}\right).$$

For example, with  $n = 16$ ,  $\bar{x} = -7.1$  mag,  $s = 1.1$  mag, a 95% confidence interval for  $\mu$  is  $-7.1 \mp 0.586$ .

If you are curious about the  $t$ -density, here it is for  $p$  degrees of freedom.

$$f(t) = \frac{\Gamma((p+1)/2)}{\Gamma(p/2)} \frac{1}{\sqrt{p\pi}} \left(1 + \frac{t^2}{p}\right)^{-(p+1)/2} \quad -\infty < t < \infty.$$

**The  $\chi^2$ -distribution:** So far we have been considering confidence intervals for  $\mu$ . Let us now discuss confidence intervals for  $\sigma$  based on a random sample  $X_1, \dots, X_n$ , under normal model. We know  $S^2$  is an unbiased and consistent estimator of  $\sigma^2$ . What is the sampling distribution of  $S^2$ ? The statistic  $(n-1)S^2/\sigma^2$  has a *chi-squared*  $\chi^2$  distribution with  $n-1$  degrees of freedom. We now construct confidence intervals as before. Consult the tables of the  $\chi^2$  distribution. Find the percentage points, and solve the various inequalities for  $\sigma^2$ . Denote the percentage points by  $a$  and  $b$ .

$$P(a < \chi_{n-1}^2 < b) = 0.95.$$

We find  $a, b$  using tables of the  $\chi^2$  distribution. Usually, this is done by choosing  $a$  so that  $P(\chi_{n-1}^2 < a) = .025$  and  $P(\chi_{n-1}^2 > b) = .025$ . Solve for  $\sigma^2$  the inequalities:  $a < \frac{(n-1)S^2}{\sigma^2} < b$ . A 95% confidence interval for  $\sigma^2$  is

$$\left(\frac{(n-1)S^2}{b}, \frac{(n-1)S^2}{a}\right)$$

For example, if  $n = 16$ ,  $s = 1.2$  mag, percentage points from the  $\chi^2$  tables (with 15 degrees of freedom) are 6.262 and 27.49. Hence a 95% confidence interval for  $\sigma^2$  is

$$\left(\frac{15 \times (1.2)^2}{27.49}, \frac{15 \times (1.2)^2}{6.262}\right) = (0.786, 3.449).$$

If you are curious about the  $\chi^2$  density, here it is for  $p$  degrees of freedom.

$$f(x) = \left[2^{p/2}\Gamma(p/2)\right]^{-1} e^{-x/2} x^{p/2-1} \quad x > 0.$$

If we want a greater level of confidence, the confidence interval will, in general, be longer. The larger the sample size, the shorter will be the confidence interval. How do we choose  $n$ ? In our 95% confidence intervals for  $\mu$ ,

the term  $1.96\sigma/\sqrt{n}$  is called the **margin of error**. We choose  $n$  to have a desired margin of error. To have a margin of error of 0.01 mag, we choose  $n$  so that

$$\frac{1.96\sigma}{\sqrt{n}} = 0.01, \quad \text{that is, } n = \left(\frac{1.96\sigma}{0.01}\right)^2.$$

A very interesting question arises now. Could we get the above confidence interval for  $\mu$  only because we assumed a normal model? On the face of it this seems so, because we used the fact that a certain statistic is normal. There is indeed more to this construction. Here is a **modified Central Limit Theorem** that will help us. Let  $X_1, \dots, X_n$  be a random sample;  $\mu$  be the population mean;  $\bar{X}$  be the sample mean and  $S$  be the sample standard deviation. If  $n$  is large, then

$$\frac{\bar{X} - \mu}{S/\sqrt{n}} \approx N(0, 1).$$

In other words, for large values of  $n$ , the probability that the above statistic lies between  $a$  and  $b$  is same as the corresponding area under the standard normal curve. The conclusion does not depend on the population probability distribution of the  $X_i$ . As long as the population mean and variance are finite, this will hold. Instead of the exact sampling distribution used earlier, we can use this approximate distribution to construct confidence intervals. The resulting confidence intervals for  $\mu$  also do not depend on the population probability distribution. Several papers on LF for globular clusters have large sample sizes like 300, 1000, etc.

## 5. Testing of Hypotheses.

A LF researcher believes that  $M_0 = -7.7$  mag for the M31 globular clusters. The researcher collects data — using a *random sample* — from M31. A natural question: “Are the data strongly in support of the claim that  $M_0 = -7.7$  mag?”

A **statistical hypothesis** is a statement about the parameters of the population. A **statistical test of significance** is a procedure for comparing observed data with a hypothesis whose plausibility is to be assessed. The **null hypothesis** is the statement being tested, usually denoted by  $H_0$ . The **alternative hypothesis** is a competing statement, usually denoted by  $H_a$ .

In general, the alternative hypothesis is chosen as the statement for which there is likely to be supporting evidence. In the case of our M31 LF researcher, the null hypothesis is  $H_0: M_0 = -7.7$ . An alternative hypothesis is  $H_a: M_0 \neq -7.7$ . This is an example of a **two-sided** alternative hypothesis. If we have reasons to believe that  $M_0$  can not be above  $-7.7$ , then we should make the alternative hypothesis **one sided**, namely,  $H_a: M_0 < -7.7$ .

The basic idea in devising a test is the following. Based on the observations, we calculate a specially chosen informative statistic. See which of the hypotheses makes the observed value of this chosen statistic more plausible. First, some terminology is needed. A **test statistic** is a statistic that will be calculated from the observed data for testing procedure. This will measure the compatibility of  $H_0$  with the observed data. It will have a sampling distribution free of unknown parameters (under the null hypothesis). A **rejection rule** is a rule which specifies the values of the test statistic for which we reject  $H_0$ . Here is an illustration.

Example: A random sample of 64 measurements has mean  $\bar{x} = 5.2$  and standard deviation  $s = 1.1$ . Test the null hypothesis  $H_0 : \mu = 4.9$  against the alternative hypothesis  $H_a : \mu \neq 4.9$

1. The null and alternative hypotheses are  $H_0 : \mu = 4.9, \quad H_a : \mu \neq 4.9$ .
2. The test statistic is  $T = \frac{\bar{X}-4.9}{S/\sqrt{n}}$ .
3. The distribution of the test statistic  $T$ , under the assumption that  $H_0$  is valid, is  $\approx N(0, 1)$ .
4. The rejection rule: Reject  $H_0$  if  $|T| > 1.96$ , the upper 95 percentage point in the tables of the standard normal distribution. Otherwise, we *fail to reject*  $H_0$ .

This cutoff point 1.96 is also called a **critical value**. This choice of critical value results in a 5% **level of significance** of the test of hypotheses. This mean that that there is a 5% chance of our rejecting hypothesis  $H_0$ , when it is actually true.

5. Calculate the value of the test statistic. It is

$$\frac{\bar{x} - 4.9}{s/\sqrt{n}} = \frac{5.2 - 4.9}{1.1/\sqrt{64}} = 2.18$$

6. Decision: We reject  $H_0$ ; the calculated value of the test statistic exceeds the critical value, 1.96.

We report that the statistic is **significant**. There is a **statistically significant** difference between the population mean and the hypothesized value of 4.9.

7. The  $P$ -value of the test is the smallest significance level at which the statistic is significant.

## 6. Return to $\chi^2$ .

We briefly encountered  $\chi^2$  in discussing confidence intervals for  $\sigma^2$ . We now discuss a little more of this. This arises in both testing **goodness of fit**, and also in estimation.

We first start with testing problem. This is best explained with a discrete model. Suppose that you have a random variable  $X$  that takes  $r$  values  $a_1, \dots, a_r$ . Someone proposes a hypothesis that for each  $i$  the chance of value  $a_i$  is  $p_i$ . Here  $p_i > 0$  for all  $i$  and  $\sum p_i = 1$ . How do we test this? Make  $n$  independent observations of  $X$  and suppose that in your data the value  $a_i$  appears  $n_i$  times. Of course  $\sum n_i = n$ . If the hypothesis is correct, we *expect* to see the value  $a_i$  approximately  $np_i$  many times. So the discrepancy relative to our expectation is  $(n_i - np_i)^2/(np_i)$  and the total discrepancy is

$$\sum_1^r \frac{(n_i - np_i)^2}{np_i}$$

and this is named as the  $\chi^2$  value for the data. This statistic is called  $\chi^2$  statistic. It can be shown that for large  $n$ , this statistic indeed has a  $\chi^2$  distribution with  $(r - 1)$  degrees of freedom. This fact can be used to test whether the proposed hypothesis is plausible — large values of this statistic being not in favour of the hypothesis.

Now We turn to an important method of estimation. As in the earlier para, assume that  $X$  takes  $r$  values  $a_1, \dots, a_r$ . Let  $P(X = a_i) = p_i(\theta)$ , that is, the probability depends on a parameter  $\theta$ . Once  $\theta$  is found out, the value  $p_i$  is known. How do we estimate  $\theta$ ? Here is a way to do it, choose that value

of  $\theta$  which minimizes the discrepancy. In other words, choose that value of  $\theta$  for which

$$\chi^2(\theta) = \sum_1^r \frac{(n_i - np_i)^2}{np_i}$$

is minimum. Note that this is **not** a statistic, it depends on the parameter  $\theta$ . You use calculus, differentiate w.r.t.  $\theta$ , remember  $p_i$  are functions of  $\theta$ . You end up solving

$$\sum_1^r \left( \frac{n_i - np_i(\theta)}{p_i(\theta)} + \frac{(n_i - np_i(\theta))^2}{2np_i^2(\theta)} \right) \frac{dp_i(\theta)}{d\theta} = 0.$$

This is called the **minimum  $\chi^2$  method** of estimation. Unfortunately, the presence of  $\theta$  in the denominator makes things messy. So one uses the **modified minimum  $\chi^2$  method** where, one ignores the second term in the above equation.

If the model is continuous and not discrete, one groups the observations and proceeds. We shall not go into the details.

## 7. Method of Moments.

$X$ : Random variable with density function  $f(x; \theta_1, \theta_2)$

Parameters to be estimated:  $\theta_1, \theta_2$

Have a random sample:  $X_1, \dots, X_n$

1. Calculate the first two sample moments:

$$m_1 = \frac{1}{n} \sum_{i=1}^n X_i, \quad m_2 = \frac{1}{n} \sum_{i=1}^n X_i^2$$

2. Calculate  $E(X)$  and  $E(X^2)$ , the first two population moments:

$$E(X^k) = \int_{-\infty}^{\infty} x^k f(x; \theta_1, \theta_2) dx$$

The results are in terms of  $\theta_1$  and  $\theta_2$ .

3. Solve for  $\theta_1, \theta_2$  the simultaneous equations

$$E(X) = m_1, \quad E(X^2) = m_2$$

The solutions are the *method-of-moments estimators* of  $\theta_1, \theta_2$ .

Example: LF for globular clusters;  $X: N(\mu, \sigma^2)$

$$f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[ -\frac{(x - \mu)^2}{2\sigma^2} \right]$$

Parameters:  $\mu$  and  $\sigma^2$

Random sample:  $X_1, \dots, X_n$

1. The first two sample moments:

$$m_1 = \frac{1}{n} \sum_{i=1}^n X_i = \bar{X} \quad m_2 = \frac{1}{n} \sum_{i=1}^n X_i^2 = \frac{n-1}{n} S^2 + \bar{X}^2$$

2. The first two population moments:

$$E(X) = \int_{-\infty}^{\infty} x f(x; \mu, \sigma^2) dx = \mu$$

$$E(X^2) = \int_{-\infty}^{\infty} x^2 f(x; \mu, \sigma^2) dx = \mu^2 + \sigma^2$$

3. Solve:  $\hat{\mu} = m_1, \hat{\mu}^2 + \hat{\sigma}^2 = m_2$

Solution:  $\hat{\mu} = \bar{X}, \hat{\sigma}^2 = m_2 - m_1^2 = \frac{n-1}{n} S^2$

$\hat{\mu}$  is unbiased;  $\hat{\sigma}^2$  is not unbiased.

We shall discuss the Maximum likelihood method in the next lecture.

## 7. Truncation.

Sometimes we need to use truncated distributions for modeling. As an example, suppose that in the LF study, we believe that there is an absolute magnitude limit. Say, we believe that the magnitude can not be above  $M^*$ . Then we should not model the LF data with normal distribution. We should use the truncated normal.

$$f(x; \mu, \sigma^2) = \begin{cases} \frac{C}{\sqrt{2\pi}\sigma} \exp \left[ -\frac{(x-\mu)^2}{2\sigma^2} \right], & \text{if } x \leq M^* \\ 0, & \text{if } x > M^* \end{cases}$$

where the constant  $C$  is so chosen as to make the area under the curve unity. See Garcia-Munoz, et al. "The relative abundances of the elements silicon through nickel in the low energy galactic cosmic rays," In: Proc. Int'l. Cosmic Ray Conference, Plovdiv, Bulgaria, 1977, Conference Papers. Volume 1. Sofia, B'lgarska Akademiia na Naukite, 1978, p. 224-229.

As another example, consider, Protheroe, et al. "Interpretation of cosmic ray composition - The path length distribution," (1981, Ap J., 247). If our instruments can not detect rays with path length below a certain value, then our observations will not be a random sample from the exponential population. Rather, they would only be a sample from the truncated exponential, namely,

$$f(x; \theta_1, \theta_2) = \begin{cases} \theta_1^{-1} \exp[-(x - \theta_2)/\theta_1], & \text{if } x \geq \theta_2 \\ 0, & \text{if } x < \theta_2 \end{cases}$$

Here we have two parameters  $\theta_1 > 0$  and  $\theta_2 > 0$ .

# Chapter 5

## CORRELATION & REGRESSION

*Notes by Rajeeva Karandikar*

## Correlation and Regression

Rajeeva Karandikar

Chennai Mathematical Institute

### Some Background

#### Some expectations

Let  $X$  be a random variable. Then the expectation of  $X$  is called the *mean* of  $X$ . If  $X$  is a random variable with mean  $\mu$ , then the *variance* of  $X$  is defined by

$$\sigma^2 = \text{VAR}(X) = E(X - \mu)^2 = EX^2 - \mu^2$$

The *standard deviation* of  $X$  is the square root of the variance.

If  $X$  and  $Y$  are random variables with means  $\mu$  and  $\nu$ , then the *covariance*  $X$  and  $Y$  is defined by

$$\text{COV}(X, Y) = E(X - \mu)(Y - \nu) = EXY - \mu\nu$$

The *correlation coefficient*  $\rho(X, Y)$  of  $X$  and  $Y$  is defined by

$$\rho(X, Y) = \frac{\text{COV}(X, Y)}{\sqrt{\text{VAR}(X) \text{VAR}(Y)}}$$

Some properties of expectation are the following

$$E(aX + b) = aEX + b,$$

$$\text{VAR}(aX + b) = a^2 \text{VAR}(X)$$

$$E(aX + bY + c) = aEX + bEY + c$$

$$\text{VAR}(aX + bY + c) = a^2 \text{VAR}(X) + b^2 \text{VAR}(Y) + 2ab \text{COV}(X, Y)$$

### Random vectors, mean vectors and covariance matrix

Let  $Y_1, \dots, Y_n$  be random variables. Then

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$$

is a  $p$ -dimensional *random vector*. Then the *mean vector*  $\boldsymbol{\mu} = E\mathbf{Y}$  and *covariance matrix*  $\boldsymbol{\Sigma} = \text{Cov}(\mathbf{Y})$  are defined by

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix}, \boldsymbol{\Sigma} = \begin{pmatrix} \Sigma_{11} & \cdots & \Sigma_{1n} \\ \vdots & \ddots & \vdots \\ \Sigma_{n1} & \cdots & \Sigma_{nn} \end{pmatrix}$$

where

$$\begin{aligned} \mu_i &= EY_i, \quad \Sigma_{ii} = \text{VAR}(Y_i), \\ \Sigma_{ij} &= \text{COV}(Y_i, Y_j), \quad i \neq j \end{aligned}$$

Then it can be shown that

$$\begin{aligned} E(\mathbf{A}\mathbf{Y} + \mathbf{b}) &= \mathbf{A}E\mathbf{Y} + \mathbf{b}, \\ \text{Cov}(\mathbf{A}\mathbf{Y} + \mathbf{b}) &= \mathbf{A}\text{Cov}(\mathbf{Y})\mathbf{A}'. \end{aligned}$$

which is the basic result used in regression.

Note that the covariance matrix is a symmetric matrix. Further,

$$0 \leq \text{VAR}(\mathbf{a}'\mathbf{Y}) = \mathbf{a}'\text{Cov}(\mathbf{Y})\mathbf{a}$$

which implies that the covariance matrix is non-negative definite, which means there is a matrix  $\mathbf{B}$  such that

$$\mathbf{B}\mathbf{B}' = \text{Cov}(\mathbf{Y})$$

Such a matrix  $\mathbf{B}$  is called a *square root* of the covariance matrix. Actually there are several such matrices. One of the most useful and easy to find in computer software is the Cholesky square root which is a triangular matrix.

### The multivariate normal distribution

We say that an  $n$ -dimensional random vector  $\mathbf{Y}$  has multivariate normal distribution with mean vector  $\mu$  and covariance matrix  $\Sigma$  and write

$$\mathbf{Y} \sim N_n(\mu, \Sigma)$$

if  $\mathbf{Y}$  has joint probability density function (pdf)

$$f(\mathbf{y}) = (2\pi)^{-n/2} |\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{y} - \mu)' \Sigma^{-1} (\mathbf{y} - \mu)\right\}, \quad \forall \mathbf{y}$$

Note that this function has the two worst things in matrices, the determinant and the inverse of a matrix.

For this reason people often prefer to characterize the normal distribution by the moment generating function (mgf)

$$M(\mathbf{t}) = Ee^{\mathbf{Y}'\mathbf{t}} = \exp\left(\mu'\mathbf{t} + \frac{1}{2}\mathbf{t}'\Sigma\mathbf{t}\right)$$

Note that the mgf is essentially the Laplace transform of the density function which is  $M(-\mathbf{t})$ . If we were going to derive properties of multivariate normal distribution, we would use the mgf.

Three important properties of multivariate normal are the following

1. (basic fact about multivariate normal)  $\mathbf{Y} \sim N_n(\mu, \Sigma)$  implies

$$\mathbf{A}\mathbf{Y} + \mathbf{b} \sim N_q(\mathbf{A}\mu + \mathbf{b}, \mathbf{A}\Sigma\mathbf{A}')$$

2.  $\mathbf{Y} \sim N_n(\mu, \Sigma)$  implies  $Y_i \sim N_1(\mu_i, \Sigma_{ii})$ .
3. If  $U$  and  $V$  are jointly normally distributed, then  $\text{Cov}(U, V) = 0 \Rightarrow U$  and  $V$  are independent.

We now give a brief digression on how to simulate a multivariate normal. Let  $Z_1, \dots, Z_n$  be independent random variables,  $Z_i \sim N(0, 1)$ ,

$$\mathbf{Z} = \begin{pmatrix} Z_1 \\ \vdots \\ Z_n \end{pmatrix} \sim N_n(0, \mathbf{I})$$

Let

$$\mathbf{Y} = \mathbf{\Sigma}^{1/2}\mathbf{Z} + \mu \sim N(\mu, \mathbf{\Sigma})$$

by the basic result above when  $\mathbf{\Sigma}^{1/2}$  is a square root of the non-negative definite matrix  $\mathbf{\Sigma}$ . This shows that a multivariate normal distribution exists for any  $\mu$  and any non-negative definite matrix  $\mathbf{\Sigma}$ , and gives a pretty easy way to simulate it.

## Multiple linear regression

### The basic model

Let  $y = f(\mathbf{x})$  be a univariate function of several variables. The  $x$ 's are known as the *predictors* and the  $y$  is called the *response*. In simple linear regression we have one predictor and one response; in multiple linear regression we have several predictors and one response; and in multivariate linear regression we have several predictors and several responses. In this tutorial we will look at multiple linear regression with simple linear regression as a special case.

We assume that we have some data. Let  $Y_i$  be the response for the  $i^{\text{th}}$  data point and let  $\mathbf{x}_i$  be the  $p$ -dimensional (row vector) of the predictors for the  $i$ th data point,  $i = 1, \dots, n$ .

We assume that

$$Y_i = \mathbf{x}_i\beta + e_i.$$

Note that  $\beta$  is  $p \times 1$ . and is an unknown parameter.

For the regression model we assume that

$$e_i \sim N_1(0, \sigma^2), \text{ and the } e_i \text{ are independent.}$$

Note that  $\sigma^2$  is another parameter for this model.

We further assume that the predictors are linearly independent. Thus we could have the second predictor be the square of the first predictor, the third one the cube of the first one, etc, so this model includes polynomial regression.

We often write this model in matrices. Let

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}, \mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{pmatrix}, \mathbf{e} = \begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix}$$

so that  $\mathbf{Y}$  and  $\mathbf{e}$  are  $n \times 1$  and  $\mathbf{X}$  is  $n \times p$ . The assumed linear independence of the predictors implies that the columns of  $\mathbf{X}$  are linearly independent and hence  $\text{rank}(\mathbf{X}) = p$ . The normal model can be stated more compactly as

$$\mathbf{Y} = \mathbf{X}\beta + \mathbf{e}, \quad \mathbf{e} \sim N_n(0, \sigma^2 \mathbf{I})$$

or as

$$\mathbf{Y} \sim N_n(\mathbf{X}\beta, \sigma^2 \mathbf{I})$$

Therefore, using the formula for the multivariate normal density function, we see that the joint density if the observations is

$$\begin{aligned} f_{\beta, \sigma^2}(\mathbf{y}) &= (2\pi)^{-n/2} |\sigma^2 \mathbf{I}|^{-1/2} \exp\left\{-\frac{1}{2} (\mathbf{y} - \mathbf{X}\beta)' (\sigma^2 \mathbf{I})^{-1} (\mathbf{y} - \mathbf{X}\beta)\right\} \\ &= (2\pi)^{-n/2} (\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{X}\beta\|^2\right\} \end{aligned}$$

Therefore the likelihood for this model is

$$L_{\mathbf{Y}}(\beta, \sigma^2) = (2\pi)^{-n/2} (\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2} \|\mathbf{Y} - \mathbf{X}\beta\|^2\right\}$$

### Estimation of $\beta$

We first mention that the assumption on the  $\mathbf{X}$  matrix implies that  $\mathbf{X}'\mathbf{X}$  is invertible.

The ordinary least square (OLS) estimator of  $\beta$  is found by minimizing

$$q(\beta) = \sum (Y_i - \mathbf{x}_i\beta)^2 = \|\mathbf{Y} - \mathbf{X}\beta\|^2$$

The formula for the OLS estimator of  $\beta$  is

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{Y}$$

To see this note that

$$\nabla q(\beta) = 2\mathbf{X}'(\mathbf{Y} - \mathbf{X}\beta) = 2(\mathbf{X}'\mathbf{Y} - \mathbf{X}'\mathbf{X}\beta)$$

setting this equal to 0 we get the above formula for  $\hat{\beta}$ . For an algebraic derivation note that

$$\begin{aligned} q(\beta) &= \left\| \mathbf{Y} - \mathbf{X}\hat{\beta} \right\|^2 + \left\| \mathbf{X}\hat{\beta} - \mathbf{X}\beta \right\|^2 \\ &\geq \left\| \mathbf{Y} - \mathbf{X}\hat{\beta} \right\|^2 = q(\hat{\beta}) \end{aligned}$$

Although this is the formula we shall use for the OLS estimator, it is not how it is computed by most software package which solve the normal equations

$$\mathbf{X}'\mathbf{X}\hat{\beta} = \mathbf{X}'\mathbf{Y}$$

typically using the sweep algorithm.

Note that

$$\begin{aligned} E\hat{\beta} &= (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'E\mathbf{Y} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{X}\beta = \beta \\ \text{Cov}(\hat{\beta}) &= (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\sigma^2\mathbf{I}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} = \sigma^2\mathbf{M} \end{aligned}$$

Therefore

$$\hat{\beta} \sim N_p(\beta, \sigma^2\mathbf{M})$$

Therefore we note that the OLS,  $\hat{\beta}$ , is an unbiased estimator of  $\beta$  ( $E\hat{\beta} = \beta$ ) and the

$$\text{VAR}(\hat{\beta}_i) = \sigma^2 M_{ii}$$

We now give some further properties of the OLS estimator.

1. (Gauss-Markov) For the non-normal model the OLS estimator is the best linear unbiased estimator (BLUE), i.e., it has smaller variance than any other linear unbiased estimator.
2. For the normal model, the OLS is the best unbiased estimator i.e., has smaller variance than any other unbiased estimator
3. Typically, the OLS estimator is consistent, i.e.  $\hat{\beta} \rightarrow \beta$

### The unbiased estimator of $\sigma^2$

In regression we typically estimate  $\sigma^2$  by

$$\hat{\sigma}^2 = \left\| \mathbf{Y} - \mathbf{X}\hat{\beta} \right\|^2 / (n - p)$$

which is called the unbiased estimator of  $\sigma^2$ . we first state the distribution of  $\hat{\sigma}^2$ .

$$\frac{(n - p) \hat{\sigma}^2}{\sigma^2} \sim \chi_{n-p}^2 \text{ independently of } \hat{\beta}$$

We now give some properties of this estimator

1. For the general model  $\hat{\sigma}^2$  is unbiased
2. For the normal model  $\hat{\sigma}^2$  is the best unbiased estimator.
3.  $\hat{\sigma}^2$  is consistent

### The maximum likelihood estimator (MLE)

Looking at the likelihood above, we see that the OLS estimator maximizes the exponent so that  $\hat{\beta}$  is the MLE of  $\beta$ . To find the MLE of  $\sigma^2$  differentiate  $\log\left(L_Y\left(\hat{\beta}, \sigma^2\right)\right)$  with respect to  $\sigma$ , getting

$$\hat{\sigma}_{MLE}^2 = \frac{n - p}{n} \hat{\sigma}^2$$

Note that if

$$p/n = q$$

then

$$E\hat{\sigma}_{MLE}^2 = (1 - q) \sigma^2, \hat{\sigma}_{MLE}^2 \rightarrow (1 - q) \sigma^2$$

so the MLE is not unbiased and is not consistent unless  $p/n \rightarrow 0$ .

### Interval estimators and tests.

We first discuss inference about  $\beta_i$  the  $i$ th component of  $\beta$ . Note that  $\hat{\beta}_i$  the  $i$ th component of the OLS estimator is the estimator of  $\beta_i$ . Further

$$\text{VAR}(\hat{\beta}_i) = \sigma^2 M_{ii}$$

which implies that the standard error of  $\hat{\beta}_i$  is

$$\hat{\sigma}_{\hat{\beta}_i} = \hat{\sigma} \sqrt{M_{ii}}$$

Therefore we see that a  $1 - \alpha$  confidence interval for  $\beta_i$  is

$$\beta_i \in \hat{\beta}_i \pm t_{n-p}^{\alpha/2} \hat{\sigma}_{\hat{\beta}_i}.$$

To test the null hypothesis  $\beta_i = c$  against one and two-sided alternatives we use the t-statistic

$$t = \frac{\hat{\beta}_i - c}{\hat{\sigma}_{\hat{\beta}_i}} \sim t_{n-p}.$$

Now consider inference for  $\delta = \mathbf{a}'\beta$ , let

$$\hat{\delta} = \mathbf{a}'\hat{\beta} \sim N_1(\delta, \sigma^2 \mathbf{a}'\mathbf{M}\mathbf{a})$$

therefore we see that  $\hat{\delta}$  is the estimator of  $\delta$ , and

$$\text{VAR}(\hat{\delta}) = \sigma^2 \mathbf{a}'\mathbf{M}\mathbf{a}$$

so that the standard error of  $\hat{\delta}$  is

$$\hat{\sigma}_{\hat{\delta}} = \hat{\sigma} \sqrt{\mathbf{a}'\mathbf{M}\mathbf{a}}$$

and therefore the confidence interval for  $\delta$  is

$$\delta \in \hat{\delta} \pm t_{n-p}^{\alpha/2} \hat{\sigma}_{\hat{\delta}}$$

and the test statistic for testing  $\delta = c$  is given by

$$\frac{\hat{\delta} - c}{\hat{\sigma}_{\hat{\delta}}} \sim t_{n-p} \text{ under the null hypothesis}$$

There are tests and confidence regions for vector generalizations of these procedures.

Let  $\mathbf{x}_0$  be a row vector of predictors for a new response  $Y_0$ . Let  $\mu_0 = \mathbf{x}_0\beta = EY_0$ . The  $\hat{\mu}_0 = \mathbf{x}_0\hat{\beta}$  is the obvious estimator of  $\mu_0$  and

$$\text{VAR}(\hat{\mu}_0) = \sigma^2 \mathbf{x}_0 \mathbf{M} \mathbf{x}'_0 \Rightarrow \hat{\sigma}_{\hat{\mu}_0} = \hat{\sigma} \sqrt{\mathbf{x}_0 \mathbf{M} \mathbf{x}'_0}$$

and therefore a confidence interval for  $\mu_0$  is

$$\mu_0 \in \hat{\mu}_0 \pm t_{n-p}^{\alpha/2} \hat{\sigma}_{\hat{\mu}_0}$$

A  $1 - \alpha$  prediction interval for  $Y_0$  is an interval such that

$$P(a(\mathbf{Y}) \leq Y_0 \leq b(\mathbf{Y})) = 1 - \alpha$$

A  $1 - \alpha$  prediction interval for  $Y_0$  is

$$Y_0 \in \hat{\mu}_0 \pm t_{n-p}^{\alpha/2} \sqrt{\hat{\sigma}^2 + \hat{\sigma}_{\hat{\mu}_0}^2}$$

The derivation of this interval is based on the fact that

$$\text{VAR}(Y_0 - \hat{\mu}_0) = \sigma^2 + \sigma_{\hat{\mu}_0}^2$$

### The hat matrix

The hat matrix  $\mathbf{H}$  is defined as

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}$$

$\mathbf{H}$  is a symmetric idempotent matrix, i.e

$$\mathbf{H}' = \mathbf{H}, \mathbf{H}^2 = \mathbf{H}$$

Let  $\mu = \mathbf{X}\beta$ ,  $\hat{\mu} = \mathbf{X}\hat{\beta}$ . Then

$$\hat{\mu} = \mathbf{H}\mathbf{Y}$$

which is why  $\mathbf{H}$  is called the hat matrix. Now let

$$\mathbf{H}^\perp = \mathbf{I} - \mathbf{H}$$

then  $\mathbf{H}^\perp$  is also a symmetric idempotent matrix which is orthogonal to  $\mathbf{H}$ , i.e

$$\mathbf{H}'\mathbf{H}^\perp = \mathbf{0}$$

Then

$$(n-p)\hat{\sigma}^2 = \|\mathbf{H}^\perp\mathbf{Y}\|^2$$

Note that

$$\mathbf{Y} = \mathbf{H}\mathbf{Y} + \mathbf{H}^\perp\mathbf{Y}$$

We think of  $\mathbf{H}\mathbf{Y}$  as having information about the signal  $\mu$  and  $\mathbf{H}^\perp\mathbf{Y}$  as having information about the noise  $Y - \mu$ . For the rest of this talk, we shall use  $\mathbf{H}$  for these matrices

### $R^2$ , adjusted $R^2$ and predictive $R^2$

Let

$$T^2 = \sum (Y_i - \bar{Y})^2, \quad S^2 = \|\mathbf{Y} - \mathbf{X}\hat{\beta}\|^2$$

be the numerators of the variance estimators for the regression model and the intercept only model. We think of these as measuring the "variation" under these two models. Then the coefficient of determination  $R^2$  is defined by

$$R^2 = \frac{T^2 - S^2}{T^2}$$

Note that

$$0 \leq R^2 \leq 1$$

Note that  $T^2 - S^2$  is the amount of variation in the regression model which has been explained by including the extra predictors of the regression model and  $R^2$  is the proportion of the variation left in the intercept only model which has been explained by including the additional predictors.

Note that

$$R^2 = \frac{\frac{T^2}{n} - \frac{S^2}{n}}{\frac{T^2}{n}}$$

which suggests that this might be improved by substituting unbiased estimator for the MLE's getting adjusted  $R^2$

$$R_a^2 = \frac{\frac{T^2}{n-1} - \frac{S^2}{n-p}}{\frac{T^2}{n-1}} = 1 - \frac{n-1}{n-p} (I - R^2)$$

Both  $R^2$  and adjusted  $R^2$  suffer from the fact that the fit is being evaluated with the same data used to compute it and therefore the fit looks better than it is. A better procedure is based on cross-validation. Suppose we delete the  $i$ th observation and compute  $\hat{\beta}_{-i}$  the OLS estimator of  $\beta$  without the  $i$ th observation. We do this for all  $i$ . We also compute  $\bar{Y}_{-i}$

$$\bar{Y}_{-i} = \sum_{j \neq i} Y_j / (n - 1)$$

the sample mean of the  $Y_i$  without the  $i$ th one. Then let

$$T_p^2 = \sum (Y_i - \bar{Y}_{-i})^2 = \frac{nT^2}{n - 1}$$

$$S_p^2 = \sum (Y_i - \mathbf{x}_i \hat{\beta}_{-i})^2 = \sum \left( \frac{Y_i - \mathbf{x}_i \hat{\beta}}{1 - H_{ii}} \right)^2$$

(where  $H_{ii}$  is the  $i$ th diagonal of the hat matrix).

Then predictive  $R^2$  is defined as

$$R_p^2 = \frac{T_p^2 - S_p^2}{T_p^2}$$

Predictive  $R^2$  computes the fit to the  $i$ th observation without using that observation and is therefore a better measure of the fit of the model than  $R^2$  or adjusted  $R^2$ .

## Diagnostics

### Residuals

Most of the assumptions in regression follow from

$$\mathbf{e} = \begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix} = \mathbf{Y} - \mathbf{X}\beta \sim N_n(\mathbf{0}, \sigma^2\mathbf{I})$$

To check these assumptions we look at residuals. The ordinary residuals are

$$\hat{\mathbf{e}} = \begin{pmatrix} \hat{e}_1 \\ \vdots \\ \hat{e}_n \end{pmatrix} = \mathbf{Y} - \mathbf{X}\hat{\beta} =$$

$$(\mathbf{I} - \mathbf{H})\mathbf{Y} \sim N_n(0, \sigma^2(\mathbf{I} - \mathbf{H}))$$

Note that the  $e_i$  are assumed to have equal variances, but even if all the assumptions are met

$$\text{VAR}(\hat{e}_i) = \sigma^2(1 - H_{ii})$$

are different. For this reason, the residuals are often standardized getting the standardized residuals

$$\hat{e}_{is} = \frac{\hat{e}_i}{\hat{\sigma}\sqrt{1 - H_{ii}}}$$

which if the assumptions are met should have constant variance about 1.

Because of the unequal variances, the ordinary residuals can be misleading, so we always look at the standardized residuals. Many other types of residuals have been suggested, e.g. delete one residual and t residual but they seem to look just like standardized residuals and so it does not seem necessary to look at any other residuals but standardized residuals. Just don't use ordinary residuals.

The assumption on the errors is really 4 assumptions

1.  $Ee_i = 0$ . This means we have included enough terms in the model. If it is not satisfied, it can often be corrected by including more terms in the model. This is often

---

a tough assumption to check with residuals, since it can be shown that the average of the residuals is always 0, even if this assumption is violated. One situation where residuals can be useful is in polynomial regression on a variable  $x$ . In that case if we plot the residuals against  $x$ , and if we have too few terms, we should see a pattern.

2.  $\text{VAR}(e_i)$  is constant. This is the most important assumption and is often violated. One way to use residuals to check this assumption is to make a residual vs. fits plot. For example, if we see a fanning pattern with large residuals vs. large fits, this means the variance is increasing with the mean. If we see this it is often remedied by a log transformation on the  $Y_i$ . Another way to go is to use weighted least squares.
3. The  $e_i$  are independent. This is another important assumption which is hard to check with residuals. If it is not true, we can model the correlation between the observations using time series methods or repeated measures or generalized least squares.
4. The  $e_i$  are normally distributed. This is the least important assumption. For moderate sample sizes it has been shown that that regression is robust against the normal assumption. To use residuals to check this assumption, look at a normal scores plot of the (standardized) residuals. It should look like a straight line. If this assumption is not met, you can transform to achieve normality, you can use an M-estimator, an R-estimator or some other less sensitive estimator than OLS or you can ignore it.

One other use for residual is for looking for outliers, points whose observations seem incorrect. One rule is that an observation is an outlier if its absolute standardized residual is greater than 3. Some data analysis programs automatically eliminate all outliers from the data.

One (true) story that suggests that this is not a good idea has to do with the hole in the ozone, which was not discovered by satellite (as it should have been), because the data analysis programs used eliminated all outliers and so eliminated the data for the hole in the ozone. It was discovered from the ground much later than it would have been discovered by satellite if the data had not been cleaned.

We should look carefully at the outliers and think about them before eliminating them. We often do separate analyses on the outliers and learn things we could not learn from the clean data. Basically, before you eliminate an outlier, you try to decide if it is a mistake or

an unusual data point. If it is a mistake, eliminate it, if it is an unusual data point then try to learn from it.

### Influence

Often the values for the predictors for one observation are quite far from the other observations which leads to that observation having a large influence on the regression line. For example in a simple regression, we might have most of the observations with predictor about 10 and one observation with predictor  $10^{10}$ . Then the regression line will basically connect the one extreme observation with the middle of the cloud of other points, so the response associated with the extreme point will essentially determine the regression line.

The leverage of the  $i$ th observation is defined as  $H_{ii}$ , the  $i$ th diagonal of the hat matrix. The reason for this definition is that if  $\mu = \mathbf{X}\beta$ , then

$$\hat{\mu} = \mathbf{H}\mathbf{Y}$$

so that the  $i$ th diagonal element of the hat matrix is the coefficient of the  $i$ th observation in its estimated mean. If this coefficient is large, then the  $i$ th observation has a large influence on its estimated mean and if the coefficient is small, then the  $i$ th observation has little influence on its estimated mean.

Using the fact that  $\mathbf{H}$  and  $\mathbf{I} - \mathbf{H}$  are idempotent and hence non-negative definite, we can show that

$$0 \leq H_{ii} \leq 1$$

so an observation is influential if the influence near 1 and not if it near 0. Note also that

$$\sum h_{ii} = \text{tr}\mathbf{H} = \text{tr}\left(\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\right) = \text{tr}\left((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\right) = \text{tr}\mathbf{I}_p = p$$

so that the average leverage is

$$\bar{H} = \sum H_{ii}/n = p/n$$

One rule of thumb which is often used is that an observation has high influence if

$$H_{ii} > \frac{3p}{n}.$$

If we find a point which has high influence, we should think about whether we should eliminate it. Sometimes such a point has an incorrect number for the predictor and could

really mess up the analysis. Sometimes, however, it is a true point and may be the most important point in fitting the regression.

### Multicollinearity

One other critical assumption in regression is that the predictors linearly independent so that  $\mathbf{X}'\mathbf{X}$  is invertible. Typically this assumption is satisfied. Often though one predictor is nearly a linear combination of some others. This is called multicollinearity. When this happens the  $\text{VAR}(\hat{\beta}_i)$  are quite large and it is not possible to draw good inference about the  $\beta_i$ . So we try to detect multicollinearity and eliminate it.

The main tool for detecting multicollinearity is the variance inflation factor (VIF) for each predictor which we now describe. Recall that

$$\text{VAR}(\hat{\beta}_i) = \sigma^2 M_{ii}$$

We say that the predictors are orthogonal if for any two columns of the  $\mathbf{X}$  matrix

$$\mathbf{X}_j' \mathbf{X}_k = 0, \quad \forall j \neq k$$

We note that orthogonality is as far from multicollinearity as possible. We note if the predictors are orthogonal then

$$\text{VAR}_O(\hat{\beta}_i) = \sigma^2 / \|\mathbf{X}_i\|^2$$

The VIF for the  $i$ th predictor is defined as

$$\frac{\text{VAR}(\hat{\beta}_i)}{\text{VAR}_O(\hat{\beta}_i)}$$

so the VIF tells how much the variance of  $\hat{\beta}_i$  has been inflated due to the multicollinearity. If it is large then something should probably be done to eliminate the multicollinearity. If it they are all near 1 then there is no multicollinearity.

There is another interpretation for VIF's which is pretty interesting. Suppose we regressed the  $j$ th predictor on the other predictors and let  $R_j^2$  be  $R^2$  from this fit. Then, it can be shown that

$$VIF_j = \frac{1}{1 - R_j^2}$$

so that if the  $j$ th predictor is nearly a linear combination of the others then  $R_j^2$  should be near 1 and the  $VIF_j$  should be large.

Typically in a polynomial regression model fit in the obvious way there is a great deal of colinearity. One method which is often used to eliminate the colinearity in this situation is to center the  $x$  term for the linear term, then square the centered  $x$ 's for the quadratic term, etc.

### Model Selection

The last regression topic we'll talk about is how to choose which predictors to include in the model. We say we have overfit the model if we have too many terms and underfit it if we have too few terms.

Some naive approaches don't work, such as choosing the model with the largest  $R^2$ . It can be shown that  $R^2$  always increases when variables are added to the model and we end up by including all the predictors in the model which is usually extreme overfitting. Maximizing adjusted  $R^2$  is a little better, but still overfits. Maximizing predictive  $R^2$  seems to work reasonably well.

Another approach which is often used is to minimize Mallows's  $C_p$ , which we now describe. Let

$$Q = \frac{E \|\hat{\mu} - \mu\|^2}{\sigma^2} = p + \frac{\hat{\mu}(\mathbf{I} - \mathbf{H})\hat{\mu}}{\sigma^2}$$

Our goal is to find a model which minimizes  $Q$ . It can be shown that an unbiased estimator of  $Q$  is

$$\hat{Q} = \frac{(n-p)\hat{\sigma}^2}{\sigma^2} - n + 2p.$$

$\hat{Q}$  is called Mallows  $C_p$ . We can already compute all of this except  $\sigma^2$ , which we estimate by regressing on all the possible predictors. Then we look at all the possible models and find the one which minimizes  $\hat{Q}$ . The main problem with this approach is the estimation of  $\sigma^2$ . Sometimes there are more potential predictors than there are observations so it is not possible to regress on all possible predictors. Also it seems bothersome that if we add more predictors to the model, we would change  $\sigma^2$ . It seems that the criterion for a particular model should depend only on that model not some larger model.

For these reasons, emphasis for model selection has shifted to penalized likelihood criteria. Note that for this model, the maximized likelihood is

$$\begin{aligned} L_Y \left( \hat{\beta}, \hat{\sigma}_{MLE}^2 \right) &= (2\pi)^{-\frac{n}{2}} \left( \hat{\sigma}_{MLE}^2 \right)^{-\frac{n}{2}} \exp \left\{ -\frac{\left\| \mathbf{Y} - \mathbf{X} \hat{\beta} \right\|^2}{2 \hat{\sigma}_{MLE}^2} \right\} \\ &= (2\pi)^{-\frac{n}{2}} \left( \hat{\sigma}_{MLE}^2 \right)^{-\frac{n}{2}} \exp \left\{ -\frac{n}{2} \right\} \end{aligned}$$

A naive approach would be to choose the model which maximizes the maximized likelihood, but that also just picks out the model with all the predictors and overfits.

The first penalized likelihood criterion suggested was the Akaike Information Criterion (AIC), which minimizes

$$AIC = -2 \log \left( L_Y \left( \hat{\beta}, \hat{\sigma}_{MLE}^2 \right) \right) + 2(p + 1)$$

This criterion is based on Kullback-Liebler information. Unfortunately, it is known to overfit.

These lecture notes are essentially taken from the notes prepared by Steven F. Arnold Professor of Statistics, Penn State University for these lectures in 2006.

# Chapter 6

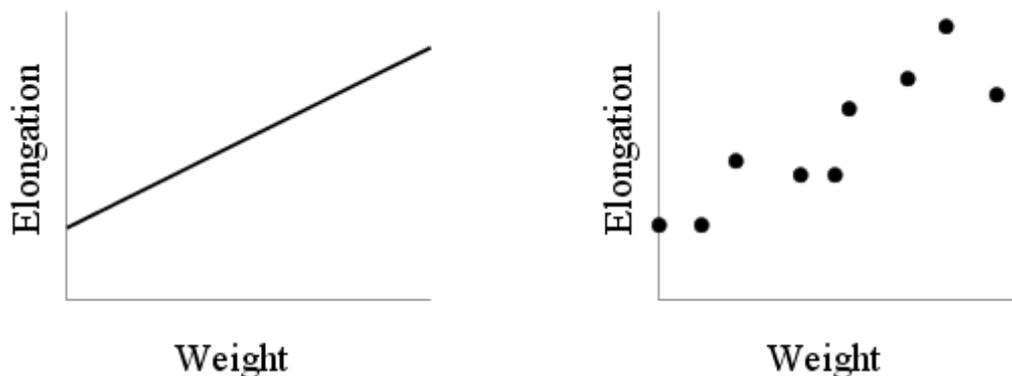
## REGRESSION WITH R

*Notes by Arnab Chakraborty*

# Regression Analysis

## Introduction

In every walk of science we come across variables related to one another. Mathematical formulations of such relations occupy an important place in scientific research. When only two variables are involved we often plot a curve to show functional relationships as in the left hand figure below which depicts the relation between the length of a steel wire and the weight suspended from it.



However, what the experimenter sees in the laboratory is not this neat line. He sees a bunch of points as shown in the right hand figure. One idealizes these points to get the straight line. This process of extracting an ideal (and, desirably, simple) relation between variables based on observed data is called **regression analysis**.

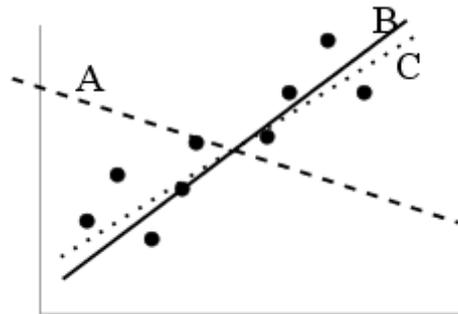
Often the relationship between two variables has a causal flavor. For example, in our example, we like to think of the weight as *causing* the elongation. Thus, weight is the variable that is under direct control of the experimenter. So we call it the **explanatory** variable. The other variable (elongation) is called the **response**. If we have a single explanatory variable and a single response (as here), then we have a **bivariate** regression problem. A **multivariate** regression problem has more than one explanatory variable, but (usually) a single response. We shall talk about bivariate regression first.

## Bivariate regression

A typical regression analysis proceeds in four steps:

1. First we postulate a *form* of the relation, like a straight line or a parabola or an exponential curve. The form involves unknown numbers to be determined. For instance, a straight line has equation  $y = a + bx$ , where  $a, b$  are unknown numbers to be determined. Typically the form of the relation is obtained by theoretical considerations and/or looking at the scatterplot.
2. Next we have to decide upon an objective criterion of

goodness of fit. For instance, in the plot below, we can see that line *A* is a bad fit. But both *B* and *C* appear equally good to the *naked eye*. An objective criterion for goodness of fit is needed to choose one over the other.



3. Once we have chosen the form as well as the criterion, it is a matter of routine computation to find a relation of our chosen form that fits the data best. Statistics textbooks spend many pages elaborating on this step. However, for us it is just a matter of invoking R.
4. Last but not the least, we have to check whether the ``best" relation is indeed what we expected. (Common sense above routine computation!)

Let us load the Hipparcos data set and extract the Hyades stars. We shall do this by just invoking the script that we had created earlier.

```
source("hyad.r")
attach(hip[HyadFilter,])
```

We shall now define a luminosity variable `logL`

```
logL = (15 - Vmag - 5 * log10(Plx)) / 2.5
```

Let us make a scatterplot of this variable against `B.V`

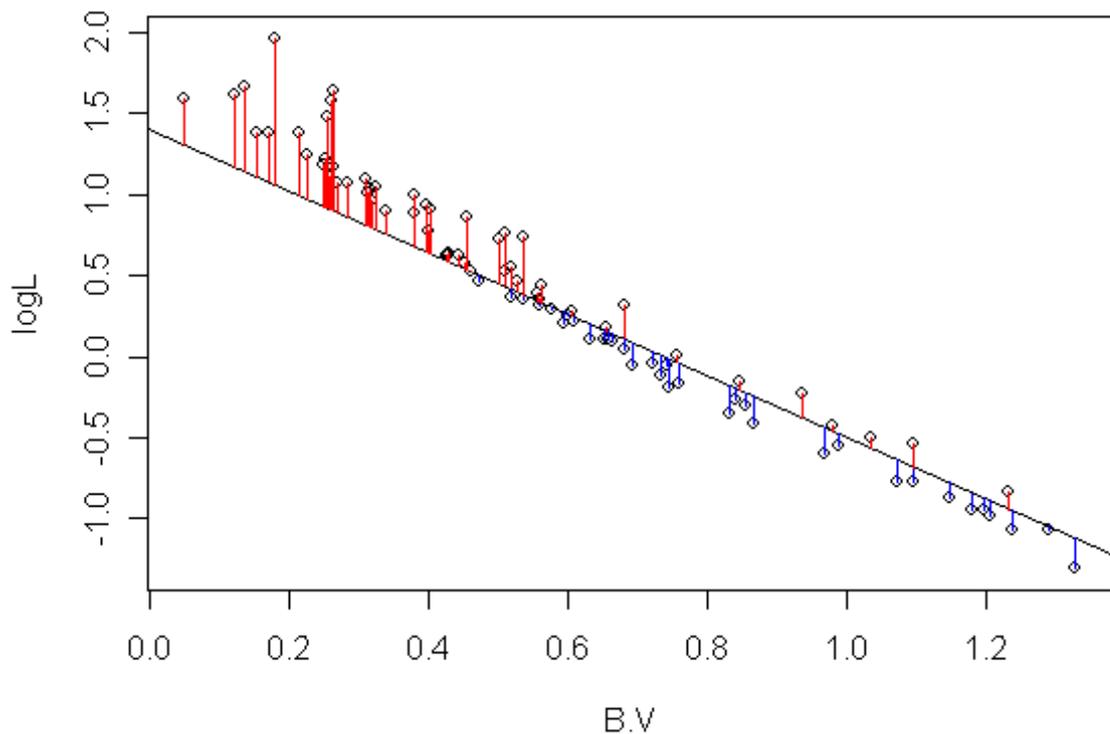
```
plot(B.V, logL)
```

Well, the plot seems to indicate a relation of the form

$$\log L = a + b B.V$$

So we have finished the first step.

Next we have to decide upon an objective criterion for goodness of fit. We shall use the most popular choice: least squares. To understand the meaning of this consider the following graph that shows a line drawn over the scatterplot.

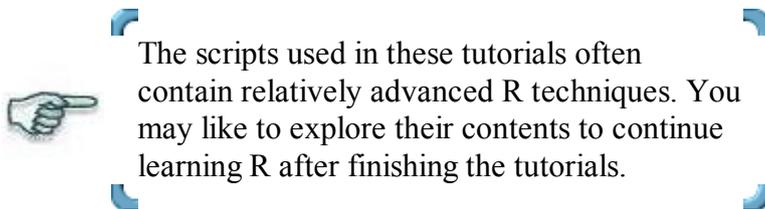


### The concept of Least Squares

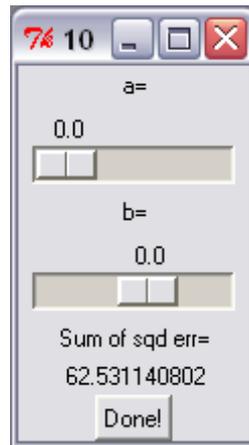
From each of the data points we have drawn a vertical to the line. The length of these verticals measure how much the actual value of `logL` differs from that predicted by the line. In Least Squares method we try to choose a line that minimizes the sum of squares of these errors.

**Exercise:** To get an idea of this run the script

```
|source("lsq.r")
```



The script will open a small window like this:



### Move the sliders to change $a, b$

Here you can interactively play with  $a$  and  $b$  to minimize the amount of error.

In the third step you will invoke the `lm` function of R to find the best values of  $a$  and  $b$  as follows.

```
bestfit = lm(logL ~ B.V)
```

Notice the somewhat weird notation `logL ~ B.V`. This means the formula

$$\log L = a + b B.V$$

In fact, if we write `y ~ x1+x2+...xk` then R interprets this as

$$y = a_0 + a_1 x_1 + \dots + a_k x_k,$$

where  $a_0, \dots, a_k$  are the parameters to be determined.

```
plot(B.V, logL)
abline(bestfit)
bestfit
```

This finishes the third step. The final step is to check how good the fit is. Remember that a fundamental aim of regression is to be able to predict the value of the response for a given value of the explanatory variable. So we must make sure that we are able to make good predictions and that we cannot improve it any further.

The very first thing to do is to plot the data and overlay the fitted line on it, as we have already done. The points should be close to the line *without any pattern*. The lack of pattern is important, because if there is some pattern then we can possibly improve the fit by taking this into account. In our example there *is* a pattern: the points in the middle are mostly below the line, while the points near the extremes are above. This suggests a slight curvature. So we may be better off with a quadratic fit.

The second thing to do is to plot the **residuals**. These are the the actual value of the response variables minus the fitted value. In terms of the least squares plot shown earlier these are the lengths of the vertical lines representing errors. For points above the line (red verticals) the sign is negative, while for the blue verticals the sign is positive.

The residuals are computed by the **lm** function automatically.

```
res = bestfit$resid
plot(res)
abline(h=0)
```

Ideally the points should all be scattered equally around the zero line. But here the points below the line look more densely spaced.

Next we should plot the residuals against the explanatory variable.

```
plot(B.V, res)
abline(h=0)
```

The clear U pattern is a most decisive demonstration that our best fit can possibly be improved further if we fit a parabola instead of a straight line.

Now that we are a bit wiser after our first attempt at regression analysis of the Hyades data set we should go back to step 1 and choose the parabolic form

$$\log L = a + b B.V + c B.V^2.$$

This form has three unknown parameters to be determined:  $a, b$  and  $c$ . Again we come to step 2. We shall still choose the least squares method. The third step is almost as before

```
bestfit2 = lm(logL ~ B.V + I(B.V^2))
```

The only unexpected piece in the above line is the **I**. This admittedly awkward symbol means the  $B.V^2$  should be treated *as is*. (What happens without this **I** is somewhat strange and its explanation is beyond the present scope.)

```
bestfit2
summary(bestfit2)
```

Now let us look at the fitted line. However, unlike the straight line case, here we do not have any ready-made function like **abline**. We shall need to draw the curve directly using the fitted values. Here is the first attempt:

```
plot(B.V, logL)
lines(B.V, bestfit2$fit)
```

Oops! We did not want this mess! Actually, what the `lines` function does is this: it plots all supplied the points and joins them *in that order*. Since the values of `B.v` are not sorted from small to large, the lines get drawn haphazardly. To cure the problem we need to sort `B.v` and order the fitted values accordingly. This is pretty easy in R:

```
ord = order(B.V)
plot(B.V, logL)
lines(B.V[ord], bestfit2$fit[ord])
```

This line does seem to be a better fit than the straight line that we fitted earlier. Let us make the residual plot. But this time we shall use a built-in feature of R.

```
plot(bestfit2, 3)
```

In fact, R can automatically make 4 different plots to assess the performance of the fit. The 3 in the command asks R to produce the 3rd of these. The others are somewhat more advanced in nature. There are three points to note here.

1. the vertical axis shows the square root of something called **standardized residual**. These are obtained by massaging the residuals that we were working with. The extra massaging makes the residuals more "comparable" (somewhat like making the denominators equal before comparing two fractions!)
2. the horizontal axis shows the fitted values.
3. the wavy red line tries to draw our attention to the general pattern of the points. Ideally it should be a horizontal line.
4. There are three points that are rather too far from the zero line. These are potential trouble-makers (outliers) and R has labeled them with their case numbers.

To see the values for the point labelled 54 you may use

```
B.V[54]
logL[54]
```

Typically it is a good idea to take a careful look at these values to make sure there is nothing wrong with them (typos etc). Also, these stars may indeed be special. Some one analyzing the data about the ozonosphere had stumbled across such outliers that turned out to be the holes in the ozone layer!

## Comparing models

Sometimes we have two competing fits for the same data. For example, we had the straight line as well as the quadratic line. How do we compare between them? "Choose the one that goes closer through the points" might look like a tempting answer. But unfortunately there is a snag. While we want the fit to pass close to

the points, we also want to keep the equation of the fit as simple as possible! (This is not merely out of an ascetic love for simplicity, there are also deep statistical implications that we cannot discuss here.) Typically, these two criteria: *simplicity* and *proximity to all the points* act in opposite directions. You gain one only by sacrificing the other. There are criteria that seek to strike a balance between the twain. We shall discuss two of them here: **Akaike's Information Criterion (AIC)** and **Bayesian Information Criterion (BIC)**, both being computed using a function called **AIC** in R:

```
AIC(bestfit)
AIC(bestfit2)
```

The smaller the AIC the better is the fit. So here the quadratic fit is indeed better. However, AIC is known to overfit: it has a bias to favour more complicated models. The BIC seeks to correct this bias:

```
n = length(B.V)
AIC(bestfit,k=log(n)) #This computes BIC
AIC(bestfit2,k=log(n))
```

Well, BIC also confirms the superiority of the quadratic. Incidentally, both the AIC and the BIC are merely dumb mathematical formulas.



It is always important to use domain knowledge and common sense to check the fit.

In particular, here one must remove those three outliers and redo the computation just to see if there is a better fit.

## Robustness

Outliers are a constant source of headache for the statistician, and astronomical data sets are well known for a high content of outliers (after all, the celestial bodies are under no obligation to follow petty statistical models!) In fact, detecting outliers is sometimes the most fruitful outcome of an analysis. In order to detect them we need statistical procedures that are dominated by the general trend of the data (so that the outliers show up in contrast!) A statistical process that is dominated only by the majority of the points (and not by the outliers) is called **robust**. The second step in a regression analysis (where we choose the criterion) determines how robust our analysis will be. The least squares criterion that we have used so far is not at all robust.

We shall now compare this with a more robust choice that is implemented in the function **lqs** of the MASS package in R.



A **package** in R is like an add-on. It is a set of functions and data sets (plus online helps on these) that may reside in your machine but are not automatically loaded into R. You have to

load them with the function **library** like

```
library(MASS) #MASS is the name of the package
```

The package mechanism is the main way R grows as more and more people all over the world write and contribute packages to R. It is possible to make R download and install necessary packages from the internet, as we shall see later.

To keep the exposition simple we shall again fit a straight line.

```
library(MASS)
fit = lm(logL ~ B.V)
robfit = lqs(logL ~ B.V)
plot(B.V, logL)
abline(fit, col="red")
abline(robfit, col="blue")
```

Well, the lines are different. But just by looking at this plot there is not much to choose the robust (blue) one over the least square (red). Indeed, the least squares line seems a slightly better fit. In order to appreciate the benefit of robustness we have to run the following script. Here you will again start with the same two lines, but now you can *add one new point* to the plot by clicking with your mouse. The plot will automatically update the two lines.

```
source("robust.r")
```

You'll see that a single extra point (the outlier) causes the least squares (red) line swing more than the robust (blue) line. In fact, the blue line does not seem to move at all!

Right click on the plot and select "Stop" from the pop-up menu to come out. You may also need to click on the little red STOP button in R toolbar.

### Parametric vs. nonparametric

Here we shall take consider the first step (choice of the form) once more. The choice of the form, as we have already pointed out, depends on the underlying theory and the general pattern of the points. There may be situations, however, where it is difficult to come up with a simple form. Then it is common to use **non-parametric regression** which internally uses a complicated form with tremendous flexibility. We shall demonstrate the use of one such method called LOWESS which is implemented as the function **lowess** in R.

```
plot(B.V, logL)
fit = lowess(logL~B.V)
lines(fit)
```

A newer variant called **loess** produces very similar result. However, drawing the line is slightly more messy here.

```
fit2 = loess(logL~B.V)
ord = order(B.V)
lines(B.V[ord], fit2$fitted[ord], col="red")
```

## Multiple regression

So far we have been working with bivariate regression, where we have one response and one explanatory variable. In multiple regression we work with a single response and at least two explanatory variables. The same functions (**lm**, **lqs**, **loess** etc) handle multiple regression in R. We shall not discuss much of multiple regression owing to its fundamental similarity with the bivariate case. We shall just show a single example introducing **RA** and **DE** as explanatory variables as well as **B.V**.

```
fit = lm(logL~B.V+RA+DE)
summary(fit)
```

We shall not go any deeper beyond pointing out that the absence of the asterisks in the **RA** and **DE** lines mean these extra explanatory variables are useless here.

## A word of wisdom

Our exposition so far has been strictly from the users' perspective with the focus being on commands to achieve things, and not how these commands are internally executed. Sometimes the same problem can be presented in different ways that are all essentially the same to the user, but quite different for the underlying mathematics. One such example is that R prefers the explanatory variables to be *centered*, that is, spread symmetrically around 0. This enhances the stability of the formulas used. So instead of

```
lm(logL ~ B.V)
```

we should do

```
B.V1 = B.V - mean(B.V)
lm(logL ~ B.V1)
```



Always center the explanatory variables before performing regression analysis. This does not apply to the response variable.

## *k*-nearest neighbors

Imagine that you are given a data set of average parent heights and

their adult sons' heights, like this (ignore the \*'s for the time being):

Parent	Son
5.5	5.9*
5.4	5.3*
5.7	5.9
5.1	5.3*
6.0	5.8
5.5	5.2*
6.1	6.0

A prospective couple come to you, report that their average height is 5.3 feet, and want you to predict (based on this data set) the height their son would attain at adulthood. How should you proceed?

A pretty simple method is this: look at the cases in your data set with average parents' height near 5.3. Say, we consider the 3 nearest cases. These are marked with \*'s in the above data set. Well, there are 4 cases, and not 3. This is because there is a **tie** (three cases 5.1, 5.5, 5.5 that are at same distance from 5.3).

Now just report the average of the sons' heights for these cases.

Well, that is all there is to it in 3-NN regression (NN = Nearest Neighbor)! Let us implement this in R. First the data set

```
parent = c(5.5, 5.4, 5.7, 5.1, 6.0, 5.5, 6.1)
son = c(5.9, 5.3, 5.9, 5.3, 5.8, 5.2, 6.0)
```

Now the new case:

```
newParent = 5.3
```

Find the distances of the parents' heights from this new height:

```
d = abs(parent - newParent)
```

Rank these:

```
rnk = rank(d, tie="min")
rnk
```

Notice the `tie="min"` option. This allows the three joint seconds to be each given rank 2. We are not using **order** here (as it does not handle ties gracefully). Now identify the 3 nearest cases (or more in case of ties).

```
nn = (rnk <= 3)
nn
```

Now it is just a matter of taking averages.

```
newSon = mean(son[nn])
```

```
newSon
```

It is instructive to write a function that performs the above computation.

```
knnRegr = function(x, y, newX, k) {
  d = abs(x-newX)
  rnk = rank(d, tie="min")
  mean(y[rnk<=k])
}
```

Now we can use it like

```
newSon = knnRegr(parent, son, newParent, 3)
```

**Exercise:** You shall apply the function that we have just written to the Hipparcos data set.

```
plot(RA, pmRA)
```

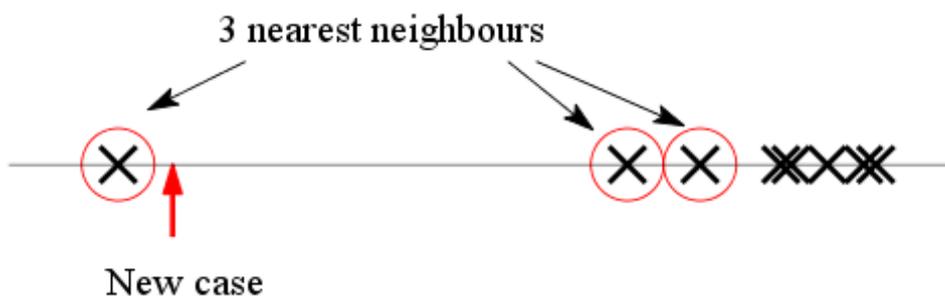
We want to explore the relation between `RA` and `pmRA` using the  $k$ -nearest neighbour method. Our aim is to predict the value `pmRA` based on a new value of `RA`:

```
newRA = 90.
```

Use the `knnRegr` function to do this with  $k=13$ .

## Nadaraya-Watson regression

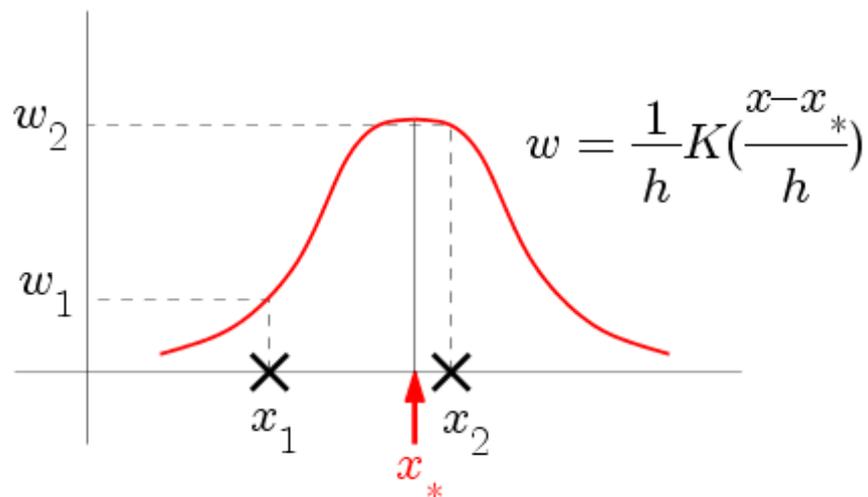
While the  $k$ -nearest neighbor method is simple and reasonable it is nevertheless open to one objection that we illustrate now. Suppose that we show the  $x$ -values along a number line using crosses as below. The new  $x$ -value is shown with an arrow. The 3 nearest neighbors are circled.



### Are the 3 NN equally relevant?

The  $k$ -NN method now requires us to simply average the  $y$ -values of the circled cases. But since two of the points are rather far away from the arrow, shouldn't a weighted average be a better method, where remote points get low weights?

This is precisely the idea behind the Nadaraya-Watson regression. Here we average over *all* the cases (not just nearest neighbors) but cases farther away get less weight. The weights are decided by a **kernel** function (typically a function peaked at zero, and tapering off symmetrically on both sides).



### How the kernel determines the weights

Now we may simply take the weighted average. We shall apply it to our parent-son example first.

```
parent = c(5.5, 5.4, 5.7, 5.1, 6.0, 5.5, 6.1)
son = c(5.9, 5.3, 5.9, 5.3, 5.8, 5.2, 6.0)
newParent = 5.3
kernel = dnorm #this is the N(0,1) density
```

Now let us find the weights (for a bin width ( $h$ ) = 0.5, say):

```
h = 0.5
wt = kernel((parent- newParent)/h)/h
```

Finally the weighted average:

```
newSon = weighted.mean(son,w = wt)
```

**Exercise:** Write a function called, say, `nw`, like this

```
nw = function(x,y,newX,kernel,h) {
  #write commands here
}
```

[Hint: Basically collect the lines that we used just now.]

Now use it to predict the value of `pmRA` when `RA` is 90 (use  $h=0.2$ ):

```
newRA = 90
newpmRA = nw(RA,pmRA,newRA,dnorm,0.2)
```



# Chapter 7

## MAXIMUM LIKELIHOOD ESTIMATION

*Notes by Donald Richards & Bhamidi V Rao*

### 1. The maximum likelihood method.

We seek a method which produces good estimators. The following method appeared in R. A. Fisher (1912), “On an absolute criterion for fitting frequency curves,” *Messenger of Math.* **41**, 155–160. This is Fisher’s first mathematical paper, written while a final-year undergraduate in mathematics and mathematical physics at Gonville and Caius College, Cambridge University. It’s not clear what motivated Fisher to study this subject; perhaps it was the influence of his tutor, the *astronomer* F. J. M. Stratton. Fisher’s paper started with a criticism of two methods of curve fitting, the least-squares and the method of moments.

$X$  is a random variable and  $f(x; \theta)$  is a statistical model for  $X$ . Here  $\theta$  is a parameter.  $X_1, \dots, X_n$  is a random sample from  $X$ . We want to construct good estimators for  $\theta$  using the random sample.

Here is an example from Protheroe, et al. “Interpretation of cosmic ray composition - The path length distribution,” (1981 *Ap J* , 247).  $X$  is length of paths.  $X$  is modeled as an exponential variable with density,

$$f(x; \theta) = \theta^{-1} \exp(-x/\theta), \quad x > 0.$$

Under this model,

$$E(X) = \int_0^{\infty} x f(x; \theta) dx = \theta.$$

Intuition suggests using  $\bar{X}$  to estimate  $\theta$ . Indeed,  $\bar{X}$  is unbiased and consistent.

Here is another example. LF for globular clusters in the Milky Way; Van den Bergh’s normal model,

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x - \mu)^2}{2\sigma^2}\right]$$

$\mu$  is the mean visual absolute magnitude and  $\sigma$  is the standard deviation of visual absolute magnitude for Galactic globulars. Again,  $\bar{X}$  is a good estimator for  $\mu$  and  $S^2$  is a good estimator for  $\sigma^2$ .

Choose a globular cluster at random; what is the chance that the LF will be *exactly* -7.1 mag? *Exactly* -7.2 mag? For any continuous random variable  $X$ , it is easy to see that,  $P(X = c) = 0$ , whatever be the number  $c$ . Specifically suppose that  $X \sim N(\mu = -6.9, \sigma^2 = 1.21)$ , then  $P(X = -7.1) = 0$ , but

$$f(-7.1) = \frac{1}{1.1\sqrt{2\pi}} \exp\left[-\frac{(-7.1 + 6.9)^2}{2(1.1)^2}\right] = 0.37$$

Fisher's interpretation is that in one simulation of the random variable  $X$ , the "likelihood" of observing the number  $-7.1$  is  $0.37$ . Similarly,  $f(-7.2) = 0.28$ . In one simulation of  $X$ , the value  $x = -7.1$  is 32% more likely to be observed than the value  $x = -7.2$ . In this model,  $x = -6.9$  is the value which has the greatest, or maximum likelihood, for it is where the probability density function is at its maximum. Do not confuse 'likelihood of observing' with 'probability of observing'. As mentioned earlier, the probability of observing any given value is zero.

Return to a general model  $f(x; \theta)$ ; and a random sample  $X_1, \dots, X_n$  from this population. The *joint* probability density function of the sample is

$$f(x_1; \theta)f(x_2; \theta) \cdots f(x_n; \theta)$$

Here the variables are the  $x$ 's, while  $\theta$  is fixed. This is probability density function in the variables  $x_1, \dots, x_n$ .

Fisher's brilliant idea: choose that value of  $\theta$  which makes the observations most likely, that is, choose that value of  $\theta$  which makes the likelihood maximum. Reverse the roles of the  $x$ 's and  $\theta$ . Regard the  $x$ 's as fixed and  $\theta$  as the variable. When we do this and think of it as a function of  $\theta$  it is called the **likelihood function**. It tells us the likelihood of coming up with the sample  $x_1, \dots, x_n$  when the model is  $f(x, \theta)$ . This is denoted simply as  $L(\theta)$ .

We define  $\hat{\theta}$ , the **maximum likelihood estimator (MLE)** of  $\theta$ , as that value of  $\theta$  where  $L$  is maximized. Thus  $\hat{\theta}$  is a function of the  $X$ 's. Several questions arise now. Does the maximum exist, is it unique, is it a good estimator etc. In the situations we consider, all these have affirmative answers, though in general the situation is not so nice.

Return to the example “cosmic ray composition - The path length distribution ...”. The length of paths,  $X$ , is exponential

$$f(x; \theta) = \theta^{-1} \exp(-x/\theta), \quad x > 0$$

Likelihood function is

$$\begin{aligned} L(\theta) &= f(X_1; \theta) f(X_2; \theta) \cdots f(X_n; \theta) \\ &= \theta^{-n} \exp(-(X_1 + \cdots + X_n)/\theta) = \theta^{-n} \exp(-n\bar{X}/\theta) \end{aligned}$$

Maximizing  $L$  is equivalent to maximizing  $\ln L$ :

$$\begin{aligned} \ln L(\theta) &= -n \ln(\theta) - n\bar{X}\theta^{-1} \\ \frac{d}{d\theta} \ln L(\theta) &= -n\theta^{-1} + n\bar{X}\theta^{-2} \\ \frac{d^2}{d\theta^2} \ln L(\theta) &= n\theta^{-2} - 2n\bar{X}\theta^{-3} \end{aligned}$$

Solving the equation  $d \ln L(\theta)/d\theta = 0$ , we get  $\theta = \bar{X}$ . Of course we need to check that  $d^2 \ln L(\theta)/d\theta^2 < 0$  at  $\theta = \bar{X}$ . Thus  $\ln L(\theta)$  is maximized at  $\theta = \bar{X}$ . Conclusion: The MLE of  $\theta$  is  $\hat{\theta} = \bar{X}$ .

Return to the LF for globular clusters;  $X \sim N(\mu, \sigma^2)$ . Assume that  $\sigma$  is known (1.1 mag, say). Likelihood function is

$$\begin{aligned} L(\mu) &= f(X_1; \mu) f(X_2; \mu) \cdots f(X_n; \mu) \\ &= (2\pi)^{-n/2} \sigma^{-n} \exp \left[ -\frac{1}{2\sigma^2} \sum_{i=1}^n (X_i - \mu)^2 \right]. \end{aligned}$$

Maximizing  $\ln L$  using calculus, we get  $\hat{\mu} = \bar{X}$ .

Continuing with LF for globular clusters, suppose now that both  $\mu$  and  $\sigma^2$  are unknown. We have now likelihood function of two variables,

$$L(\mu, \sigma^2) = f(X_1; \mu, \sigma^2) \cdots f(X_n; \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp \left[ -\frac{1}{2\sigma^2} \sum_{i=1}^n (X_i - \mu)^2 \right]$$

$$\begin{aligned}\ln L &= -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (X_i - \mu)^2 \\ \frac{\partial}{\partial \mu} \ln L &= \frac{1}{\sigma^2} \sum_{i=1}^n (X_i - \mu) \\ \frac{\partial}{\partial (\sigma^2)} \ln L &= -\frac{n}{2\sigma^2} + \frac{1}{2(\sigma^2)^2} \sum_{i=1}^n (X_i - \mu)^2\end{aligned}$$

Solve for  $\mu$  and  $\sigma^2$ , the simultaneous equations:

$$\frac{\partial}{\partial \mu} \ln L = 0, \quad \frac{\partial}{\partial (\sigma^2)} \ln L = 0$$

We also verify that  $L$  is concave at the solutions of these equations (Hessian matrix). Conclusion: The MLEs are

$$\hat{\mu} = \bar{X}, \quad \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$$

$\hat{\mu}$  is unbiased:  $E(\hat{\mu}) = \mu$ .  $\hat{\sigma}^2$  is not unbiased:  $E(\hat{\sigma}^2) = \frac{n-1}{n} \sigma^2 \neq \sigma^2$ . For this reason, we use  $\frac{n}{n-1} \hat{\sigma}^2 \equiv S^2$ .

Calculus cannot always be used to find MLEs. Return to the “cosmic ray composition” example.

$$f(x; \theta) = \begin{cases} \exp(-(x - \theta)) & \text{if } x \geq \theta \\ 0, & \text{if } x < \theta \end{cases}$$

The likelihood function is

$$\begin{aligned}L(\theta) &= f(X_1; \theta) \cdots f(X_n; \theta) \\ &= \begin{cases} \exp[-\sum_{i=1}^n (X_i - \theta)], & \text{if all } X_i \geq \theta, \\ 0, & \text{if any } X_i < \theta \end{cases}\end{aligned}$$

Let  $X_{(1)}$  be the smallest observation in the sample. “All  $X_i \geq \theta$ ” is equivalent to “ $X_{(1)} \geq \theta$ ”. Thus

$$L(\theta) = \begin{cases} \exp(-n(\bar{X} - \theta)), & \text{if } \theta \leq X_{(1)} \\ 0, & \text{if } X_{(1)} < \theta \end{cases}$$

Conclusion:  $\hat{\theta} = X_{(1)}$ .

General Properties of the MLE  $\hat{\theta}$ :

(a)  $\hat{\theta}$  may be biased. We often can remove this bias by multiplying  $\hat{\theta}$  by a constant.

(b) For many models,  $\hat{\theta}$  is consistent.

(c) The Invariance Property: For many nice functions  $g$ , if  $\hat{\theta}$  is the MLE of  $\theta$ , then  $g(\hat{\theta})$  is the MLE of  $g(\theta)$ .

(d) The Asymptotic Property: For large  $n$ ,  $\hat{\theta}$  has an approximate normal distribution with mean  $\theta$  and variance  $1/B$  where

$$B = nE \left[ \frac{\partial}{\partial \theta} \ln f(X; \theta) \right]^2$$

This asymptotic property can be used to develop confidence intervals for  $\theta$ .

The method of maximum likelihood works well when intuition fails and no obvious estimator can be found. When an obvious estimator exists the method of ML often will find it. The method can be applied to many statistical problems: regression analysis, analysis of variance, discriminant analysis, hypothesis testing, principal components, etc.

## 2. The ML Method for Testing Hypotheses.

Suppose that  $X \sim N(\mu, \sigma^2)$ , so that

$$f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(x - \mu)^2}{2\sigma^2} \right].$$

Using a random sample  $X_1, \dots, X_n$ , we wish to test  $H_0 : \mu = 3$  vs.  $H_a : \mu \neq 3$ .

Here the parameter space is the space of all permissible values of the parameters

$$\Omega = \{(\mu, \sigma) : -\infty < \mu < \infty, \sigma > 0\}.$$

The hypotheses  $H_0$  and  $H_a$  represent restrictions on the parameters, so we are led to parameter subspaces

$$\omega_0 = \{(\mu, \sigma) : \mu = 3, \sigma > 0\}; \quad \omega_a = \{(\mu, \sigma) : \mu \neq 3, \sigma > 0\}.$$

$$L(\mu, \sigma^2) = f(X_1; \mu, \sigma^2) \cdots f(X_n; \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp \left[ -\frac{1}{2\sigma^2} \sum_{i=1}^n (X_i - \mu)^2 \right]$$

Maximize  $L(\mu, \sigma^2)$  over  $\omega_0$  and  $\omega_a$ . The **likelihood ratio test statistic (LRT)** is

$$\lambda = \frac{\max_{\omega_0} L(\mu, \sigma^2)}{\max_{\omega_a} L(\mu, \sigma^2)} = \frac{\max_{\sigma > 0} L(3, \sigma^2)}{\max_{\mu \neq 3, \sigma > 0} L(\mu, \sigma^2)}.$$

Fact:  $0 \leq \lambda \leq 1$ . Clearly,  $L(3, \sigma^2)$  is maximized over  $\omega_0$  at

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (X_i - 3)^2$$

$$\max_{\omega_0} L(3, \sigma^2) = L\left(3, \frac{1}{n} \sum_{i=1}^n (X_i - 3)^2\right) = \left[ \frac{n}{2\pi e \sum_{i=1}^n (X_i - 3)^2} \right]^{n/2}.$$

$L(\mu, \sigma^2)$  is maximized over  $\omega_a$  at

$$\mu = \bar{X}, \quad \sigma^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$$

$$\max_{\omega_a} L(\mu, \sigma^2) = L\left(\bar{X}, \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2\right) = \left[ \frac{n}{2\pi e \sum_{i=1}^n (X_i - \bar{X})^2} \right]^{n/2}.$$

The likelihood ratio test statistic is

$$\begin{aligned} \lambda &= \left[ \frac{n}{2\pi e \sum_{i=1}^n (X_i - 3)^2} \right]^{n/2} \div \left[ \frac{n}{2\pi e \sum_{i=1}^n (X_i - \bar{X})^2} \right]^{n/2} \\ &= \left[ \sum_{i=1}^n (X_i - \bar{X})^2 \div \sum_{i=1}^n (X_i - 3)^2 \right]^{n/2} \end{aligned}$$

$\lambda$  is close to 1 iff  $\bar{X}$  is close to 3.  $\lambda$  is close to 0 iff  $\bar{X}$  is far from 3. This particular LRT statistic  $\lambda$  is equivalent to the  $t$ -statistic seen earlier. In this case, the ML method discovers the obvious test statistic.

### 3. Cramér-Rao inequality.

Given two unbiased estimators, we prefer the one with smaller variance. In our quest for unbiased estimators with minimum possible variance, we need to know how small their variances can be.

Suppose that  $X$  is a random variable modeled by the density  $f(x; \theta)$  where  $\theta$  is a parameter. The “support” of  $f$  is the region where  $f > 0$ . We assume that the “support” of  $f$  does not depend on  $\theta$ . We have a random sample,  $X_1, \dots, X_n$ . Suppose that  $Y$  is an unbiased estimator of  $\theta$ .

**The Cramér-Rao Inequality:** If  $Y$  is an unbiased estimator of  $\theta$  based on a sample of size  $n$ , then the smallest possible value that  $\text{Var}(Y)$  can attain is  $1/B$  where

$$B = nE \left[ \frac{\partial}{\partial \theta} \ln f(X; \theta) \right]^2 = -nE \left[ \frac{\partial^2}{\partial \theta^2} \ln f(X; \theta) \right].$$

This is the same  $B$  which appeared in our study of MLEs. Here the expected value is taken under the model  $f(x, \theta)$ . Certain regularity conditions need to hold for this to be true, but we shall not go into the mathematical details.

To illustrate, let us consider the example: “... cosmic ray composition - The path length distribution ...”. Here  $X$  is the length of paths modeled by

$$f(x; \theta) = \theta^{-1} \exp(-x/\theta), \quad x > 0.$$

$$\ln f(X; \theta) = -\ln \theta - \theta^{-1}X; \quad \frac{\partial}{\partial \theta} \ln f(X; \theta) = -\theta^{-1} + \theta^{-2}X.$$

$$\frac{\partial^2}{\partial \theta^2} \ln f(X; \theta) = \theta^{-2} - 2\theta^{-3}X; \quad E \left[ \frac{\partial^2}{\partial \theta^2} \ln f(X; \theta) \right] = -\theta^{-2}.$$

The smallest possible value of  $\text{Var}(Y)$  is  $\theta^2/n$ . This is attained by  $\bar{X}$ . For this problem,  $\bar{X}$  is *the* best unbiased estimator of  $\theta$ .

Suppose that  $Y$  is an unbiased estimator of the parameter  $\theta$ . We compare  $\text{Var}(Y)$  with  $1/B$ , the lower bound in the Cramér-Rao inequality:

$$\frac{1}{B} \div \text{Var}(Y)$$

This number is called the **efficiency** of  $Y$ . Obviously,  $0 \leq \text{efficiency} \leq 1$ . If  $Y$  has 50% efficiency then about  $1/0.5 = 2$  times as many sample observations are needed for  $Y$  to perform as well as the MVUE. The use of  $Y$  results in confidence intervals which generally are longer than those arising from the MVUE. If the MLE is unbiased then as  $n$  becomes large, its efficiency increases to 1.

The Cramér-Rao inequality can be stated as follows. If  $Y$  is any unbiased estimator of  $\theta$ , based on sample of size  $n$ , then

$$\text{Var}(Y) \geq \frac{1}{nI}; \quad \text{where} \quad I = E \left[ \frac{\partial}{\partial \theta} \ln f(X; \theta) \right]^2.$$

The quantity  $I$  is called Fisher information number. Under the same regularity conditions needed for the Cramér-Rao inequality to hold, one has a nice desirable property of the MLE. Let  $\hat{\theta}_n$  be the MLE for  $\theta$  based on a sample of size  $n$ . Then, for large  $n$ ,

$$\sqrt{n} (\hat{\theta}_n - \theta) \approx N(0, 1/I).$$

In other words,  $\hat{\theta}$  is consistent as well as asymptotically efficient. Not only that, the above approximate distribution can be used to get confidence intervals also.

Dembo, Cover, and Thomas (1991) “Information-theoretic inequalities,” IEEE Trans. Information Theory 37, 1501–1518, provide a unified treatment of the Cramér-Rao inequality, the Heisenberg uncertainty principle, entropy inequalities, Fisher information, and many other inequalities in statistics, mathematics, information theory, and physics. In particular, they show that the Heisenberg uncertainty principle is a consequence of the Cramér-Rao inequality and in a sense, they are equivalent. This remarkable paper demonstrates that there is a basic oneness among these various fields.

Several interesting questions arise. For example, if you have found an estimator that did not attain the minimum variance as in the Cramér-Rao bound, what should we do. Should we look for an estimator whose variance attains the bound? Is there one such? If not what is to be done? We are not going into the details now.

## The Snake Example

### Uniform Distribution

A random variable  $X$  is said to be uniformly distributed over the interval  $[a, b]$  if for any  $c, d$  and  $\eta$  with  $a \leq c \leq d \leq d + \eta \leq b$ , we have

$$P(X \in [c, c + \eta]) = P(X \in [d, d + \eta]).$$

**Notation:**  $X \sim [a, b]$ .

It may be shown from the above condition that the probability density function  $f$  of this random variable is given by

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq x \leq b \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

and its probability distribution function is

$$F(x) = \begin{cases} 0 & \text{if } x < a \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ 1 & \text{if } x > b. \end{cases} \quad (2)$$

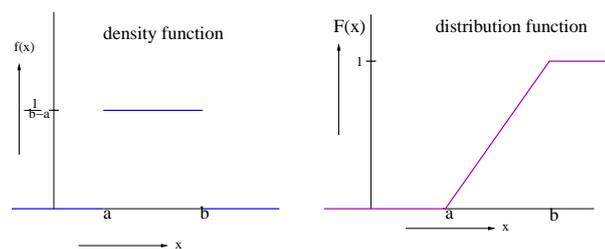


Figure 1: The density function and distribution function of a  $U[a, b]$  random variable.

### Estimation

Suppose we are interested in determining lengths of baby snakes.

**Assumption:** The length of a baby snake is uniformly distributed over the interval  $[0, \theta]$  and  $\theta$  is unknown. We want to estimate  $\theta$ .

**Procedure to be followed:** We take a sample of  $n$  baby snakes and measure their lengths. Let  $X_1, X_2, \dots, X_n$  be the lengths of the snakes obtained in the sample. *Note, these are i.i.d. random variables, because we have not yet drawn the sample, but just setting up the procedure to be followed after the sample is drawn.*

By our assumption, each of  $X_1, X_2, \dots, X_n$  follows a  $U[0, \theta]$  distribution with  $\theta$  unknown.

Let  $X_{(n)} = \max\{X_1, X_2, \dots, X_n\}$  denote the length of the longest snake in the sample. *Again remember that this quantity  $X_{(n)}$  is also a random variable.*

*Results*

$\frac{n+1}{n}X_{(n)}$  is an unbiased estimator of  $\theta$ .

$X_{(n)}$  is the MLE of  $\theta$ .

*Reason*

First let us compute the distribution of the random variable  $X_{(n)}$ .

Note

- (a) no snake will be less than 0 in length, so  $P(X_{(n)} < 0) = 0$ ,
- (b) no snake will be longer than  $\theta$  in length, so  $P(X_{(n)} \leq \theta) = 1$
- (c) For  $x$  such that  $0 \leq x \leq \theta$ , we have

$$\begin{aligned}
 P(X_{(n)} \leq x) &= P(\max\{X_1, X_2, \dots, X_n\} \leq x) \\
 &= P(X_1 \leq x, X_2 \leq x, \dots, X_n \leq x) \\
 &= P(X_1 \leq x)P(X_2 \leq x) \dots P(X_n \leq x) \text{ by independence} \\
 &= P(X_1 \leq x)^n \text{ because } X_1, X_2, \dots, X_n \text{ are i.i.d.} \\
 &= [F(x)]^n \text{ where } F, \text{ the distribution fn. of } X_1 \text{ is as given in (2)}
 \end{aligned}$$

So, letting  $G(x)$  and  $g(x)$  denote, respectively, the distribution function and density function of  $X_{(n)}$  we have from the above,

$$G(x) = \begin{cases} 0 & \text{if } x < 0 \\ [F(x)]^n & \text{if } 0 \leq x \leq \theta \\ 1 & \text{if } x > \theta. \end{cases}$$

and

$$\begin{aligned}
 g(x) = \frac{d}{dx}G(x) &= \begin{cases} 0 & \text{if } x < 0 \\ n[F(x)]^{n-1} \frac{d}{dx}F(x) & \text{if } 0 \leq x \leq \theta \\ 0 & \text{if } x > \theta \end{cases} \\
 &= \begin{cases} 0 & \text{if } x < 0 \\ n[F(x)]^{n-1} f(x) & \text{if } 0 \leq x \leq \theta \\ 0 & \text{if } x > \theta \end{cases} \\
 &= \begin{cases} 0 & \text{if } x < 0 \\ n \left[\frac{x}{\theta}\right]^{n-1} \frac{1}{\theta} & \text{if } 0 \leq x \leq \theta \\ 0 & \text{if } x > \theta. \end{cases}
 \end{aligned}$$

And we see after elementary calculations

$$E(X_{(n)}) = \int_0^\theta xg(x)dx = \frac{n}{n+1}\theta$$

thereby yielding that  $\frac{n+1}{n}X_{(n)}$  is an unbiased estimator of  $\theta$ .

To obtain the MLE, note that the joint density function of  $X_1, X_2, \dots, X_n$  is

$$f_n(x_1, x_2, \dots, x_n) = \begin{cases} 0 & \text{if, for any } i, x_i < 0 \\ \left[\frac{1}{\theta}\right]^n & \text{if } 0 \leq x_1, \dots, x_n \leq \theta \\ 0 & \text{if, for any } i, x_i > \theta, \end{cases}$$

so the likelihood function is

$$L(\theta|x_1, \dots, x_n) = \begin{cases} \left[\frac{1}{\theta}\right]^n & \text{if } 0 \leq \max\{x_1, \dots, x_n\} \leq \theta \\ 0 & \text{otherwise,} \end{cases}$$

And this function is maximized when  $\theta = \max\{x_1, \dots, x_n\}$ .

Thus the MLE of  $\theta$  is  $X_{(n)}$ .

*After collecting the sample*

Now if we obtain the following sample of size 10

length in metres	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$
	.2	.48	.15	.3	.12	.09	.39	.26	.44	.17

then we say that

an unbiased estimate for  $\theta$  is  $\frac{11}{10}(0.48)$

and

the MLE for  $\theta$  is 0.48.



## Chapter 8

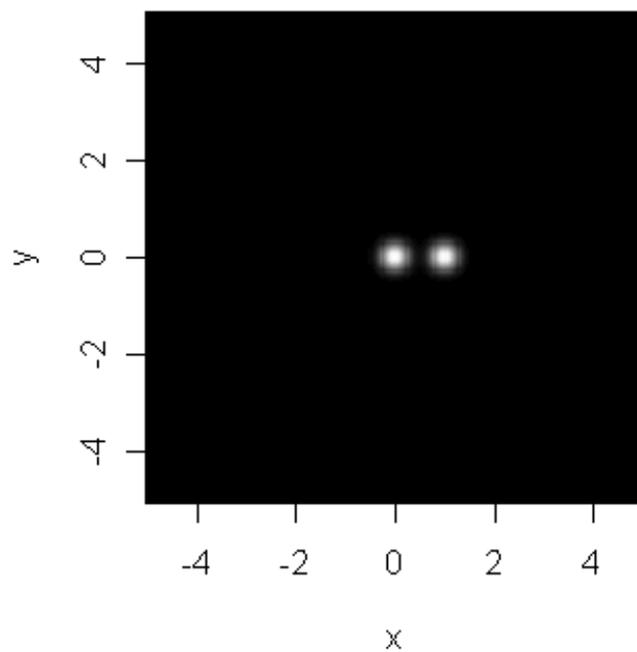
# TESTING AND ESTIMATION IN R

*Notes by Arnab Chakraborty*

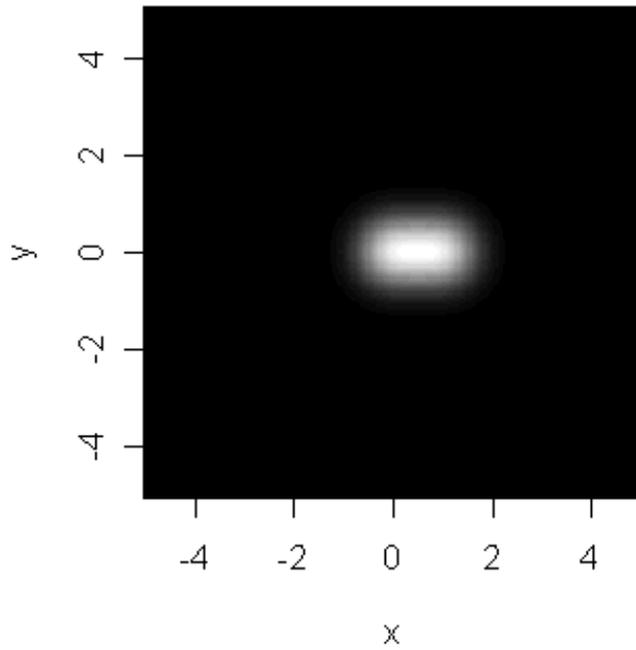
# Testing and Estimation

## A simple example

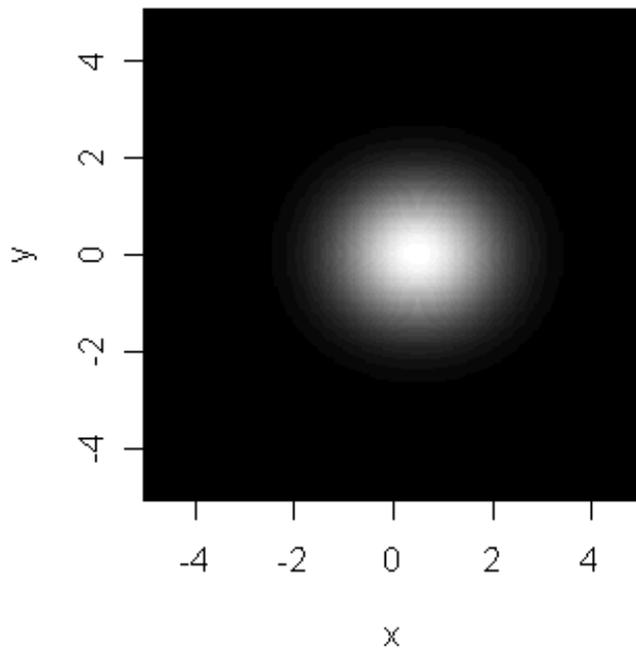
Consider the computer-generated picture below which is supposed to mimic a photograph taken by a low resolution telescope. Is it reasonable to say that there are two distinct stars in the picture?



The next image is obtained by reducing the resolution of the telescope. Each star is now more blurred. Indeed, their distance in the picture is almost overwhelmed in the blur, and they appear to be an elongated blur caused by a single star.



The last picture is of the same two stars taken through a very low resolution telescope. *Based on this image alone* there is just no reason to believe that we are seeing *two* stars.



What we did just now is an informal version of a statistical test of hypothesis. Each time we tried to answer the question "Are there two stars or one?" based on a blurry picture. The two possibilities

**"There is just one star"** as opposed to **"There are two stars"**

are called two **hypotheses**. The simpler of the two gets the name **null hypothesis** while the other is called the **alternative hypothesis**. Here we shall take the first one as the null, and the second one as the alternative hypothesis.

The blurry picture we use to make a decision is called the statistical data (which always contains random errors in the form of blur/noise). Notice how our final verdict changes with the amount of noise present in the data.

Let us follow the thought process that led us to a conclusion based on the first picture. We first mentally made a note of the amount of blur. Next we imagined the centers of the bright blobs. If there are two stars then they are most likely to be here. Now we compare the distance between these centers and the amount of blur present. If the distance seems too small compared to the blur then we pass off the entire bundle as a single star.

This is precisely the idea behind most statistical tests. We shall see this for the case of the two sample *t*-test.

## Do Hyades stars differ in color from the rest?

Recall in the Hipparcos data set we had 92 Hyades stars and 2586 non-Hyades stars. We want to test the null hypothesis

"The Hyades stars have the same color as the non-Hyades stars"

versus the alternative hypothesis

"They have different colors."

First let us get hold of the data for both the groups. For this we shall use the little script we had saved earlier. This script is going to load the entire Hipparcos data set, and extract the Hyades stars. **So you must make sure that both [HIP.dat](#) and [hyad.r](#) are saved on your desktop.** Right-click on the links in the last line, choose "Save link as..." or "Save target as..." (and be careful that the files do not get saved as .txt files). Then you must navigate R to the desktop. The simplest way to achieve this is to use the File > Change dir... menu item. This will open a pop up like this:



### Changing directory in R (Windows version)

Click on Desktop, and then OK.

```
source("hyad.r")
color = B.V
H = color[HyadFilter]
nH = color[!HyadFilter & !is.na(color)]
m = length(H)
n = length(nH)
```

In the definition of `nH` above, we needed to exclude the **NA** values. `H` is a list of  $m$  numbers and `nH` is a list of  $n$  numbers.

First we shall make an estimate of the "blur" present in the data. For this we shall compute the pooled estimate of standard deviation.

```
blur.H = var(H)
blur.nH = var(nH)
blur.pool = ((m-1)*var(H) + (n-1)*var(nH)) / (m+n-2)
```

Next we shall find the difference of the two means:

```
meanDiff = mean(H) - mean(nH)
```

Finally we have to compare the difference with the blur. One way is to form their ratio.

```
(meanDiff/sqrt(blur.pool)) / sqrt(1/m + 1/n)
```

This last factor (which is a constant) is there only for technical reasons (you may think of it as a special constant to make the "units match").

The important question now is "Is this ratio small or large?" For the image example we provided a subjective answer. But in statistics we have an objective way to proceed. Before we see that let us quickly learn a one-line shortcut to compute the above ratio using the **t.test** function.

```
t.test(H,nH,var.eq=T) #we shall explain the "var.eq" soon
```

Do you see the ratio in the output? Also this output tells us whether the ratio is to be considered small or large. It does so in a somewhat diplomatic way using a number called the **p-value**. Here the *p*-value is near 0, meaning

if the colors were really the same then the chance of observing a ratio this large (or larger) is almost 0.

Typically, if the *p*-value is smaller than 0.05 then we reject the null hypothesis. So we conclude that the mean of the color of the Hyades stars is indeed different from that of the rest.

 A rule of thumb: For any statistical test (not just *t*-test) accept the null hypothesis if and only if the *p*-value is above 0.05. Such a test fails to recognize a true null hypothesis at most 5% of the time.

The `var.eq=T` option means we are assuming that the colors of the Hyades and non-Hyades stars have more or less the same variance. If we do not want to make this assumption, we should simply write

```
t.test(H,nH)
```

 Remember: **t.test** is for comparing means.

## Chi-squared tests for categorical data

Suppose that you are to summarize the result of a public examination. It is not reasonable to report the grades obtained by each and every student in a summary report. Instead, we break the range of grades into *categories* like A,B,C etc and then report the numbers of students in each category. This gives an overall idea about the distribution of grades.

The **cut** function in R does precisely this.

```
bvcat = cut(color, breaks=c(-Inf,0.5,0.75,1,Inf))
```

Here we have broken the range of values of the `B.V` variable into 4 categories:

`(-Inf, 0.5]`, `(0.5, 0.75]`, `(0.75, 1]` and `(1, Inf)`.

The result (stored in `bvcat`) is a vector that records the category in which each star falls.

```
bvcat
table(bvcat)
plot(bvcat)
```

It is possible to tabulate this information for Hyades and non-Hyades stars in the same table.

```
table(bvcat, HyadFilter)
```

To perform a chi-squared test of the null hypothesis that the true population proportions falling in the four categories are the same for both the Hyades and non-Hyades stars, use the

**chisq.test** function:

```
chisq.test(bvcat, HyadFilter)
```

Since we already know these two groups differ with respect to the `B.V` variable, the result of this test is not too surprising. But it does give a qualitatively different way to compare these two distributions than simply comparing their means.

The test above is usually called a **chi-squared test of homogeneity**. If we observe only one sample, but we wish to test whether the categories occur in some pre-specified proportions, a similar test (and the same R function) may be applied. In this case, the test is usually called the **chi-squared test of goodness-of-fit**. We shall see an example of this next.

Consider once again the Hipparcos data. We want to know if the stars in the Hipparcos survey come equally from all corners of the sky. In fact, we shall focus our attention only on the `RA` values. First we shall break the range of `RA` into 20 equal intervals (each of width 18 degrees), and find how many stars fall in each bin.

```
count = table(cut(RA, breaks=seq(0, 360, len=20)))
chisq.test(count)
```

## Kolmogorov-Smirnov Test

There is yet another way (a better way in many situations) to perform the same test. This is called the **Kolmogorov-Smirnov test**.

```
ks.test(RA, "punif", 0, 360)
```

Here `punif` is the name of the distribution with which we are comparing the data. `punif` denoted the uniform distribution, the range being from 0 to 360. Thus, here we are testing if `RA` is taking all values from 0 to 360 with equal likelihood, or are some values being taken more or less frequently.

The Kolmogorov-Smirnov test has the advantage that we do not need to group the data into categories as for the chi-squared test.

 Remember: `chisq.test` and `ks.test` are for comparing distributions.

## Estimation

Finding (or, rather, guessing about) the unknown based on approximate information (data) is the aim of statistics. Testing hypotheses is one aspect of it where we seek to answer yes-no questions about the unknown. The problem of estimation is about guessing the values of unknown quantities.

There are many methods of estimation, but most start off with a **statistical model** of the data. This is a statement of how the observed data set is (probabilistically) linked with the unknown quantity of interest.

For example, if I am asked to estimate  $p$  based on the data

Head, Head, Head, Tail, Tail, Head, Head, Tail, Head,  
Tail, Tail, Head

then I cannot make head-or-tail of the question. I need to link this data set with  $p$  through a statement like

A coin with probability  $p$  was tossed 12 times and the data set was the result.

This is a **statistical model** for the data. Now the problem of estimating  $p$  from the data looks like a meaningful one.

 Statistical models provide the link between the observed data and the unknown reality. They are indispensable in any statistical analysis. Misspecification or over-simplification of the statistical model is the most frequent cause behind misuse of statistics.

## Estimation using R

You might be thinking that R has some in-built tool that can solve all estimation problems. Well, there isn't. In fact, due to the tremendous diversity among statistical models and estimation methods no statistical software can have tools to cope with *all* estimation problems. R tackles estimation in three major ways.

1. Many books/articles give formulas to estimate various quantities. You may use R as a calculator to implement them. With enough theoretical background you may be able to come up with your own formulas that R will happily compute for you.
2. Sometimes estimation problems lead to complicated equations. R can solve such equations for you numerically.
3. For some frequently used statistical methods (like regression or time series analysis) R has the estimation methods built into it.

To see estimation in action let us load the Hipparcos data set.

```
hip = read.table("HIP.dat", head=T)
attach(hip)
Vmag
```

If we assume the statistical model that the `Vmag` variable has a normal distribution with unknown mean  $\mu$  and unknown variance  $\sigma^2$  then it is known from the literature that a good estimator of  $\mu$  is  $\bar{X}$  and a 95% **confidence interval** is

$$\left( \bar{x} - 1.96 \frac{s}{\sqrt{n}}, \bar{x} + 1.96 \frac{s}{\sqrt{n}} \right),$$

where

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2.$$

This means that the true value of  $\mu$  will lie in this interval with 95% chance. You may think of  $\mu$  as a peg on the wall and the confidence interval as a hoop thrown at it. Then the hoop will miss the peg in only about 5% of the cases.

**Exercise:** Find the estimate and confidence interval for  $\mu$  based on the observed values of `Vmag` using R as a calculator.

Next we shall see a less trivial example. The data set comes from NASA's Swift satellite. The statistical problem at hand is modeling

the X-ray afterglow of gamma ray bursts. First, read in the dataset [GRB.dat](#) (right-click, save on your desktop, without any .txt extension).

```
dat = read.table("GRB.dat", head=T)
flux = dat[,2]
```

Suppose that it is known that the `flux` variable has an Exponential distribution. This means that its density function is of the form

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x > 0 \\ 0 & \text{else} \end{cases}$$

Here  $\lambda$  is a parameter, which must be positive. To get a feel of the density function let us plot it for different values of  $\lambda$

```
x = seq(0, 200, .1)
y = dexp(x, 1)
plot(x, y, ty="l")
```

Now let us look at the histogram of the observed `flux` values.

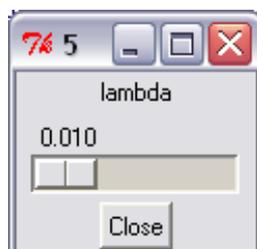
```
hist(flux)
```

The problem is estimating  $\lambda$  based on the data may be considered as finding a value of  $\lambda$  such that the the density is as close as possible to the histogram.

We shall first try to achieve this interactively. For this you need to download [interact.r](#) on your desktop first.

```
source("interact.r")
```

This should open a tiny window as shown below with a slider and a button in it.



**Screenshot of the tiny window**

Move the slider to see how the density curve moves over the histogram. Choose a position of the slider for which the density curve appears to be a good approximation.

**Exercise:** It is known from the theory of Exponential

distribution that the **Maximum Likelihood Estimate (MLE)** of  $\lambda$  is the reciprocal of the sample mean

$$\hat{\lambda} = \frac{1}{\bar{x}}.$$

Compute this estimate using R and store it in a variable called `lambdaHat`.

Draw the density curve on top of the histogram using the following commands.

```
y = dexp(x, lambdaHat)
lines(x, y, col="blue")
```

## Some nonparametric tests

The remaining part of this tutorial deals with a class of inference procedures called nonparametric inference. These may be skipped without loss of continuity. Also the theory part for this will be covered in a later theory class. So you may like to save the rest of this lab until that time. I have tried to make the lab largely self-explanatory though, with a little theoretical discussion to explain terms and concepts.

### Distributions

In statistics we often come across questions like

“Is a given data from such-n-such distribution?”

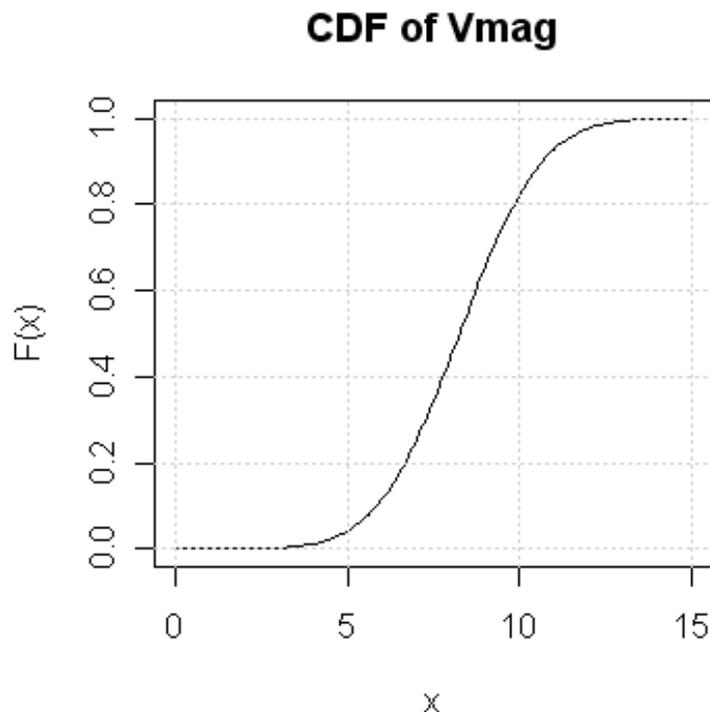
Before we can answer this question we need to know what is meant by a *distribution*. We shall talk about this first.

In statistics, we work with random variables. Chance controls their values.

The **distribution** of a random variable is the rule by which chance governs the values. If you know the distribution then you know the chance of this random variable taking value in any given range. This rule may be expressed in various ways. One popular way is via the **Cumulative Distribution Function (CDF)**, that we illustrate now. If a random variable ( $X$ ) has distribution with CDF  $F(x)$  then for any given number  $a$  the chance that  $X$  will be  $\leq a$  is  $F(a)$ .

**Example:** Suppose that I choose a random star from the Hipparcos data set. If I tell you that its `Vmag` is a random

variable with the following CDF then what is the chance that it takes values  $\leq 10$ ? Also find out the value  $x$  such that the chance of  $V_{\text{mag}}$  being  $\leq x$  is 0.4.



For the first question the required probability is  $F(10)$  which, according to the graph, is slightly above 0.8.

In the second part we need  $F(x) = 0.4$ . From the graph it seems to be about 7.5.

Just to make sure that these are indeed meaningful, let us load the Hipparcos data set and check:

```
hip = read.table("HIP.dat", head=T)
attach(hip)
n = length(Vmag) #total number of cases
count = sum(Vmag<=10) #how many <= 10
count/n #should be slightly above 0.8
```

This checks the answer of the first problem. For the second

```
count = sum(Vmag<=7.5) #how many <= 7.5
count/n #should be around 0.4
```

So you see how powerful a CDF is: in a sense it stores as much information as the entire  $V_{\text{mag}}$  data. It is a common practice in statistics to regard the distribution as the *ultimate truth* behind the data. We want to infer about the underlying distribution based on

the data. When we look at a data set we actually try to look at the underlying distribution *through* the data set!

R has many standard distributions already built into it. This basically means that R has functions to compute their CDFs. These functions all start with the letter **p**.

**Example:** A random variable has standard Gaussian distribution. We know that R computes its CDF using the function **pnorm**. How to find the probability that the random variable takes values  $\leq 1$ ?

The answer may be found as follows:

```
pnorm(1)
```

For every **p**-function there is a **q**-function that is basically its inverse.

**Example:** A random variable has standard Gaussian distribution. Find  $x$  such that the random variable is  $\leq x$  with chance 0.3.

Now we shall use the function **qnorm**:

```
qnorm(0.3)
```

OK, now that we are through our little theory session, we are ready for the nonparametric lab.

## One sample nonparametric tests

### Sign-test

In a sense this is the simplest possible of all tests. Here we shall consider the data set [LMC.dat](#) that stores the measured distances to the Large Magellanic Cloud. (As always, you'll need to save the file on your desktop.)

```
LMC = read.table("LMC.dat", head=T)
data = LMC[,2] #These are the measurements
data
```

We want to test if the measurements exceed 18.41 *on an average*. Now, this does *not* mean whether the average of the data exceeds 18.41, which is trivial to find out. The question here is actually about the underlying distribution. We want to know if the median of the underlying distribution exceeds 18.41, which is a less trivial

question, since we are not given that distribution.

We shall use the sign test to see if the median is 18.41 or larger. First it finds how many of the observations are above this value:

```
abv = sum(data>18.41)
abv
```

Clearly, if this number is large, then we should think that the median (of the underlying distribution) exceeds 18.41. The question is how large is "large enough"? For this we consult the binomial distribution to get the  $p$ -value. The rationale behind this should come from the theory class.

```
n = nrow(LMC)
pValue = 1-pbinom(abv-1, n, 0.5)
pValue
```

We shall learn more about  $p$ -values in a later tutorial. For now, we shall use the following rule of thumb:

If this  $p$ -value is below 0.05, say, we shall conclude that the median is indeed larger than 18.41.

### Wilcoxon's Signed Rank Test

As you should already know from the theoretical class, there is a test called Wilcoxon's Signed Rank test that is better (if a little more complicated) than the sign test. R provides the function `wilcox.test` for this purpose. We shall work with the Hipparcos data set this time, which we have already loaded. We want to see if the median of the distribution of the `pmRA` variable is 0 or not.

```
wilcox.test(pmRA, mu=0)
```

Since the  $p$ -value is pretty large (above 0.05, say) we shall conclude that the median is indeed equal to 0. R itself has come to the same conclusion.

Incidentally, there is a little caveat. For Wilcoxon's Rank Sum Test to be applicable we need the underlying distribution to be symmetric around the median. To get an idea about this we should draw the histogram.

```
hist(pmRA)
```

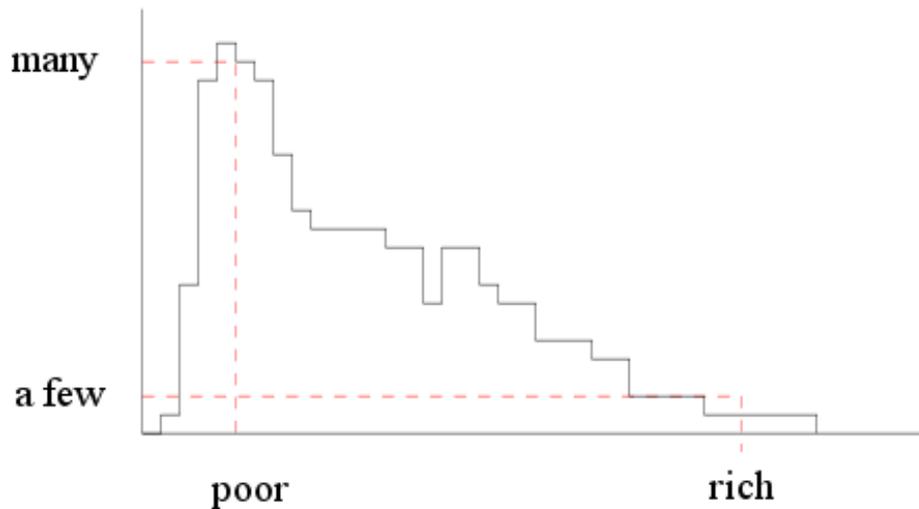
Well, it looks pretty symmetric (around a center line). But would you apply Wilcoxon's Rank Sum Test to the variable `DE`?

```
hist(DE)
```

No, this is not at all symmetric!

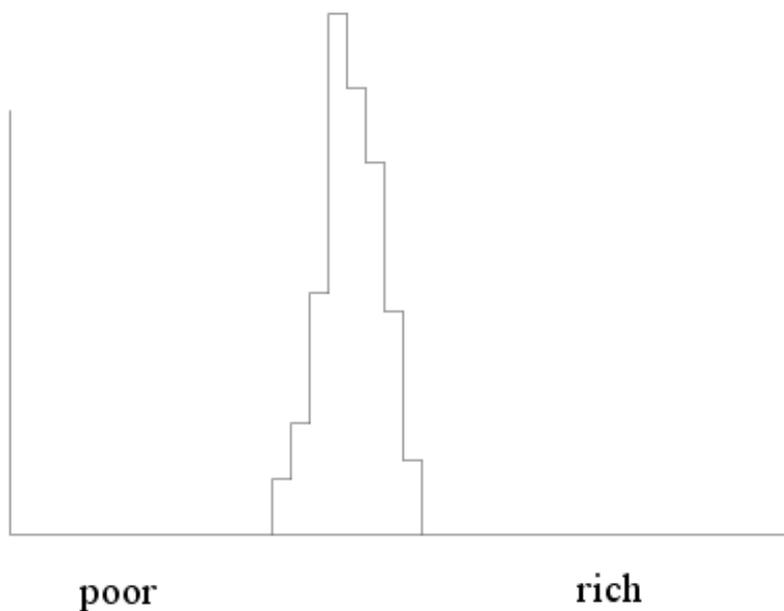
## Two-sample nonparametric tests

Here we shall be talking about the shape of distributions. Let us first make sure of this concept. Suppose that I make a list of all the people in a capitalist country and make a histogram of their incomes. I should get a histogram like this



**Income histogram of a capitalist country**

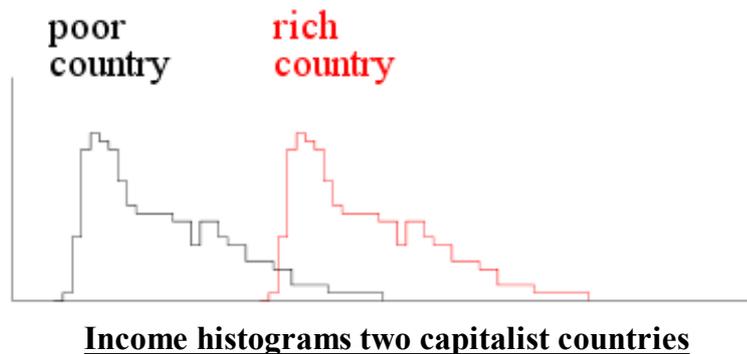
But if the same thing is done for a socialist country we shall see something like this (where almost everybody is in the middle income group).



**Income histogram of a socialist country**

Clearly, the *shapes* of the histograms differ for the two populations. We must be careful about the term "shape" here. For

example, the following two histograms are for two capitalist countries, one rich the other poor.



Here the histograms have the same *shape* but differ in *location*. You can get one from the other by applying a shift in the location.

This is a very common phenomenon in statistics. If we have two populations with comparable structure they often have similar shapes, and differ in just a location shift. In such a situation we are interested in knowing about the amount of shift. If the amount of shift is zero, then the two populations behave identically.

Wilcoxon's rank sum test is one method to learn about the amount of shift based on samples from the two populations.

We shall start with the example done in the theory class, where we had two samples

```
x = c(37, 49, 55, 57)
y = c(23, 31, 46)
m = length(x)
n = length(y)
```

We are told that both the samples come from populations with the same *shape*, though one may be a shifted version of the other.

Our aim is to check if indeed there is any shift or not.

We shall consider the pooled data (*i.e.*, all the 7 numbers taken together) and rank them

```
pool = c(x, y)
pool
r = rank(pool)
r
```

The first  $m=4$  of these ranks are for the first sample. Let us sum them:

```
H = sum(r[1:m])
H
```

If the the two distributions are really identical (*i.e.*, if there is no shift) then we should have (according to the theory class)  $H$  close to the value

```
|m*(m+n+1)/2    #Remember: * means multiplication
```

Can the  $H$  that we computed from our data be considered "close enough" to this value? We shall let us R determine that for us.

```
|wilcox.test(x,y)
```

So R clearly tells us that there is a shift in location. The output also mentions a  $W$  and a  $p$ -value. But  $W$  is essentially what we had called  $H$ . Well,  $H$  was 21, while  $W$  is 11. This is because R has the habit of subtracting

$$m(m+1)/2$$

from  $H$  and calling it  $W$ . Indeed for our data set

$$m(m+1)/2 = 10,$$

which perfectly accounts for the difference between our  $H$  and R's  $W$ .

You may recall from the theory class that  $H$  is called Wilcoxon's rank sum statistic, while  $W$  is the Mann-Whitney statistic (denoted by  $U$  in class).

Now that R has told us that there is a shift, we should demand to estimate the amount of shift.

```
|wilcox.test(x,y,conf.int=T)
```

The output gives two different forms of estimates. It gives a single value (a **point estimate**) which is 16. Before it gives a 95% **confidence interval**: -9 to 34. This basically means that

we can say with 95% confidence that the true value of the shift is between -9 and 34.

We shall talk more about confidence intervals later. For now let us see how R got the point estimate 16. It used the so-called Hodges-Lehman formula, that we have seen in the theoretical class. To see how this method works we shall first form the following "difference table"

		23	31	46
	-----			
37		14	6	-9
49		26	18	3

```
55| 32 24 9
57| 34 26 11
```

Here the rows are headed by the first sample, and columns by the second sample. Each entry is obtained by subtracting the column heading from the row heading (e.g.,  $-9 = 37 - 46$ ). This table, by the way, can be created very easily with the function **outer**.

```
outer(x, y, "-")
```



The **outer** function is a very useful (and somewhat tricky!) tool in R. It takes two vectors **x**, and **y**, say, and some function  $f(x,y)$ .

Then it computes a matrix where the  $(i,j)$ -th entry is

$$f(x[i], y[j]).$$

Now we take the median of all the entries in the table:

```
median(outer(x, y, "-"))
```

This gives us the Hodges-Lehman estimate of the location shift.

## Chapter 9

# TRUNCATION & CENSORING

*Notes by Jogesh Babu*

## Selection Biases: Truncation and Censoring

Jogesh Babu

Penn State University

### Outline

- 1 Censoring vs Truncation
- 2 Censoring
- 3 Statistical inferences for censoring
- 4 Truncation
- 5 Statistical inferences for truncation
- 6 Doubly truncated data
- 7 Recent methods

### Censoring vs Truncation

### Example: Left/Right Censoring

- **Censoring:** Sources/events can be detected, but the values (measurements) are not known completely. We only know that the value is less than some number.
- **Truncation:** An object can be detected only if its value is greater than some number; and the value is completely known in the case of detection. For example, objects of certain type in a specific region of the sky will not be detected by the instrument if the apparent luminosity of objects is less than a certain lower limit. This often happens due to instrumental limitations or due to our position in the universe.
- The main difference between censoring and truncation is that censored object is detectable while the object is not even detectable in the case of truncation.
- **Right Censoring:** the exact value  $X$  is not measurable, but only  $T = \min(X, C)$  and  $\delta = I(X \leq C)$  are observed.
- **Left Censoring:** Only  $T = \max(X, C)$  and  $\delta = I(X \geq C)$  are observed.

## Example: Interval/Double Censoring

## Survival Function

This occurs when we do not observe the exact time of failure, but rather two time points between which the event occurred:

$$(T, \delta) = \begin{cases} (X, 1) & : L < X < R \\ (R, 0) & : X > R \\ (L, -1) & : X < L \end{cases}$$

where  $L$  and  $R$  are left and right censoring variables.

- Cumulative failure function:

$$F(t) = P(T \leq t)$$

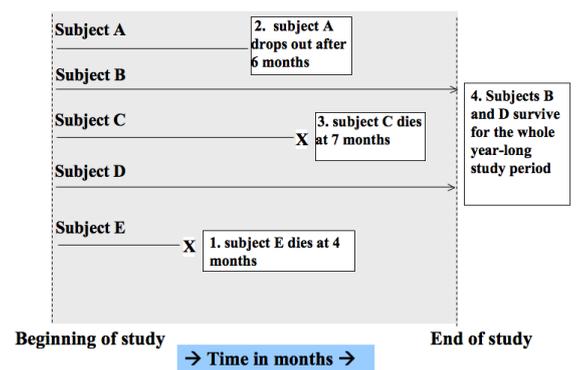
- Survivor function:

$$S(t) = P(T > t) = 1 - F(t)$$

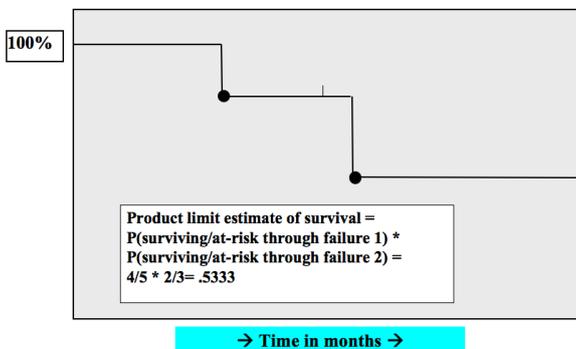
## Kaplan-Meier Estimator

- Nonparametric estimate of survivor function  
 $S(t) = P(T > t)$
- Intuitive graphical presentation
- Commonly used to compare two populations

## Survival Data



### Corresponding Kaplan-Meier Curve



### Kaplan-Meier Estimator (continued)

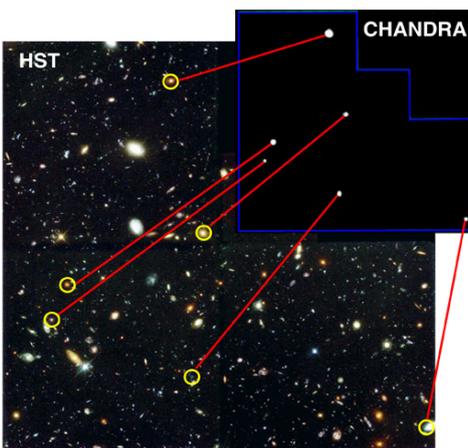
Let

- $t_i$ :  $i$ th ordered observation
- $d_i$ : number of 'censored' events at  $i$ th ordered observation
- $R_i$ : number of subjects 'at-risk' at  $i$ th ordered observation

The Kaplan Meier estimator of the survival function is

$$S(t) = \prod_{t_i \leq t} \left( 1 - \frac{d_i}{R_i} \right)$$

### Truncation



- **Left Truncation:** An event/source is detected if its measurement is greater than a truncation variable.
- **Right Truncation:** An event/source is detected if its measurement is less than a truncation variable.
- **Double Truncation:** This occurs when the time to event of interest in the study sample is in an interval.

The pair  $(X, Y)$  is observed only if  $X \geq Y$ ,  $X$  is the measurement of interest and  $Y$  is the truncation variable

$$M = m + 5 \log P - 5$$

$P$  parallax

Object is detected only if  $P \geq \ell$ .

Forty years ago, distinguished astrophysicist Donald Lynden-Bell derived a fundamental statistical result in the appendix to an astronomical study: the unique nonparametric maximum likelihood estimator for a univariate dataset subject to random truncation. The method is needed to establish luminosity functions from flux-limited surveys, and is far better than the more popular heuristic  $1/V_{max}$  method by Schmidt (1968). Two young astrostatisticians are now developing Lynden-Bell's method further. Schafer (2007) gives a nonparametric estimation for estimating the bivariate distribution when one variable is truncated. Kelly et al. use a Bayesian approach for a normal mixture model (combination of Gaussians) to the luminosity function. Lynden-Bell (1971, MNRAS.155, 95)

## Lynden-Bell Estimator

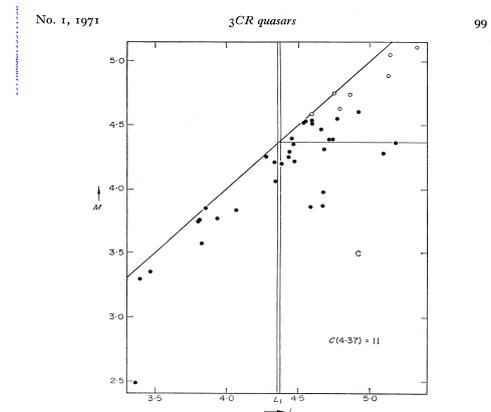


FIG. 1. Plot of the 40 3CR quasars in the  $L, M$  plane.  $L = P = \log(F_R/F_0)$   $M =$  limiting value of  $L$  beyond which an object of the same optical flux is rejected from 3CR as being too radio weak. The number of points in the box is denoted by  $C$ , the number (o) in the infinitesimal column is  $dX$ .

## Lynden-Bell-Woodroffe Estimator

## Lynden-Bell- Woodroffe Estimator (continued)

- Model: observe  $y$  only if  $y > u(x)$ .
- Data:  $(x_1, y_1), \dots, (x_n, y_n)$ .
- Risk set numbers:

$$N_j = \#\{i : u_i \leq y_{(j)} \text{ and } y_i \leq y_{(j)}\}$$

where  $u_i = u(x_i)$  and  $y_{(i)}$  is  $i$ th ordered value of  $y = (y_1, \dots, y_n)$

- In the KM estimator,  $N_j$  is the number of points at risk just before the  $j$ th event.
- The only differences in comparable points between truncated cases and censoring cases is that points with  $y_i > y_k$  but  $t(x_i) > y_k$  are not considered at risk in the truncated case. This is because these points cannot be observed.

Lynden-Bell-Woodroffe survival function estimator:

$$LBW(t) = \prod_{y_{(j)} \leq t} \left(1 - \frac{1}{N_j}\right)$$

Lynden-Bell, D. 1971, MNRAS, 155, 95  
Woodroffe, M. 1985, Ann. Stat., 13, 163

## Doubly truncated data: Efron's Nonparametric MLE

15,343 quasars

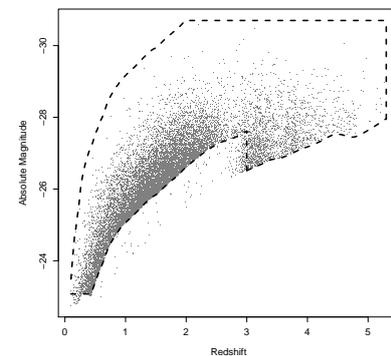
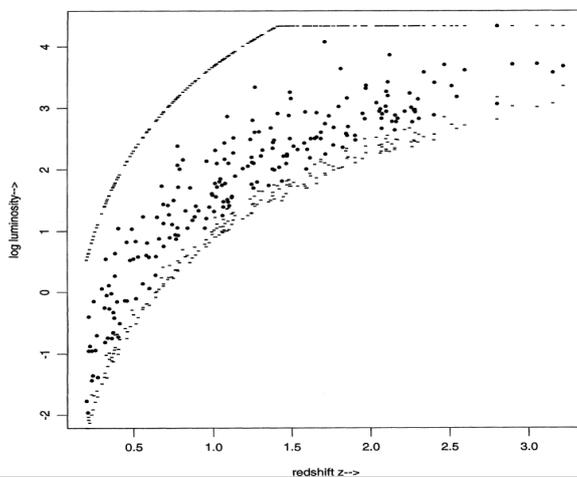


Fig. 1.— Quasar data from the Sloan Digital Sky Survey, the sample from Richards et al. (2006). Quasars within the dashed region are used in this analysis. The removed quasars are those with  $M \leq -23.075$ , which fall into the irregularly-shaped corner at the lower left of the plot, and those with  $z \leq 3$  and apparent magnitude greater than 19.1, which fall into a very sparsely sampled region.

## Data

The data shown in the figure, consists of 15,343 quasars. From these, any quasar is removed if it has  $z \geq 5.3$ ,  $z \leq 0.1$ ,  $M \geq -23.075$ , or  $M \leq -30.7$ . In addition, for quasars of redshift less than 3.0, only those with apparent magnitude between 15.0 and 19.1, inclusive, are kept; for quasars of redshift greater than or equal to 3.0, only those with apparent magnitude between 15.0 and 20.2 are retained. These boundaries combine to create the irregular shape shown by the dashed line. This truncation removes two groups of quasars from the Richards et al. (2006) sample. There are 15,057 quasars remaining after this truncation.

- Schafer, C. M. (2007, ApJ, 661, 703) uses semi-parametric methods  

$$\log \phi(z, M) = f(z) + g(M) + h(z, M, \theta), (z_i, M_i)$$
observed.
- Kelly et al. (2008, ApJ, 682, 874) use Bayesian approach for normal mixture model.
- The results obtained are better than the heuristic  $1/V_{\max}$  of Schmidt (1968, ApJ, 151, 393)

## Chapter 10

# NON-PARAMETRIC STATISTICS

*Notes by Sushama Bendre*

## 1. Parametric and Nonparametric models

A *parametric statistical model* is a model where the joint distribution of the observations involves several unknown constants called *parameters*. The functional form of the joint distribution is assumed to be known and the only unknowns in the model are the parameters. Two parametric models commonly encountered in astronomical experiments are

1. The Poisson model in which we assume that the observations are independent Poisson random variables with unknown common mean  $\theta$ .
2. The normal model in which the observations are independently distributed with unknown mean  $\mu$  and unknown variance  $\sigma^2$ .

In the first model  $\theta$  is the parameter and in the second  $\mu$  and  $\sigma^2$  are the parameters.

Anything we can compute from the observations is called a *statistic*. In parametric statistics the goal is to use observations to draw inference about the unobserved parameters and hence about the underlined model.

A *nonparametric model* is the one in which no assumption is made about the functional form of the joint distribution except that the observations are independent identically distributed (i.i.d.) from an arbitrary continuous distribution. As a result, the nonparametric statistics is also called *distribution free* statistics. There are no parameters in a nonparametric model.

A *semiparametric model* is the one which has parameters but very weak assumptions are made about the actual form of the distribution of the observations.

Both nonparametric and semiparametric models are often lumped together and called nonparametric models.

## 2. Why Nonparametric?

While in many situations parametric assumptions are reasonable (e.g. assumption of Normal distribution for the background noise, Poisson distribution for a photon counting signal of a nonvariable source), we often have no prior knowledge of the underlying distributions. In

such situations, the use of parametric statistics can give misleading or even wrong results.

We need statistical procedures which are insensitive to the model assumptions in the sense that the procedures retain their properties in the neighborhood of the model assumptions.

Insensitivity to model assumptions : **Robustness**

In particular, for

- Estimation

The estimators such that

- the variance (precision) of an estimator is not sensitive to model assumptions (Variance Robustness).

- Hypothesis Testing

We need test procedures where

- the level of significance is not sensitive to model assumptions (Level Robustness).
- the statistical power of a test to detect important alternative hypotheses is not sensitive to model assumptions (Power Robustness).

Apart from this, we also need procedures which are robust against the presence of outliers in the data.

**Examples:**

1. The sample mean is not robust against the presence of even one outlier in the data and is not variance robust as well. The sample median is robust against outliers and is variance robust.
2. The t-test does not have t-distribution if the underlying distribution is not normal and the sample size is small. For large sample size, it is asymptotically level robust but is not power robust. Also, it is not robust against the presence of outliers.

Procedures derived for nonparametric and semiparametric models are often called *robust* procedures since they depend on very weak assumptions.

### 3. Nonparametric Density Estimation

Let  $X_1, X_2, \dots, X_n$  be a random sample from an unknown probability density function  $f$ . The interest is to estimate the density function  $f$  itself.

Suppose the random sample is drawn from a distribution with known probability density function, say normal with mean  $\mu$  and variance  $\sigma^2$ . The density  $f$  can then be estimated by estimating the values of the unknown parameters  $\mu$  and  $\sigma^2$  from the data and substituting these estimates in the expression for normal density. Thus the *parametric density estimator* is

$$\hat{f}(x) = \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} \exp -\frac{1}{2\hat{\sigma}^2}(x - \hat{\mu})^2$$

where  $\hat{\mu} = \frac{\sum_i x_i}{n}$  and  $\hat{\sigma}^2 = \frac{\sum_i (x_i - \hat{\mu})^2}{n - 1}$ .

In case of the *nonparametric* estimation of the density function, the functional form of the density function is assumed to be unknown. We, however, assume that the underlined distribution has a probability density  $f$  and determine its form based on the data at hand. The oldest and widely used *nonparametric density estimator* is the histogram. Given an origin  $x_0$  and a *bandwidth*  $h$ , we consider the intervals of length  $h$ , also called *bins*, given by  $B_i = [x_0 + mh, x_0 + (m + 1)h)$  where  $m = 0, \pm 1, \pm 2, \dots$  and define the histogram by

$$\begin{aligned} \hat{f}_n(x) &= \frac{1}{nh} [\text{number of observations in the same bin as } x] \\ &= \frac{1}{nh} \sum_{i=1}^n n_j I[x \in B_j] \end{aligned}$$

where  $n_j =$  number of observations lying in bin  $B_j$ .

Though it is a very simple estimate, the histogram has many drawbacks, the main one is that we are estimating a continuous function by a non-smooth discrete function. It is not robust against the choice of origin  $x_0$  and bandwidth  $h$ . Also, it is not sensitive enough to local properties of  $f$ . Various density estimation techniques are proposed to overcome these drawbacks, one of which is the *kernel density estimation*.

### Kernel Density Estimation

We consider a specified *kernel function*  $K(\cdot)$  satisfying the conditions

- $\int_{-\infty}^{\infty} K(x)dx = 1$
- $K(\cdot)$  is symmetric around 0, giving  $\int_{-\infty}^{\infty} xK(x)dx = 0$
- $\int_{-\infty}^{\infty} x^2K(x)dx = \sigma^2(K) > 0$

and define the *kernel density estimator* by

$$\hat{f}_n(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right).$$

The estimate of  $f$  at point  $x$  is obtained using a weighted function of observations in the  $h$ -neighborhood of  $x$  where the weight given to each of the observations in the neighborhood depends on the choice of kernel function  $K(\cdot)$ . Some kernel functions are

- Uniform kernel:  $K(u) = \frac{1}{2}I[|u| \leq 1]$
- Triangle kernel:  $K(u) = (1 - |u|)I[|u| \leq 1]$
- Epanechnikov kernel:  $K(u) = \frac{3}{4}(1 - u^2)I[|u| \leq 1]$
- Gaussian kernel:  $K(u) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}u^2)$

The kernel density estimator satisfies the property

$$\int_{-\infty}^{\infty} \hat{f}_n(x)dx = 1$$

and on the whole gives a better estimate of the underlined density. Some of the properties are

- The kernel estimates do not depend on the choice of the origin, unlike histogram.
- The kernel density estimators are 'smoother' than the histogram estimators since they inherit the property of the smoothness of the kernel chosen.

- The kernel density estimator has a faster rate of convergence.
- Increasing the bandwidth is equivalent to increasing the amount of smoothing in the estimate. Very large  $h(\rightarrow \infty)$  will give an oversmooth estimate and  $h \rightarrow 0$  will lead to a needlepoint estimate giving a noisy representation of the data.
- The choice of the kernel function is not very crucial. The choice of the bandwidth, however, is crucial and the optimal bandwidth choice is extensively discussed and derived in the literature. For instance, with Gaussian kernel, the optimal (MISE) bandwidth is

$$h_{\text{opt}} = 1.06\sigma n^{-\frac{1}{5}}$$

where  $\sigma$  is the population standard deviation, which is estimated from the data.

- The kernel density estimation can be easily generalized from univariate to multivariate data in theory.

#### 4. Some Nonparametric Goodness-of-fit Tests

Though the samples are drawn from unknown populations, the investigators wish to confirm whether the data fit some proposed model. The Goodness-of-fit tests are useful procedures to confirm whether the proposed model satisfactorily approximates the observed situation. Apart from the usual Chi-Square goodness of fit test, we have Kolmogorov-Smirnov tests which are discussed here.

##### 4.1 One-sample Kolmogorov-Smirnov Test

This is a test of hypothesis that the sampled population follows some specified distribution.

Suppose we observe  $X_1, \dots, X_n$  i.i.d. from a continuous distribution function  $F(x)$ . We want to test the null hypothesis that  $F(x) = F_0(x)$  for all  $x$ , against the alternative that  $F(x) \neq F_0(x)$  for some  $x$ , where  $F_0$  is a distribution which is completely specified before we collect the data. Let  $\hat{F}_n(x)$  be the empirical distribution function

(e.d.f.) defined by

$$\widehat{F}_n(x) = \frac{1}{n} \sum_{i=1}^n I[X_i \leq x]$$

The one sample *Kolmogorov-Smirnov* (K-S) statistic is

$$M = \max_x \left| \widehat{F}_n(x) - F_0(x) \right|$$

A large value of  $M$  supports  $F(x) \neq F_0(x)$  and we reject the null hypothesis if  $M$  is too large.

It is not hard to show that the exact null distribution of  $M$  is the same for all  $F_0$ , but different for different  $n$ . Table of critical values are given in many books. For large  $n$

$$P(nM > q) \doteq 2 \sum_{i=1}^{\infty} (-1)^{i-1} \exp(-2i^2 q^2) \doteq 2 \exp(-2q^2)$$

Use of the last formula is quite accurate and conservative. Hence for a size  $\alpha$  test we reject  $H_0 : F(x) = F_0(x)$  if

$$nM > \left( -\frac{1}{2} \log \left( \frac{\alpha}{2} \right) \right)^{1/2} = M^\alpha$$

The K-S statistic is also called *K-S distance* since it provides a measure of closeness of  $\widehat{F}_n$  to  $F_0$ . It gives a method of constructing confidence band for the distribution which helps in identifying departures from the assumed distribution  $F_0$ , as we now show. First note that the distribution of

$$M(F) = \max_x \left| \widehat{F}_n(x) - F(x) \right|$$

is the same as null distribution for the K-S test statistic. Therefore

$$\begin{aligned} 1 - \alpha &= P(M(F) \leq M^\alpha) = P \left( \left| \widehat{F}_n(x) - F(x) \right| \leq \frac{M^\alpha}{n} \text{ for all } x \right) \\ &= P \left( F(x) \in \widehat{F}_n(x) \pm \frac{M^\alpha}{n} \text{ for all } x \right). \end{aligned}$$

One situation in which K-S is misused is in testing for normality. For K-S to be applied, the distribution  $F_0$  must be completely specified

before we collect the data. In testing for normality, we have to choose the mean and the variance based on the data. This means that we have chosen a normal distribution which is a closer to the data than the true  $F$  so that  $M$  is too small. We must adjust the critical value to adjust for this as we do in  $\chi^2$  goodness of fit tests. Lilliefors has investigated the adjustment of p-values necessary to have a correct test for this situation and shown that the test is more powerful than the  $\chi^2$  goodness of fit test for normality.

Other tests of this kind for testing  $F = F_0$  are the Anderson-Darling test based on the statistic

$$n \int_{-\infty}^{\infty} \left[ \widehat{F}_n(x) - F_0(x) \right]^2 [F_0(x)(1 - F_0(x))]^{-1} dF_0(x)$$

and the Cramer-von Mises test based on the statistic

$$\int_{-\infty}^{\infty} \left( \widehat{F}_n(x) - F_0(x) \right)^2 dF_0(x).$$

In addition, the Shapiro-Wilk test is specifically designed to test for normality.

### Kullback-Liebler Distance

Kolmogorov-Smirnov test can be used as a model selection test. However, if there are more than one distributions to choose from, a better measure of closeness between  $\widehat{F}_n(x)$  and  $F_0(x)$  is given by Kullback-Liebler distance, also called the *relative entropy*.

The Kullback-Liebler (K-L) distance between two distribution functions  $F$  and  $G$  with corresponding probability density functions  $f(\cdot)$  and  $g(\cdot)$  respectively is given by

$$KL(f, g) = \int_{-\infty}^{\infty} f(x) \log \frac{f(x)}{g(x)} d(x).$$

Note that  $KL(f, g) \geq 0$  and the equality holds for  $f(\cdot) = g(\cdot)$ . The K-L distance based on a sample of size  $n$  reduces to

$$\sum_{i=1}^n f(x_i) \log \frac{f(x_i)}{g(x_i)}.$$

For computing the distance between the empirical distribution function and the specified distribution function  $F_0$ , one can use the estimate of the density. Alternatively, for specific  $F_0$  such as Normal,

Uniform, Exponential etc, the entropy estimates are available in the literature. In addition, relative entropy based Goodness-of-fit tests are also discussed in the literature.

Broadly, to select the distribution which fits best, it is better to first screen the possible distributions using the Q-Q(P-P) plots and the K-S goodness-of-fit tests and then select one of the screened distributions based on the K-L distance.

#### 4.2 Two-sample Kolmogorov-Smirnov Test

Alternatively, one may be interested in verifying whether two independent samples come from identically distributed populations. Suppose we have two samples  $X_1, \dots, X_m$  and  $Y_1, \dots, Y_n$  from continuous distribution functions  $F(x)$  and  $G(y)$ . We want to test the null hypothesis that  $F(x) = G(x)$  for all  $x$  against the alternative that  $F(x) \neq G(x)$  for some  $x$ . Let  $\widehat{F}_n(x)$  and  $\widehat{G}_n(y)$  be the empirical distribution functions for the  $x$ 's and  $y$ 's. The two sample *Kolmogorov-Smirnov* (K-S) test is based on the statistic

$$M = \max_x \left| \widehat{F}_n(x) - \widehat{G}_n(x) \right|$$

We reject the null hypothesis if  $M$  is too large. As in the one sample case, if  $n$  and  $m$  are large,

$$P(dM > q) \doteq 2 \sum_{i=1}^{\infty} (-1)^{i-1} \exp(-2i^2q^2) \doteq 2 \exp(-2q^2)$$

(where  $d = 1/(\frac{1}{m} + \frac{1}{n})$ ) so that critical values may be determined easily.

### 5. Nonparametric Tests and Confidence Intervals

The nonparametric tests described here are often called distribution free procedures because their significance levels do not depend on the underlying model assumption i.e., they are level robust. They are also power robust and robust against outliers.

We will mainly discuss the so-called **rank procedures**. In these procedures, the observations are jointly ranked in some fashion. In using these procedures, it is occasionally important that the small ranks go with small observations, though often it does not matter

which order we rank in. The models for these procedures are typically semiparametric models.

One advantage of using ranks instead of the original observations is that the ranks are not affected by monotone transformations. Hence there is no need of transforming the observations before doing a rank procedure. Another advantage of replacing the observations with the ranks is that the more extreme observations are pulled in closer to the other observations.

As a consequence, a disadvantage is that nearby observations are spread out.

For example

<i>Obs</i>	1	1.05	1.10	2	3	100	1,000,00
<i>Rank</i>	1	2	3	4	5	6	7

The main reason we continue to study these rank procedures is the power of the procedures. Suppose the sample size is moderately large. If the observations are really normally distributed, then the rank procedures are nearly as powerful as the parametric ones (which are the best for normal data). In fact it can be shown that Pitman asymptotic relative efficiency (ARE) of the rank procedure to the parametric procedure is

$$3/\pi = .95$$

and in fact the ARE is always greater than  $3/\pi$ . However the ARE is  $\infty$  for some non-normal distributions. What this means is the rank procedure is never much worse than parametric procedure, but can be much better.

**Ties:**

We assume that the underlying probability distribution is continuous for the rank procedures and hence, theoretically, there are no ties in the sample. However, the samples often have ties in practice and procedures have been developed for dealing with these ties. They are rather complicated and not uniquely defined so we do not discuss them here. (refer Higgins for details).

### 5.1 Single Sample Procedures

We introduce the concept of *location parameter* first.

A population is said to be located at  $\mu_0$  if the population median is  $\mu_0$ .

Suppose  $X_1, \dots, X_n$  is a sample from the population. We say that  $X_1, \dots, X_n$  is located at  $\mu$  if  $X_1 - \mu, \dots, X_n - \mu$  is located at 0. Thus any statistic

$$S(\mu) = S(X_1 - \mu, \dots, X_n - \mu)$$

is useful for the location analysis if  $E[S(\mu_0)] = 0$  when the population is located at  $\mu_0$ . This simple fact leads to some test procedures to test the hypothesis of population locations.

### Sign Test

This is one of the oldest nonparametric procedures where the data are converted to a series of plus and minus signs. Let  $S(\mu)$  be the *sign statistic* defined by

$$\begin{aligned} S(\mu) &= \sum_{i=1}^n \text{sign}(X_i - \mu) \\ &= \#[X_i > \mu] - \#[X_i < \mu] \\ &= S^+(\mu) - S^-(\mu) \\ &= 2S^+(\mu) - n \end{aligned}$$

To find a  $\hat{\mu}$  such that  $S(\hat{\mu}) = 0$ , we get  $\hat{\mu} = \text{median}(X_i)$ . Thus if  $\mu_0$  is the median of the population, we expect  $E[S(\mu_0)] = 0$ .

Suppose we wish to test the hypothesis that the population median is  $\mu_0$  giving

$$H_0 : \mu = \mu_0 \quad \text{against} \quad H_1 : \mu \neq \mu_0.$$

Based on  $S(\mu_0)$ , the proposed decision rule is:

$$\text{Reject } H_0 \text{ if } |S(\mu_0)| = |2S^+(\mu_0) - n| \geq c$$

where  $c$  is chosen such that

$$P_{\mu_0}[|2S^+(\mu_0) - n| \geq c] = \alpha.$$

It is easy to see that under  $H_0 : \mu = \mu_0$ , the distribution of  $S^+(\mu_0)$  is Binomial $\left(n, \frac{1}{2}\right)$  irrespective of the underlined distribution of  $X_i$ 's and hence  $c$  can be chosen appropriately. Equivalently, we reject  $H_0$  if

$$S^+(\mu_0) \leq k \quad \text{or} \quad S^+(\mu_0) \geq n - k$$

where

$$P_{\mu_0}[S^+(\mu_0) \leq k] = \frac{\alpha}{2}.$$

This fact can be used to construct a confidence interval for the population median  $\mu$ . Consider

$$P_d[k < S^+(d) < n - k] = 1 - \alpha$$

and find the smallest  $d$  such that [the number of  $X_i > d$ ]  $< n - k$ . Suppose we get

$$\begin{aligned} d = X_{(k)} & : \#[X_i > X(k)] = n - k \\ d_{min} = X_{(k+1)} & : \#[X_i > X(k+1)] = n - k - 1. \end{aligned}$$

On the same lines, we find  $d_{max} = X_{(n-k)}$ . Then a  $(1 - \alpha)100\%$  distribution-free confidence interval for  $\mu$  is given by  $[X_{(k+1)}, X_{(n-k)}]$ . Note that the median is a robust measure of location and does not get affected by the outliers. The sign test is also robust and insensitive to the outliers and hence the confidence interval is robust too.

### Wilcoxon Signed Rank test

The sign test above utilizes only the signs of the differences between the observed values and the hypothesized median. We can use the signs as well as the ranks of the differences, which leads to an alternative procedure.

Suppose  $X_1, \dots, X_n$  is a random sample from an unknown population with median  $\mu$ . We assume that the population is symmetric around  $\mu$ . The hypothesis to be tested is  $\mu = \mu_0$  against the alternative that  $\mu \neq \mu_0$ .

We define  $Y_i = X_i - \mu_0$  and first rank the absolute values of  $|Y_i|$ . Let  $R_i$  be the rank of the absolute value of  $Y_i$  corresponding to the  $i^{th}$  observation,  $i = 1, \dots, n$ . The signed rank of an observation is the rank of the observation times the sign of the corresponding  $Y_i$ .

Let

$$S_i = \begin{cases} 1 & \text{if } (X_i - \mu_0) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

By arguments similar to the one mentioned for earlier test, we can construct a test using the statistic

$$WS = \sum_{i=1}^n S_i R_i.$$

$WS$  is called the *Wilcoxon signed rank statistic*.

Note that  $WS$  is the sum of ranks with positive signs of  $Y_i$ , i.e. the positive signed ranks. If  $H_0$  is true, the probability of observing a positive difference  $Y_i = X_i - \mu_0$  of a given magnitude is equal to the probability of observing a negative difference of the same magnitude. Hence, under the null hypothesis the sum of the positive signed ranks is expected to have the same value as that of the negative signed ranks. Thus a large or a small value of  $WS$  indicates a departure from the null hypothesis and we reject the null hypothesis if  $WS$  is too large or too small.

The critical values of the Wilcoxon Signed Rank test statistic are tabulated for various sample sizes. The tables of exact distribution of  $WS$  based on permutations is given in Higgins(2004).

### Normal approximation

It can be shown that for large sample, the null distribution of  $WS$  is approximately normal with mean  $\mu$  and variance  $\sigma^2$  where

$$\mu = \frac{n(n+1)}{4}, \quad \sigma^2 = \frac{n(n+1)(2n+1)}{24}$$

and the Normal cut-off points can be used for large values of  $n$ .

### Hodges-Lehmann confidence Interval for $\mu$

We can construct a  $(1 - \alpha)100\%$  confidence interval for population median  $\mu$  using Wilcoxon Signed rank statistic, under the assumption that the underlined population is symmetric around  $\mu$ .

Let

$$W_{ij} = \frac{X_i + X_j}{2}, \quad n \geq i \geq j \geq 1.$$

be the average of the  $i^{th}$  and  $j^{th}$  original observations, called a *Walsh average*.

For example, consider a single sample with 5 observations  $X_1, \dots, X_5$  given by  $-3, 1, 4, 6, 8$ . Then the Walsh averages are

		-3	1	4	6	8
-3	-3	-1	.5	1.5	2.5	
1		1	2.5	3.5	4.5	
4			4	5	6	
6				6	7	
8					8	

We order the  $W_{ij}$  according to their magnitude and let  $U_{[i]}$  be the  $i^{\text{th}}$  largest  $W_{ij}$ .

The median of  $W_{ij}$ 's provides a point estimation of the population median  $\mu$ . This median of Walsh averages is known as the *Hodges-Lehmann* estimator of the population median  $\mu$ .

For instance, in the data set above, the Hodges-Lehman estimator  $\hat{\mu}$  is the 8<sup>th</sup> largest Walsh average, namely  $\hat{\mu} = 3.5$  whereas the parametric estimate of  $\mu$  is  $\bar{X} = 3.2$ .

Using the Walsh averages, it is easy to see that another representation for the Wilcoxon Signed Rank statistic is

$$WS = \# [W_{ij} \geq 0]$$

(Note that this definition gives  $WS = 13$  for the example.)

Now suppose that we do not know  $\mu$ . Define

$$WS(\mu) = \# [W_{ij} \geq \mu]$$

Then the general distribution of  $WS(\mu)$  is the same as null distribution  $WS$  statistic.

Suppose that a size  $1 - \alpha$  two-sided Wilcoxon Signed Rank test for  $\mu = 0$  accepts the null hypothesis if

$$a \leq WS < b,$$

where  $a$  and  $b$  depend on  $\alpha$ . Then a  $(1 - \alpha)100\%$  confidence interval for  $\mu$  is

$$a \leq WS(\mu) < b \quad \Leftrightarrow \quad U_{[a]} < \mu \leq U_{[b]}$$

This confidence interval is called the *Hodges-Lehmann confidence interval* for  $\theta$

For the data above, it can be seen from the table values that the acceptance region for a  $\alpha = .125$  test is

$$2 \leq WS < 14$$

so that

$$U_{[2]} < \mu \leq U_{[14]} \quad \Leftrightarrow \quad -1 < \mu \leq 7$$

is a 87.5% confidence interval for  $\mu$ . Note that the assumed continuity implies that the inequality can be replaced by an equality in the last formula (but not the one before it) or vice versa.

Note that the H-L interval is associated with the Wilcoxon signed rank test in that the two-sided Wilcoxon test rejects  $\mu = 0$  iff 0 is not in the confidence interval. Also note that there is no problem with ties in either the H-L confidence interval or H-L estimator.

## 5.2 Two Sample Procedures

Suppose we observe two independent random samples  $X_1, \dots, X_n$  from distribution function  $F(x)$ , and  $Y_1, \dots, Y_n$  from distribution  $G(y)$  where both  $F$  and  $G$  are continuous distributions.

We discuss the nonparametric procedures for making inference about the difference between the two location parameters of  $F$  and  $G$  here. In particular, we make the assumption that the distribution functions of the two populations differ only with respect to the location parameter, if they differ at all. This can alternatively be stated by expressing  $G(y) = F(y + \delta)$  where  $\delta$  is the difference between the medians.

There is no assumption of symmetry in the two sample model. The continuity of the distributions implies there will be no ties.

### Wilcoxon rank sum statistic

Consider testing  $\delta = 0$  against  $\delta \neq 0$ . We first combine and jointly rank all the observations. Let  $R_i$  and  $S_j$  be the ranks associated with  $X_i$  and  $Y_j$ . Then we could compute a two-sample t based on these ranks. However, an equivalent test is based on

$$H = \sum_{i=1}^n R_i$$

Note that if  $\delta > 0$ , then the  $X_i$ 's should be greater than the  $Y_j$ 's, hence the  $R_i$ 's should be large and hence  $H$  should be large. A similar motivation works when  $\delta < 0$ . Thus we reject the null hypothesis  $H_0 : \delta = 0$  if  $H$  is too large or too small. This test is called the *Wilcoxon rank-sum test*.

Tables of exact distribution of  $H$  are available in Higgins (p 340).

For example, suppose we have two independent random samples of size 4 and 3. Suppose further that we observe 37, 49, 55, 57 in the

first sample and 23, 31, 46 in the second. We get

<i>obs</i>	37	49	55	57	23	31	46
<i>rank</i>	3	5	6	7	1	2	4

Therefore, for the observed data

$$H = 21$$

Again we reject if the observed  $H$  is one of the two largest or two smallest values. Based on the exact permutation distribution, we reject the null hypothesis as the p-value is  $2 \times 2/35 = .101$ .

### Normal approximation

It can be shown that for large sample the null distribution of  $H$  is approximately normal with mean  $\mu$  and variance  $\sigma^2$  where

$$\mu = \frac{m(m+n+1)}{2}, \quad \sigma^2 = \frac{mn(m+n+1)}{12}$$

Suppose, as above, we compute  $H = 21$  based on a samples of size 4 and 3. In this case  $\mu = 16$ ,  $\sigma^2 = 8$ , so the approximate p-value is (using a continuity correction)

$$\begin{aligned} 2P(H \geq 21) &= 2P(H \geq 20.5) = \\ 2P\left(\frac{Q - 16}{\sqrt{8}} \geq \frac{20.5 - 16}{\sqrt{8}}\right) &= 2P(Z \geq 1.59) = .11 \end{aligned}$$

which is close to the true p-value derived above even for this small sample size.

### Mann-Whitney test

Let

$$V_{ij} = X_i - Y_j,$$

We define

$$U = \#(V_{ij} > 0)$$

which is the *Mann-Whitney* statistic. The Mann-Whitney test rejects the null hypothesis  $H_0 : \delta = 0$  if  $U$  is too large or too small.

For our example we see that

	23	31	46
37	14	6	-9
49	26	18	3
55	32	24	9
57	34	26	11

Therefore, for this data set  $U = 11$ .

It can be shown that there is a relationship between the Wilcoxon rank sum  $H$  and the Mann-Whitney  $U$  :

$$H = U + \frac{n(n+1)}{2}.$$

Hence the critical values and p-values for  $U$  can be determined from those for  $H$ .

### The Hodges-Lehmann confidence interval for $\delta$

Analogous to the single sample procedure, we can construct a  $(1 - \alpha)100\%$  confidence interval for  $\delta$  using the Mann-Whitney procedure. We order  $V_{ij}$  according to their magnitude and let  $V_{[i]}$  be the  $i^{\text{th}}$  largest  $V_{ij}$ . Then the *Hodges Lehmann estimator* for  $\delta$  is the median of the  $V_{ij}$ .

Let

$$U(\delta) = \# [V_{ij} > \delta].$$

Then the general distribution of  $U(\delta)$  is the same as the null distribution of  $U$ . Suppose that two-sided size  $\alpha$  test the  $\delta = 0$  against  $\delta \neq 0$  accepts the null hypothesis if

$$a \leq U < b$$

Then a  $(1 - \alpha)100\%$  confidence region for  $\delta$  is given by

$$a \leq U(\delta) < b \quad \Leftrightarrow \quad V_{[a]} < \delta \leq V_{[b]}$$

which is the Hodges-Lehmann confidence interval for  $\delta$ . In our example the estimator is the average of the 6th and 7th largest of the  $V_{ij}$ , giving

$$\hat{\delta} = 16$$

The parametric estimator is  $\bar{X} - \bar{Y} = 16.2$ .

To find the confidence interval, note that  $H = U + 10$

$$.89 = P(12 \leq H < 21) = P(2 \leq U < 11)$$

Therefore the 89% Hodges-Lehmann confidence interval for  $\delta$  is

$$V_{[2]} \leq \delta < V_{[11]} \Leftrightarrow 3 \leq \delta < 32$$

The classical (t) confidence interval for the data based on t-statistics is  $1.12 < \delta \leq 31.22$ .

### Paired data

Analogous to the paired t-test in parametric inference, we can propose a nonparametric test of hypothesis that the median of the population of differences between pairs of observations is zero.

Suppose we observe a sequence of i.i.d. paired observations

$(X_1, Y_1), \dots, (X_n, Y_n)$ . Let  $\mu_D$  be the median of the population of differences between the pairs. The goal is to draw inference about  $\mu_D$ . Let

$$D_i = X_i - Y_i$$

The distribution of  $D_i$  is symmetric about  $\mu_D$ . Therefore, we may use the procedures discussed earlier for the one-sample model, based on the observations  $D_i$ .

### 5.3 $k$ -Sample Procedure

Suppose we wish to test the hypothesis that the  $k$  samples are drawn from the populations with equal location parameter. The Mann-Witney-Wilcoxon procedure discussed above can be generalized to  $k$  independent samples. The test procedure we consider is the *Kruskal-Wallis Test* which is the nonparametric analogue of the parametric one-way analysis of variance procedure.

Suppose we have  $k$  independent random samples of sizes  $n_i, i = 1, \dots, k$  each, represented by  $X_{ij}, j = 1, \dots, n_i; i = 1, \dots, k$ . Let the underlined location parameters be denoted by  $\mu_i, i = 1, \dots, k$ . The null hypothesis to test is that the  $\mu_i$  are all equal against the alternative that at least one pair  $\mu_i, \mu_{i^*}$  is different.

For the Kruskal Wallis test procedure, we combine the  $k$  samples and rank the observations. Let  $R_{ij}$  be the rank associated with  $X_{ij}$  and let  $\bar{R}_i$  be the average of the ranks in the  $i^{th}$  sample. If the null

hypothesis is true, the distribution of ranks over different samples will be random and no sample will get a concentration of large or small ranks. Thus under the null hypothesis, the average of ranks in each sample will be close to the average of ranks for under the null hypothesis.

The Kruskal-Wallis test statistic is given by

$$KW = \frac{12}{N(N+1)} \sum n_i \left( \bar{R}_i - \frac{N+1}{2} \right)^2$$

If the null hypothesis is not true, the test statistic  $KW$  is expected to be large and hence we reject the null hypothesis of equal locations for large values of  $KW$ .

The tables of exact critical values are available in the literature. We generally use a  $\chi^2$  distribution with  $k - 1$  degrees of freedom as an approximate sampling distribution for the statistic.

## 6. Permutation tests

The parametric test statistics can also be used to carry out the non-parametric test procedures. The parametric assumptions determine the distribution of the test statistic and hence the cut-off values under the null hypothesis. Instead, we use permutation tests to determine the cutoff points.

We give an example below.

Consider a two sample problem with 4 observations  $X_1, X_2, X_3, X_4$  in the first sample from cdf  $F(x)$  and 3 observations  $Y_1, Y_2, Y_3$  in the second sample from cdf  $G(y)$ . We want to test the null hypothesis  $F(x) = G(x)$  against the alternative hypothesis  $F(x) \neq G(x)$ .

Suppose we observe 37, 49, 55, 57 in the first sample and 23, 31, 46 in the second (Section 5.2). Suppose we want a test with size .10.

1. The parametric test for this situation is the two-sample t-test which rejects if

$$|T| = \left| \frac{\bar{X} - \bar{Y}}{S_p \sqrt{\frac{1}{4} + \frac{1}{3}}} \right| > t_5^{.05} = 2.015$$

For this data set,  $T = 2.08$  so we reject (barely). The p-value for these data is .092. Note that this analysis depends on the

assumptions that the data are normally distributed with equal variances.

2. We now look at rearrangements of the data observed. One possible rearrangement is 31, 37, 46, 55 in the first sample and 23, 49, 57 in the second. For each rearrangement, we compute the value of the  $T$ . Note that there are

$$\binom{7}{4} = 35$$

such rearrangements. Under the null hypothesis (that all 7 observations come from the same distribution) all 35 rearrangements are equally likely, each with probability  $1/35$ . With the permutation test, we reject if the value of  $T$  for the original data is one of the 2 largest or 2 smallest. This test has  $\alpha = 4/35 = .11$ . The p-value for the permutation test is twice the rank of the original data divided by 35.

3. If we do this to the data above, we see that the original data gives the second largest value for  $T$ . (Only the rearrangement 46, 49, 55, 57 and 23, 31, 37 gives a higher  $T$ .) Therefore we reject the null hypothesis. The p-value is  $2 \times 2/35 = .11$ . Note that the only assumption necessary for these calculations to be valid is that under the null hypothesis the two distributions be the same (so that each rearrangement is equally likely). That is, the assumptions are much lower for this nonparametric computation.

These permutation computations are only practical for small data sets. For the two sample model with  $m$  and  $n$  observations in the samples, there are

$$\binom{m+n}{m} = \binom{m+n}{n}$$

possible rearrangements. For example

$$\binom{20}{10} = 184,756$$

so that if we had two samples of size 10, we would need to compute  $V$  for a total of 184,756 rearrangements. A recent suggestion is that we

don't look at all rearrangements, but rather look a randomly chosen subset of them and estimate critical values and p-values from the sample.

What most people who use these tests would do in practice is use the t-test for large samples, where the t-test is fairly robust and use the permutation calculation in small samples where the test is much more sensitive to assumptions.

## 7. Correlation coefficients

### Pearson's r

The parametric analysis assumes that we have a set of i.i.d. two-dimensional vectors,  $(X_1, Y_1), \dots, (X_n, Y_n)$  which are normally distributed with correlation coefficient

$$\rho = \frac{\text{cov}(X_i, Y_i)}{\sqrt{\text{var}(X_i) \text{var}(Y_i)}}.$$

$\rho$  is estimated by the sample correlation coefficient (Pearson's r)

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$$

The null hypothesis  $\rho = 0$  is tested with the test statistic

$$t = \sqrt{\frac{n-2}{1-r^2}} r \sim t_{n-2}$$

under the null hypothesis.

To make this test more robust, we can use a permutation test to get nonparametric critical values and p-values. To do the rearrangements for this test, we fix the  $X$ 's and permute the  $Y$ 's.

### Some Semiparametric correlation coefficients

A semiparametric model alternative for the normal correlation model above is to assume that the  $(X_1, Y_1), \dots, (X_n, Y_n)$  are i.i.d. from a continuous bivariate distribution, implying no ties.

### Spearman's rank correlation

We rank the X's and Y's separately getting ranks  $R_i$  and  $S_i$ . The sample correlation coefficient between the  $R_i$  and  $S_i$  is called *Spearman's rank correlation*. Suppose, for example the we observe

$x$	1	3	6	9	15
$r$	1	2	3	4	5
$y$	1	9	36	81	225
$s$	1	2	3	4	5

Then the rank correlation  $r_S$  is obviously one. Note that this happens because  $Y = X^2$ . Since  $Y$  is not a linear function of  $X$ , the correlation coefficient is less than 1. In fact the correlation coefficient is .967. We often want to test that  $X$  and  $Y$  are independent. We reject if  $r_S$  is too large or too small. We determine the critical values and p-values from the permutation test as described above. For reasonably large sample sizes, it can be shown that under the null hypothesis

$$r_S \overset{\bullet}{\sim} N\left(0, \frac{1}{n-1}\right)$$

### Kendall's coefficient of concordance

We say two of the vectors  $(X_i, Y_i)$  and  $(X_{i*}, Y_{i*})$  are concordant if

$$(X_i - Y_i)(X_{i*} - Y_{i*}) > 0$$

Kendall's  $\tau$  is defined by

$$\tau = 2P[(X_i - Y_i)(X_{i*} - Y_{i*}) > 0] - 1$$

We estimate Kendall's  $\tau$  by

$$r_K = 2 \frac{\#(\text{concordant pairs})}{\binom{n}{2}} - 1$$

To test  $\tau = 0$ , we would use  $r_K$ . One and two sided (exact) critical values can be determined from permutation arguments. Approximate critical value and p-values can be determined from the fact that for reasonably large  $n$ , the null distribution is

$$r_K \overset{\bullet}{\sim} N\left(0, \frac{4n+10}{9(n^2-n)}\right).$$

## 8 Nonparametric Regression

Suppose we have  $n$  observations  $(Y_1, X_1), \dots, (Y_n, X_n)$  where  $Y$  is the response variable and  $X$  is the predictor variable. The aim is to model  $Y$  as a function of  $X$ , which is helpful in predicting future values of  $Y$  and in constructing tests and interval estimates for predictions and parameters.

The most widely used statistical procedure for such a problem is *linear regression model* where we assume that  $E[Y|X = x]$  is a linear function of  $X$ , specified by

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i, \quad i = 1, \dots, n,$$

with the errors  $\epsilon_i$  taken to be uncorrelated with zero mean and variance  $\sigma^2$ . When not appropriate, fitting a linear regression model to a nonlinear relationship results in a totally misleading and unreliable inference.

Nonparametric regression is a more general model alternative where no parametric model is assumed. In particular, the model considered is

$$Y_i = m(X_i) + \epsilon_i$$

where the regression curve  $m(x)$  is the conditional expectation  $m(x) = E[Y|X = x]$  with  $E[\epsilon|X = x] = 0$  and  $\text{Var}[\epsilon|X = x] = \sigma^2(x)$ . The model removes the parametric restrictions on  $m(x)$  and allows the data to dictate the alternative structure of  $m(x)$  by using the data based estimate of  $m(X)$ .

Different estimation procedures lead to different nonparametric regression models. We briefly discuss the *Kernel Regression* here.

### Kernel Regression

We have

$$\begin{aligned} m(x) &= E[Y|X = x] \\ &= \int y \frac{f(x, y)}{f(x)} dy \end{aligned}$$

where  $f(x)$  and  $f(x, y)$  are the marginal density of  $X$  and the joint density of  $X$  and  $Y$  respectively. On substituting the univariate and bivariate kernel density estimates of the two densities and noting the

properties of kernel function  $K(\cdot)$  specified in Section 3, we get

$$\begin{aligned}\hat{m}_{NW}(x) &= \frac{\sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) Y_i}{\sum_{i=1}^n K\left(\frac{x - X_i}{h}\right)} \\ &\equiv \sum_{i=1}^n W_{hi} Y_i\end{aligned}$$

which is a linear function of  $Y$  with weights

$$W_{hi} = (nh)^{-1} \frac{K\left(\frac{x - X_i}{h}\right)}{\hat{f}(x)}.$$

This is called the *Nadaraya-Watson kernel estimator*. Note that

- The weights depend on the kernel function  $K(\cdot)$ , the bandwidth  $h$  and the whole sample  $\{X_i, i = 1, \dots, n\}$  through the kernel density estimate  $\hat{f}(x)$ .
- For the uniform kernel, the estimate of  $m(X_i) = E[Y_i | X = X_i]$  is the average of  $Y_j$ 's corresponding to the  $X_j$ 's in the  $h$  neighborhood of  $X_i$ .
- Observations  $Y_i$  obtain more weight in those areas where the corresponding  $X_i$  are sparse.
- When the denominator is zero, the numerator is also equal to zero and the estimate is set to be zero.
- Analogous to kernel density estimation, the bandwidth  $h$  determines the level of smoothness of the estimate. Decreasing bandwidth leads to a less smooth estimate. In particular, for  $h \rightarrow 0$  the estimate  $\hat{m}(X_i)$  converges to  $Y_i$  and for  $h \rightarrow \infty$  the estimate converges to  $\bar{Y}$ . The criteria of bandwidth selection and guidelines for selecting the optimal bandwidth are available in the literature.

In case the predictors  $X_i, i = 1, \dots, n$  are not random, alternative estimators such as *Gasser-Müller kernel estimator* are more appropriate.

It can be shown that the Nadaraya-Watson kernel estimator is the solution of the weighted least squares estimator obtained on minimizing

$$\sum_{i=1}^n (Y_i - \beta_0)^2 K\left(\frac{x - X_i}{h}\right)$$

over  $\beta_0$ . This corresponds to locally approximating  $m(X)$  with a constant while giving higher weight to the values of  $Y$  corresponding to the  $X_i$ 's in the neighborhood of  $X$ . This led to further generalizations where higher order local polynomial fitting is attempted. In particular, we consider minimizing

$$\sum_{i=1}^n [Y_i - \beta_0 - \beta_1(x - X_i) - \beta_2(x - X_i)^2 - \dots - \beta_p(x - X_i)^p]^2 K\left(\frac{x - X_i}{h}\right)$$

over  $\beta_0, \beta_1, \dots, \beta_p$ . The resulting estimator is called *local polynomial regression estimator* and the appropriate choice of the degree of polynomial  $p$  can be made based on the data.

## References

1. Arnold, Steven (1990), *Mathematical Statistics* ( Chapter 17). Prentice Hall, Englewood Cliffs, N. J
2. Beers, Flynn, and Gebhardt (1990), Measures of Location and Scale for Velocities in Cluster of Galaxies-A Robust Approach. *Astron Jr*, 100, 32-46.
3. Härdle, W. (1990) *Applied Nonparametric Regression*. Cambridge Univ Press, Cambridge.
4. Hettmansperger, T and McKean, J (1998), *Robust nonparametric Statistical Methods*. Arnold, London.
5. Higgins, James (2004), *Introduction to Modern Nonparametric Statistics*. Duxbury Press.
6. Hollander and Wolfe, (1999), *Nonparametric Statistical Methods* John Wiley, N.Y.
7. Johnson, Morrell, and Schick (1992), Two-Sample Nonparametric Estimation and Confidence Intervals Under Truncation, *Biometrics*, 48, 1043-1056.

8. Summer School in Statistics for Astronomers(2005). Lecture Notes by Steven Arnold Website: <http://astrostatistics.psu.edu/>
  9. Summer School in Statistics for Astronomers(2008). Lecture Notes by Tom Hettmansperger. Nonparametrics.zip. Website: <http://astrostatistics.psu.edu/>
  10. Website: <http://www.stat.wmich.edu/slab/RGLM/>
- \* Part of these notes borrow heavily from the material in 8 and 9

# Chapter 11

## NON-PARAMETRIC STATISTICS WITH R

*Notes by Arnab Chakraborty*

# Nonparametric statistics

This is the practical lab tutorial for nonparametric statistics. But we shall start with a little theory that will be crucial for understanding the underlying concepts.

## Distributions

In statistics we often come across questions like

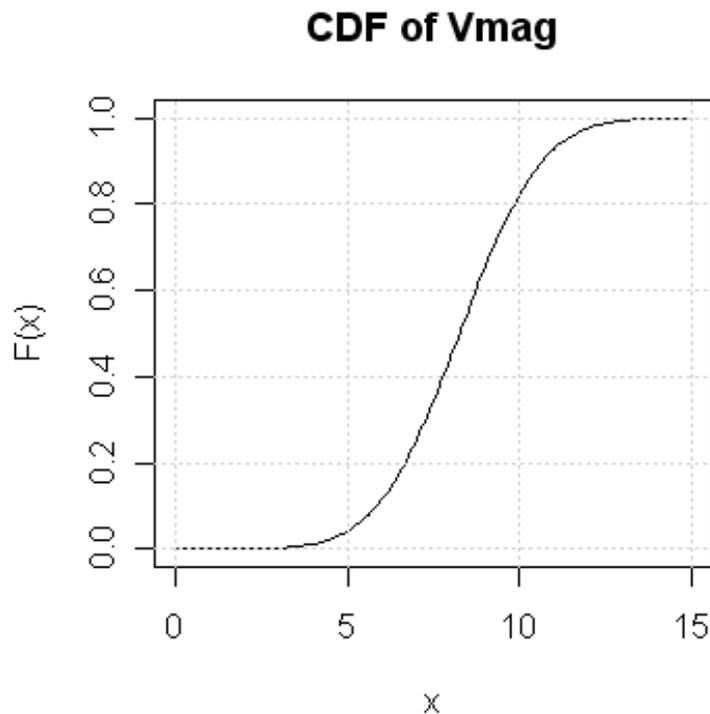
“Is a given data from such-n-such distribution?”

Before we can answer this question we need to know what is meant by a *distribution*. We shall talk about this first.

In statistics, we work with random variables. Chance controls their values.

The **distribution** of a random variable is the rule by which chance governs the values. If you know the distribution then you know the chance of this random variable taking value in any given range. This rule may be expressed in various ways. One popular way is via the **Cumulative Distribution Function (CDF)**, that we illustrate now. If a random variable ( $X$ ) has distribution with CDF  $F(x)$  then for any given number  $a$  the chance that  $X$  will be  $\leq a$  is  $F(a)$ .

**Example:** Suppose that I choose a random star from the Hipparcos data set. If I tell you that its  $v_{mag}$  is a random variable with the following CDF then what is the chance that it takes values  $\leq 10$ ? Also find out the value  $x$  such that the chance of  $v_{mag}$  being  $\leq x$  is 0.4.



For the first question the required probability is  $F(10)$  which, according to the graph, is slightly above 0.8.

In the second part we need  $F(x) = 0.4$ . From the graph it seems to be about 7.5.

Just to make sure that these are indeed meaningful, let us load the Hipparcos data set and check:

```
hip = read.table("HIP.dat", head=T)
attach(hip)
n = length(Vmag) #total number of cases
count = sum(Vmag<=10) #how many <= 10
count/n #should be slightly above 0.8
```

This checks the answer of the first problem. For the second

```
count = sum(Vmag<=7.5) #how many <= 7.5
count/n #should be around 0.4
```

So you see how powerful a CDF is: in a sense it stores as much information as the entire `Vmag` data. It is a common practice in statistics to regard the distribution as the *ultimate truth* behind the data. We want to infer about the underlying distribution based on the data.

R has many standard distributions already built into it. This basically means that R has functions to compute their CDFs.

These functions all start with the letter **p**.

**Example:** A random variable has standard Gaussian distribution. We know that R computes its CDF using the function **pnorm**. How to find the probability that the random variable takes values  $\leq 1$ ?

The answer may be found as follows:

```
| pnorm(1)
```

For every **p**-function there is a **q**-function that is basically its inverse.

**Example:** A random variable has standard Gaussian distribution. Find  $x$  such that the random variable is  $\leq x$  with chance 0.3.

Now we shall use the function **qnorm**:

```
| qnorm(0.3)
```

OK, now that we are through our little theory session, we are ready for the nonparametric lab.

## One sample nonparametric tests

### Sign-test

In a sense this is the simplest possible of all tests. Here we shall consider the data set LMC.dat that stores the measured distances to the Large Magellanic Cloud.

```
| LMC = read.table("LMC.dat",head=T)
| data = LMC[,2] #These are the measurements
| data
```

We want to test if the measurements exceed 18.41 *on an average*. Now, this does *not* mean whether the average of the data exceeds 18.41, which is trivial to find out. The question here is actually about the underlying distribution. We want to know if the median of the underlying distribution exceeds 18.41, which is a less trivial question, since we are not given that distribution.

We shall use the sign test to see if the median is 18.41 or larger. First it finds how many of the observations are above this value:

```
| abv = sum(data>18.41)
```

```
|abv
```

Clearly, if this number is large, then we should think that the median (of the underlying distribution) exceeds 18.41. The question is how large is "large enough"? For this we consult the binomial distribution to get the **p**-value. The rationale behind this should come from the theory class.

```
|n = nrow(LMC)
|pValue = 1-pbinom(abv-1, n, 0.5)
|pValue
```

We shall learn more about **p**-values in a later tutorial. For now, we shall use the following rule of thumb:

If this **p**-value is below 0.05, say, we shall conclude that the median is indeed larger than 18.41.

### Wilcoxon's Signed Rank Test

As you should already know from the theoretical class, there is a test called Wilcoxon's Signed Rank test that is better (if a little more complicated) than the sign test. R provides the function **wilcox.test** for this purpose. We shall work with the Hipparcos data set this time, which we have already loaded. We want to see if the median of the distribution of the `pmRA` variable is 0 or not.

```
|wilcox.test(pmRA, mu=0)
```

Since the **p**-value is pretty large (above 0.05, say) we shall conclude that the median is indeed equal to 0. R itself has come to the same conclusion.

Incidentally, there is a little caveat. For Wilcoxon's Rank Sum Test to be applicable we need the underlying distribution to be symmetric around the median. To get an idea about this we should draw the histogram.

```
|hist(pmRA)
```

Well, it looks pretty symmetric (around a centre line). But would you apply Wilcoxon's Rank Sum Test to the variable `DE`?

```
|hist(DE)
```

No, this is not at all symmetric!

### Kolmogorov-Smirnov test

The theory class has talked about a non-parametric test called Kolmogorov-Smirnov test. We shall cover it in a later tutorial.

## k-nearest neighbours

Imagine that you are given a data set of average parent heights and their adult sons' heights, like this (ignore the colour for the time being):

Parent	Son
5.5	5.9
5.4	5.3
5.7	5.9
5.1	5.3
6.0	5.8
5.5	5.2
6.1	6.0

A prospective couple comes to you, reports that their average height is 5.3 feet, and wants you to predict (based on this data set) the height their son would attain at adulthood. How should you proceed?

A pretty simple method is this: look at the cases in your data set with average parents' height near 5.3. Say, we consider the 3 nearest cases. These are shown in red in the above data set. Well, there are 4 cases, and not 3. This is because there is a **tie** (three cases 5.1, 5.5, 5.5 that are at same distance from 5.3).

Now just report the average of the sons' height for these cases.

Well, that is all there is to it in 3-NN regression (NN = Nearest Neighbour)! Let us implement this in R. First the data set

```
parent = c(5.5, 5.4, 5.7, 5.1, 6.0, 5.5, 6.1)
son = c(5.9, 5.3, 5.9, 5.3, 5.8, 5.2, 6.0)
```

Now the new case:

```
newParent = 5.3
```

Find the distances of the parents' heights from this new height:

```
d = abs(parent - newParent)
```

Rank these:

```
rnk = rank(d, tie="min")
rnk
```

Notice the `tie="min"` option. This allows the three joint seconds to be each given rank 2. We are not using `order` here (as it does not handle ties gracefully). Now identify the 3 nearest cases (or more in case of ties).

```
nn = (rnk<=3)
nn
```

Now it is just a matter of taking averages.

```
newSon = mean(son[nn])
newSon
```

It is instructive to write a function that performs the above computation.

```
knnRegr = function(x, y, newX, k) {
  d = abs(x-newX)
  rnk = rank(d, tie="min")
  mean(y[rnk<=k])
}
```

Now we can use it like

```
newSon = knnRegr(parent, son, newParent, 3)
```

**Exercise:** You shall apply the function that we have just written to the Hipparcos data set.

```
plot(RA, pmRA)
```

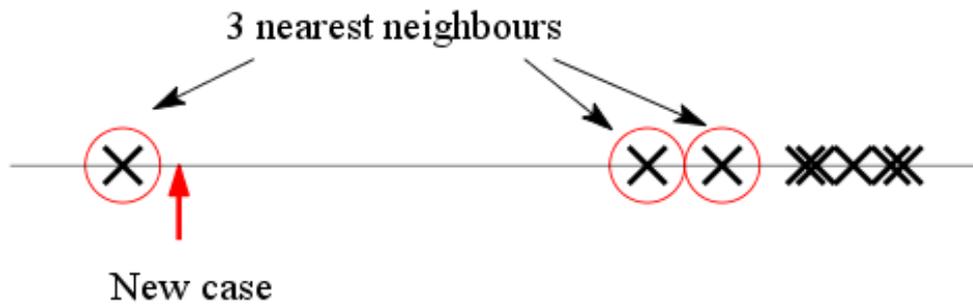
We want to explore the relation between `RA` and `pmRA` using the `k`-nearest neighbour method. Our aim is to predict the value `pmRA` based on a new value of `RA`:

```
newRA = 90.
```

Use the `knnRegr` function to do this with `k=13`.

## Nadaraya-Watson regression

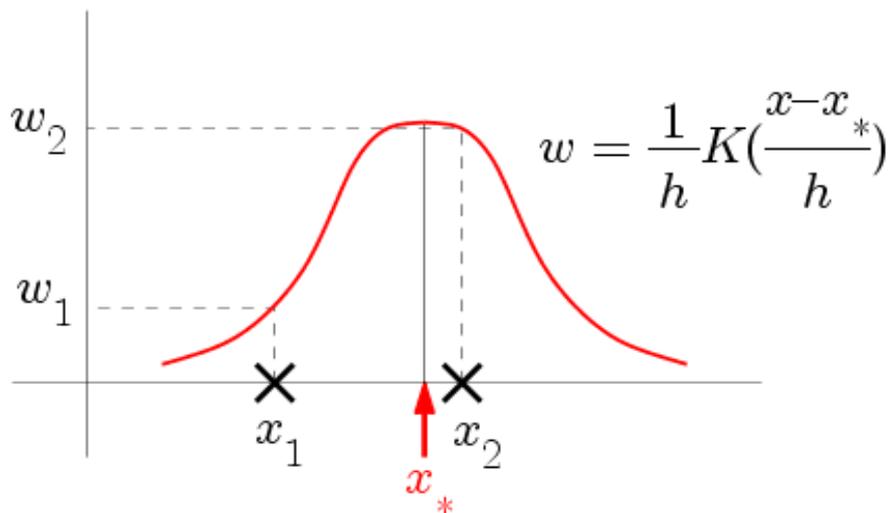
While the `k`-nearest neighbour method is simple and reasonable it is nevertheless open to one objection that we illustrate now. Suppose that we show the `x`-values along a number line using crosses as below. The new `x`-value is shown with an arrow. The 3 nearest neighbours are circled.



### Are the 3 NN equally relevant?

The  $k$ -NN method now requires us to simply average the  $y$ -values of the circled cases. But since two of the points are rather far away from the arrow, shouldn't a weighted average be a better method, where remote points get low weights?

This is precisely the idea behind the Nadaraya-Watson regression. Here we average over *all* the cases (not just nearest neighbours) but cases farther away get less weight. The weights are decided by a **kernel** function (typically a function peaked at zero, and tapering off symmetrically on both sides).



### How the kernel determines the weights

Now we may simply take the weighted average. We shall apply it to our parent-son example first.

```
parent = c(5.5, 5.4, 5.7, 5.1, 6.0, 5.5, 6.1)
son = c(5.9, 5.3, 5.9, 5.3, 5.8, 5.2, 6.0)
newParent = 5.3
kernel = dnorm #this is the N(0,1) density
```

Now let us find the weights (for a bin width ( $h$ ) = 0.5, say):

```
h = 0.5
wt = kernel((parent - newParent)/h)/h
```

Finally the weighted average:

```
newSon = weighted.mean(son,w = wt)
```

**Exercise:** Write a function called, say, `nw`, like this

```
nw = function(x,y,newX, kernel,h) {
  #write commands here
}
```

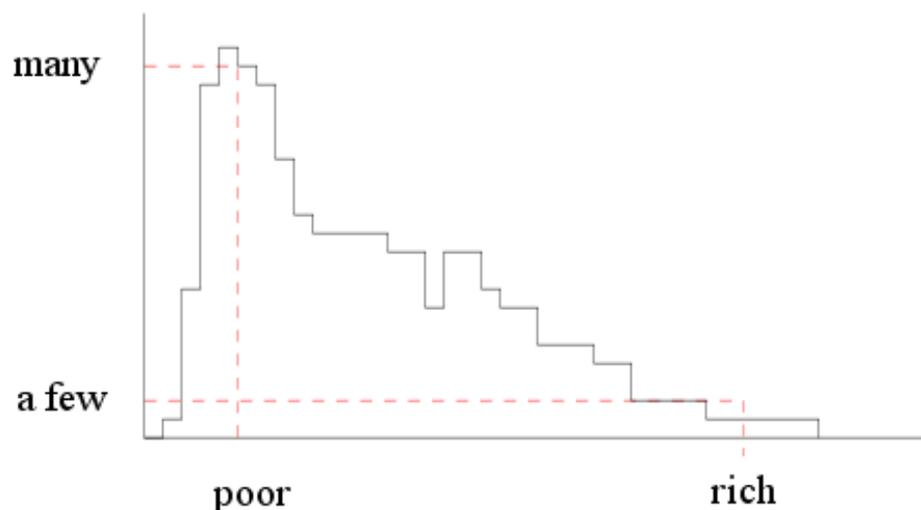
[Hint: Basically collect the lines that we used just now.]

Now use it to predict the value of `pmRA` when `RA` is 90 (use `h=0.2`):

```
newRA = 90
newpmRA = nw(RA,pmRA,newRA,dnorm,0.2)
```

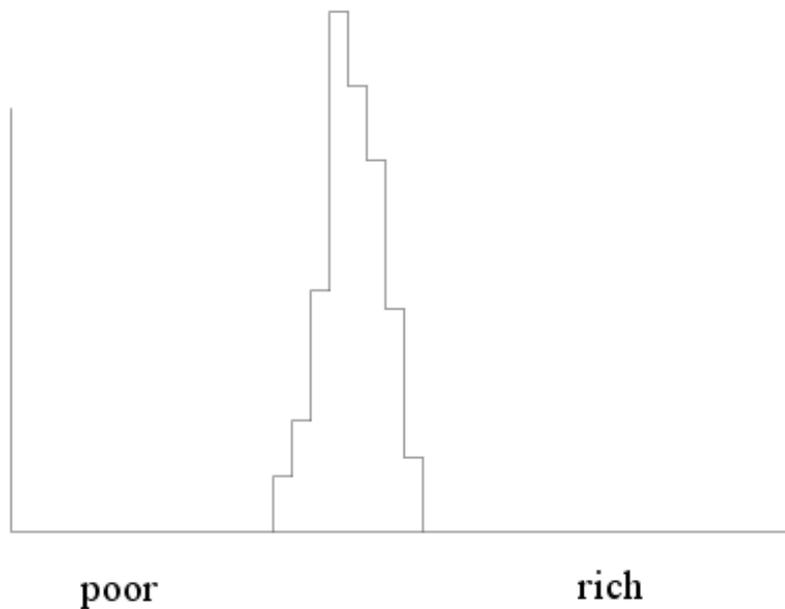
## Two-sample nonparametric tests

Here we shall be taking about the shape of distributions. Let us first make sure of this concept. Suppose that I make a list of all the people in a capitalist country and make a histogram of their incomes. I should get a histogram like this



**Income histogram of a capitalist country**

But if the same thing is done for a socialist country we shall see something like this (where almost everybody is in the middle income group).



**Income histogram of a socialist country**

Clearly, the *shapes* of the histograms differ for the two populations. We must be careful about the term "shape" here. For example, the following two histograms are for two capitalist countries, one rich the other poor.



**Income histograms two capitalist countries**

Here the histograms have the same *shape* but differ in *location*. You can get one from the other by applying a shift in the location.

This is a very common phenomenon in statistics. If we have two populations with comparable structure they often have similar shapes, and differ in just a location shift. In such a situation we are interested in knowing about the amount of shift. If the amount of shift is zero, then the two populations behave identically.

Wilcoxon's rank sum test is one method to learn about the amount of shift based on samples from the two populations.

We shall start with the example done in the theory class, where we had two samples

$$|x = c(37, 49, 55, 57)$$

```
y = c(23, 31, 46)
m = length(x)
n = length(y)
```

We are told that both the samples come from populations with the same *shape*, though one may be a shifted version of the other.

Our aim is to check if indeed there is any shift or not.

We shall consider the pooled data (*i.e.*, all the 7 numbers taken together) and rank them

```
pool = c(x, y)
pool
r = rank(pool)
r
```

The first  $m=4$  of these ranks are for the first sample. Let us sum them:

```
H = sum(r[1:m])
H
```

If the the two distributions are really identical (*i.e.*, if there is no shift) then we should have (according to the theory class)  $H$  close to the value

```
m*(m+n+1)/2 #Remember: * means multiplication
```

Can the  $H$  that we computed from our data be considered "close enough" to this value? We shall let us R determine that for us.

```
wilcox.test(x, y)
```

So R clearly tells us that there is a shift in location. The output also mentions a  $W$  and a  $p$ -value. tutorial. But  $W$  is essentially what we had called  $H$ . Well,  $H$  was 21, while  $W$  is 11. This is because R has the habit of subtracting

$$m(m+1)/2$$

from  $H$  and calling it  $W$ . Indeed for our data set

$$m(m+1)/2 = 10,$$

which perfectly accounts for the difference between our  $H$  and R's  $W$ .

You may recall from the theory class that  $H$  is called Wilcoxon's rank sum statistic, while  $W$  is the Mann-Whitney statistic (denoted by  $U$  in class).

Now that R has told us that there is a shift, we should demand to estimate the amount of shift.

```
wilcox.test(x, y, conf.int=T)
```

The output gives two different forms of estimates. It gives a single value (a **point estimate**) which is 16. Before it gives a 95% **confidence interval**: -9 to 34. This basically means that

we can say with 95% confidence that the true value of the shift is between -9 and 34.

We shall talk more about confidence intervals later. For now let us see how R got the point estimate 16. It used the so-called Hodges-Lehman formula, that we have seen in the theoretical class. To see how this method works we shall first form the following "difference table"

	23	31	46
37	14	6	-9
49	26	18	3
55	32	24	9
57	34	26	11

Here the rows are headed by the first sample, and columns by the second sample. Each entry is obtained by subtracting the column heading from the row heading (e.g.,  $-9 = 37 - 46$ ). This table, by the way, can be created very easily with the function **outer**.

```
outer(x, y, "-")
```

 The **outer** function is a very useful (and somewhat tricky!) tool in R. It takes two vectors **x**, and **y**, say, and some function **f(x,y)**. Then it computes a matrix where the (i,j)-th entry is

$$f(x[i], y[j]).$$

Now we take the median of all the entries in the table:

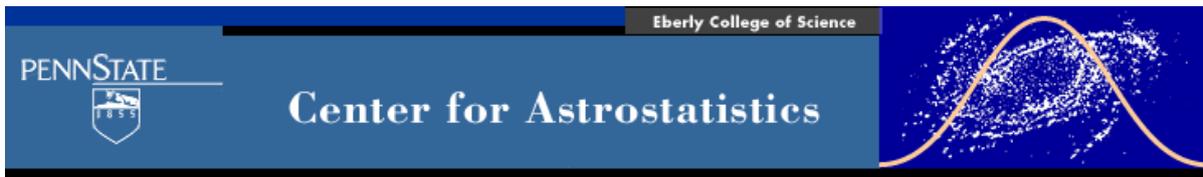
```
median(outer(x, y, "-"))
```

This gives us the Hodges-Lehman estimate of the location shift.

# Chapter 12

## ANALYSIS OF DATACUBES

NOTES BY JOGESH BABU



# Analysis of astronomical datacubes

G. Jogesh Babu

<http://astrostatistics.psu.edu>  
Penn State University

Most 20-th century astronomical data have been 2-dimensional images or 1-dimensional spectra/time series. But 3-dimensional hyperspectral images and videos are becoming increasingly prevalent:

1. Datacubes from radio interferometers  
(once restricted to 21-cm and small molecular line maps, nearly all data from the EVLA and ALMA will be 3-dim spectro-image datacubes)
2. Integral Field Units spectrographs  
(bundled-fiber-fed spectrographs give spectro-image cubes)
3. Multiepoch visible-light surveys  
(Palomar QUEST, Pan-STARRS, LSST, etc. produce huge datasets of time-image video-like cubes)

*Extensive methodology developed in other fields for 3D analysis: digital video processing, remote sensing, animation, etc*

## Astronomical datacubes today

All major new telescopes produce datacubes as their primary data products: ALMA, ELVA, ASKAP, MeerKAT, LOFAR, SKA.

For many observations, ALMA will produce 1-100 GBy datacubes consisting of ~1-10 million spatial pixels with 1-10 thousand frequency channels. Over a decade, ALMA produces petabytes of datacubes.

### Main interest here is in LSST-type video problems

- There are some differences from radio. E.g., night-to-night differences in PSF shape that affects the unresolved sources, whereas radio has RFI autocorrelated noise across some 2D images.
- The overall approach of computationally efficient, intermediate-level statistics (e.g. Mahalanobis distance) and computer vision techniques holds in both cases

### Data analysis & science goals include:

#### a. Faint continuum source detection

Source catalogs (logN-LogS), multiwavelength & transient studies

#### b. Faint spectral lines & transient detection

Redshifted Ly $\alpha$  & HI proto/dwarf galaxies, SN Ia surveys, orphan GRBs, etc.

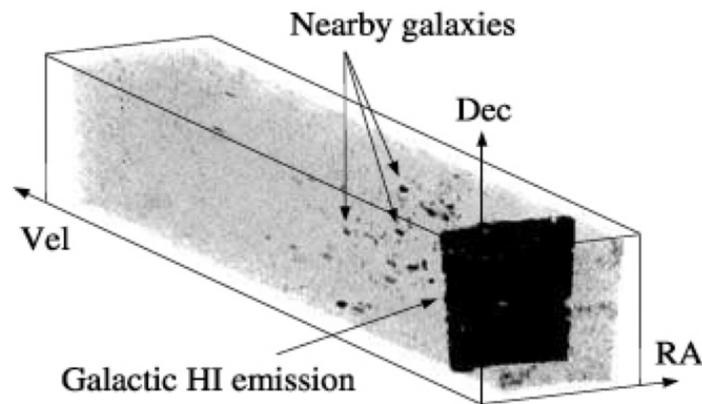
#### c. Characterization of bright features

Photometry, galaxy morphology, radio jets/lobes, molecular clouds, etc.

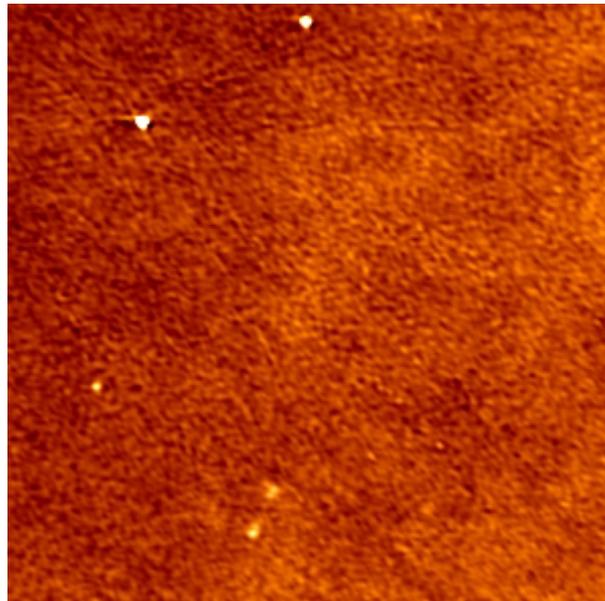
*Methodologists have to think very broadly  
about all types of data cubes*

## Bump hunting in radio astronomical datacubes

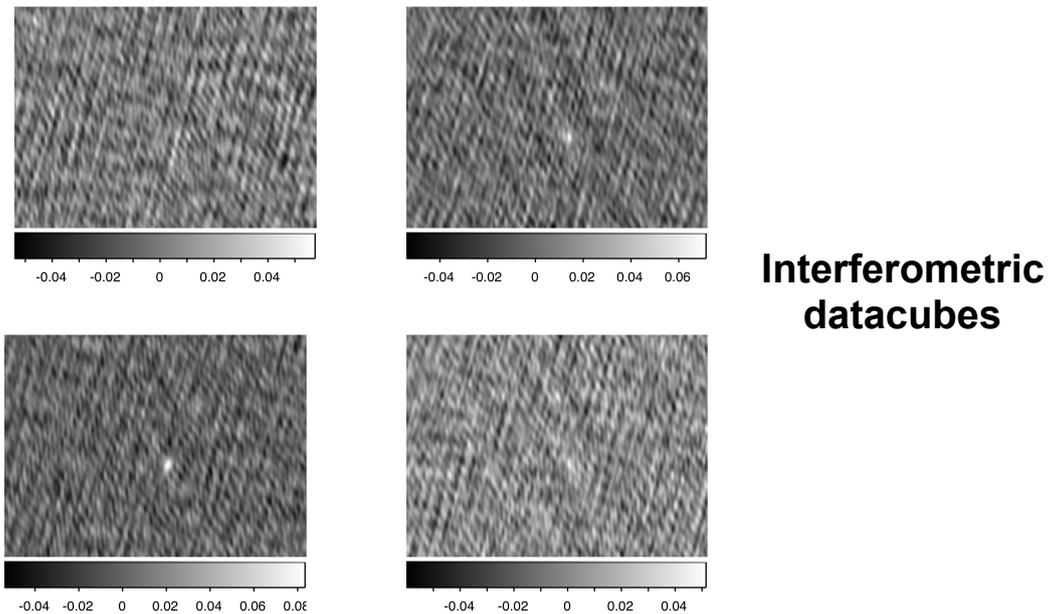
### Single-dish data



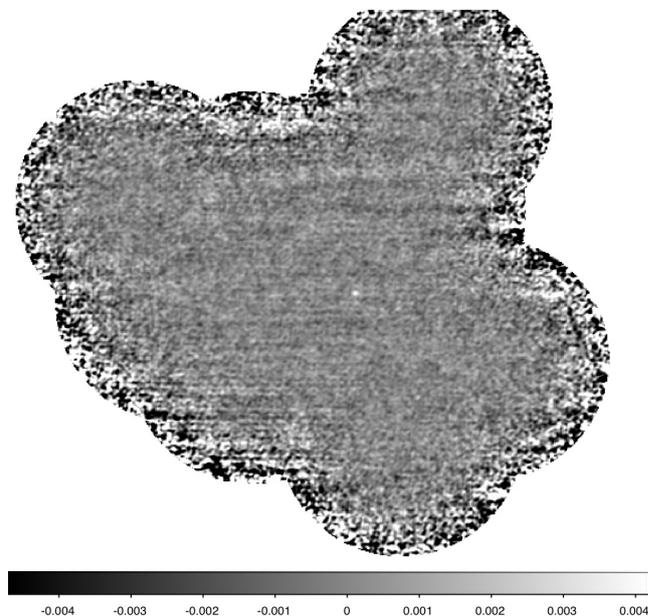
HI Parkes All-Sky Survey (HIPASS) 21-cm data cube showing nearby galaxies (dark spots) and the Galactic Plane (dark sheet). (Meyer et al. 2004). Understanding the noise properties is particularly important for finding the faintest sources.



VLA 20 cm image in the Galactic Plane. This  $10' \times 10'$  is selected to illustrate the non-Gaussian backgrounds. (Becker et al. 2007)



VLA subcube showing four adjacent channels of a large datacube of molecular maser emission in a Galactic star forming region. This shows the spatially correlated, non-Gaussian noise and faint sources common in radio datacubes. (Courtesy NRAO)



A channel of a HI mosaic from the VLA illustrating spatially heteroscedastic noise from primary beam and ripples from mild radio frequency interference. (Courtesy J. van Gorkom)

## Noise structure

- **Spatial autocorrelation** from Fourier transform of visibilities
- **Non-Gaussian `tails`** due to incomplete visibility coverage, un-CLEANed sidelobes of bright sources, un-flagged RFI
- **Heteroscedasticity** (i.e. varies across the 3-dimensional image) due to primary beam, sidelobes and RFI

*If we are lucky, these problems will not be severe in ALMA data. But the methodology should be able to treat them.*

## Steps towards ALMA feature detection & characterization

**Datacube construction:** RFI removal from visibilities, Image formation from visibilities, CLEAN sidelobes and construct datacube

### **Feature identification**

- **Local noise characterization** – Heteroscedasticity & non-Gaussianity
- **Signal detection** - Procedures to mark regions in the datacube with likely continuum and/or line emission. Multiscale for both unresolved and extended structures. Most sources are near the noise and detection thresholds should control for false positives. Important development: False Detection Rate (Benjamini & Hochberg 1995).

### **Feature characterization**

- **Source consolidation** - Procedures to merge adjacent marked subregions into distinct continuum sources and line structures must be adopted. `Image segmentation` from computer vision technology.
- **Source characterization** - Source outline, line center and width, total flux, normal mixture model (Gaussian components), spectral indices, etc.

## Other methods used in radio astronomy

**Visual examination:** Commonly used and praised for sensitivity. But not practical for Tby archives

**ClumpFind, Picasso, MultiFind:** User-specified global S/N threshold after image cleaning (e.g. polynomial fits to continuum; Cornwell et al. 1992, Minchin 1999). ClumpFind then applies: local median/noise threshold, reject peaks smaller than beam, reject peaks near edge (Williams et al. 1994; Di Francesco et al. 2008). MultiFind uses S/N threshold with noise is the robust median absolute deviation measured after Hanning smoothing (Meyer et al. 2004).

**TopHat:** 3-step procedure that scans through frequency planes, removes continuum ripples with median filter, cross-correlates image with multiresolution tophat filter weighted by local noise, groups features in adjacent velocity planes (Meyer et al. 2004).

**Multiresolution wavelet decomposition** (Motte et al. 1998, Knudsen et al. 2006).

**Matched filter algorithms:** CLEAN-type algorithms plus peak-finding (Enoch et al. 2006; Young et al. 2006). Also developed for ALFALFA (Sointong 2007) where it cross-correlates a Fourier transform of datacube with Hermite polynomials. Effective in locating low-surface brightness galaxies, robust to baseline fluctuations, performs all scales in a single calculation.

## Methods from other fields

**Wavelet transform:** Effective for multiscale image decomposition and denoising on both small- & large-scale (low- & high-pass filtering). Strong mathematical foundation (Mallat 1999; Starck & Murtagh 2006).

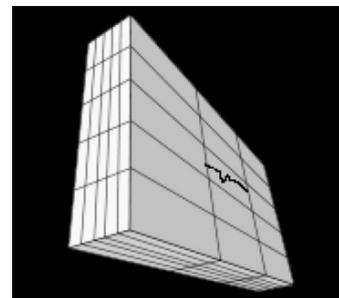
**Local regression:** Long-established techniques for smoothing nonlinear data (Cleveland 1988) and mapping geological strata (kriging, Journel & Huijbregts 1978). Procedures estimate value and noise (variogram) around each location with local windows. Can use robust estimation. Effective for spatial background gradients and heteroscedasticity.

**Semiparametric regression:** New techniques extending nonparametric density estimation (e.g. smoothing, spline fits) to give data-dependent confidence intervals around the estimator (Ruppert et al. 2003; Wasserman 2006). Particularly effective for spatial heteroscedasticity. Not computationally efficient.

**Gamma Test:** First Hanning smooth data with user-specified bandwidth. Second, pass moving window of the  $q$ -th nearest neighbors for specified range of  $q$ . Local background is then estimated for each location by linear regression of noise vs.  $q$ . Signal is then plotted over this background. Developed by Stefansson et al. (1997), Jones et al. (2006, 2007). Applied to 1-dim radio astronomical spectra by Boyce (2003).

## Overview of the suggested procedure

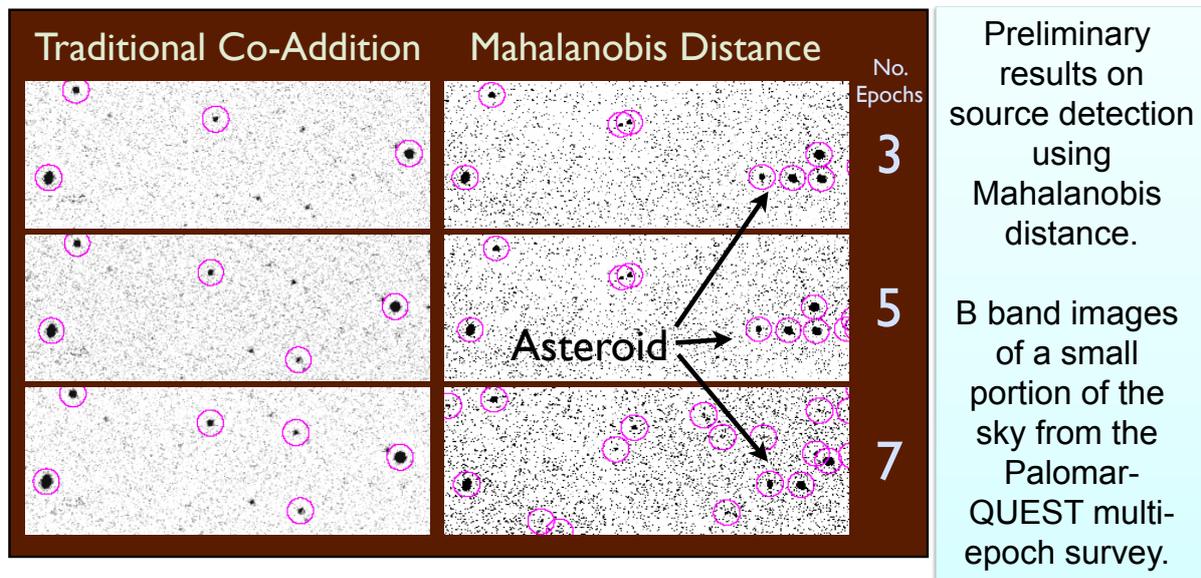
- Robust rejection of bad planes (unflagged RFI)
- Divide the data cube into small overlapping cubes
- Fully 3-dim signal detection in locally homoscedastic subcubes using non-parametric regression
- Pixel-based identification of continuum/ constant sources (rods in 3-dim) and line/transient sources (spots in 3-dim) using thresholds based on the local variance structure and autocorrelation between spectral/temporal planes
- False Discovery Rate control for false positives
- Image segmentation to unify adjacent hits
- Pyramid aggregation or 'Active contour' of resulting 3-dim structures, including non-convex topologies



Local 3-dim subcube  
for source detection

### Statistical approach focuses on:

- The noise distribution in 2D images or 3D subcubes, that exhibit locally homoscedastic noise.
- Local regression methods, if large-scale correlations are present.
- Pixel-level signal detection based on Mahalanobis-type distance  $[D_M(\mathbf{x}, \boldsymbol{\mu}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{S}^{-1} (\mathbf{x} - \boldsymbol{\mu})}]$
- To address spatial correlation from the PSF and from extended sources, the joint quantiles in adjacent channels within a subcube should be examined to set local thresholds.



The three panels show co-adds from 3 (top), 5 (middle) and 7 (bottom) epochs. The left images are formed by a traditional pixel-by-pixel averaging method with  $4\sigma$  sources circled. The images on the right are formed by the statistic based on Mahalanobis distance with significant sources circled. Mahalanobis distance techniques can provide a richer view near the detection threshold.

The emergence of new objects in the right-hand panels is likely due to the passage of a minor planetary body (an asteroid) through the field.

## Analysis

- The analysis proceeds by calculating flux averages in each subcube to treat the heteroscedastic errors.
- The data vectors will be compared with averages over the entire image to set thresholds based on quantiles to identify subcubes with potential source contributions. Noise-only subcubes are discarded.
- Thresholds for multi-pixel triggers will be set based on a Mahalanobis-type statistics.

- Operationally, overlapping windows are passed over each homoscedastic subcube.

- The additive model

$$f_i^r = E(f_i^r) + (f_i^r - E(f_i^r)) = M_i^r + \varepsilon_i^r, \quad i=1, \dots, s,$$

is considered for the data at location  $r$  and channel  $i$ , where

$$\text{Var}(f_i^r) = \text{Var}(\varepsilon_i^r)$$

may vary over the spatial domain. The parameter  $M_i^r$  may be viewed as the true signal of the source at location  $r$  and channel  $i$ .

As all subcubes have an autocorrelation structure due to the synthesized beam:

- We first consider the case where the noise distribution is heteroscedastic (varies with location), but roughly normal (Gaussian).
- In this case the distribution of  $(f_1^r, \dots, f_s^r)$  is multivariate normal and generalized  $\chi^2$ -type statistics can be constructed to test hypotheses of signal at different locations.
- If an autocorrelation structure does not hold and/or Gaussianity fails to hold, **block bootstrap methods** to estimate covariance structure may help in the development of corresponding test statistics.

- In some cases, the multi-epoch correlations of the noise of a datacube may follow an  $AR(1)$  model
- That is, a signal at a location  $i$  is correlated with its neighbors one epoch apart according to
 
$$f_1^r = \varepsilon_1^r, \quad f_i^r = \mu_i^r + \lambda f_{i-1}^r + \varepsilon_i^r,$$
 where  $\varepsilon_i^r$  is zero mean noise.
- The hypothesis  $H_0: (\mu_1^r, \dots, \mu_s^r) = (0, \dots, 0)$  may be tested using a quadratic form  $(f_1^r, \dots, f_s^r) S^{-1} (f_1^r, \dots, f_s^r)^T$ , where the entries of the matrix  $S$  are polynomials of estimate of  $\lambda$ .
- The autocorrelated structure at longer lags can capture much of the deviations from normality in the noise.
- Improved False Discovery Rates (FDR) procedures help to control false positive detections for autocorrelated data.
- Popular procedures used to control the FDR in large-scale multiple testing are stepwise procedures where the p-values are ordered and compared with specified cutoffs according to a stopping rule.
- Starting with the most significant p-value, the step-down procedure rejects each null hypothesis as long as the corresponding cutoff is not exceeded.
- The step-up procedure starts with the least significant p-value and proceeds in the opposite direction and accepts each null hypothesis, provided the p-value does not exceed its cutoff. These procedures have been shown to control the FDR under certain types of dependencies.

We recently started investigating a generalization of these stepwise procedures that allows it to continue rejecting as long as the fraction of p-values not exceeding their cutoffs is sufficiently large.

Preliminary studies including large-sample results indicate that, for appropriate choices of this fractional bound, increased statistical power may be obtained.

A modified FDR procedure that controls for  $P(V/R > c) < \alpha$ , where  $c$  and  $\alpha$  are user-defined is more appropriate. We are investigating such a procedure.

## **CONCLUSION**

Analysis of astronomical datacubes encounter difficulties due to:

- non-Gaussianity and heteroscedasticity of the noise
- evaluation of FDR in the presence of spatial autocorrelation
- need for computational efficiency.

A variety of approaches can be explored involving (semi-)parametric techniques and enhanced FDR calculations.

- The methods described here are designed for, and will be useful for faint source detection in datacubes from new EVLA and ALMA radio/millimeter-band telescopes.
- Techniques developed for the faint line sources in multi-channel radio band surveys may also be directly applicable for detection of faint transients in multi-epoch visible band surveys.



# Chapter 13

## BAYESIAN ANALYSIS

*Notes by Mohan Delampady*

## 1 What is Statistical Inference?

It is an *inverse problem* as in ‘Toy Example’:

**Example 1 (Toy).** Suppose a million candidate stars are examined for the presence of planetary systems associated with them. If 272 ‘successes’ are noticed, how likely that the success rate is 1%, 0.1%, 0.01%,  $\dots$  for the entire universe?

Probability models for observed data involve *direct probabilities*:

**Example 2.** An astronomical study involved 100 galaxies of which 20 are Seyfert galaxies and the rest are starburst galaxies. To illustrate generalization of certain conclusions, say 10 of these 100 galaxies are randomly drawn. How many galaxies drawn will be Seyfert galaxies?

This is exactly like an artificial problem involving an urn having 100 marbles of which 20 are red and the rest blue. 10 marbles are drawn at random with replacement (repeatedly, one by one, after replacing the one previously drawn and mixing the marbles well). How many marbles drawn will be red?

### Data and Models

$X$  = number of Seyfert galaxies (red marbles) in the sample (out of sample size  $n = 10$ )

$$P(X = k|\theta) = \binom{n}{k} \theta^k (1 - \theta)^{(n-k)}, \quad k = 0, 1, \dots, n \quad (1)$$

In (1)  $\theta$  is the proportion of Seyfert galaxies (red marbles) in the urn, which is also the probability of drawing a Seyfert galaxy at each draw. In Example 2,  $\theta = \frac{20}{100} = 0.2$  and  $n = 10$ . So,

$$P(X = 0|\theta = 0.2) = 0.8^{10}, \quad P(X = 1|\theta = 0.2) = 10 \times 0.2 \times 0.8^9, \text{ and so on.}$$

In practice, as in ‘Toy Example’,  $\theta$  is unknown and inference about it is the question to solve.

In the Seyfert/starburst galaxy example, if  $\theta$  is not known and 3 galaxies out of 10 turned out to be Seyfert, one could ask:

how likely is  $\theta = 0.1$ , or 0.2 or 0.3 or  $\dots$ ?

Thus inference about  $\theta$  is an inverse problem:

Causes (parameters)  $\leftarrow$  Effects (observations)

How does this *inversion* work?

The direct probability model  $P(X = k|\theta)$  provides a *likelihood function* for the unknown *parameter*  $\theta$  when data  $X = x$  is observed:

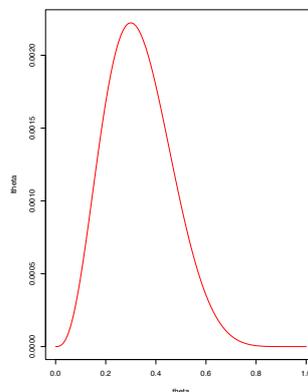
$l(\theta|x) = f(x|\theta)$  ( $= P(X = x|\theta)$  when  $X$  is a discrete random variable) as function of  $\theta$  for given  $x$ .

**Interpretation:**  $f(x|\theta)$  says how likely  $x$  is under different  $\theta$  or the model  $P(\cdot|\theta)$ , so if  $x$  is observed, then  $P(X = x|\theta) = f(x|\theta) = l(\theta|x)$  should be able to indicate what the likelihood of different  $\theta$  values or  $P(\cdot|\theta)$  are for that  $x$ .

As a function of  $x$  for fixed  $\theta$   $P(X = x|\theta)$  is a probability mass function or density, but as a function of  $\theta$  for fixed  $x$ , it has no such meaning, but just a measure of likelihood.

After an experiment is conducted and seeing data  $x$ , the only entity available to convey the information about  $\theta$  obtained from the experiment is  $l(\theta|x)$ .

For the Urn Example we have  $l(\theta|X = 3) \propto \theta^3(1 - \theta)^7$ :



**Maximum Likelihood Estimation (MLE):** If  $l(\theta|x)$  measures the likelihood of different  $\theta$  (or the corresponding models  $P(\cdot|\theta)$ ), just find that  $\theta = \hat{\theta}$  which maximizes the likelihood.

For model (1)

$$\hat{\theta} = \hat{\theta}(x) = x/n = \text{sample proportion of successes .}$$

This is only an estimate. How good is it? What is the possible error in estimation?

Likelihood function  $l(\theta|x)$  has nothing to say about these.

## 2 Frequentist Statistics

Consider repeating this experiment again and again. Then one can look at all possible sample data. i.e. all possible  $x$  values. Utilize *long-run average behaviour* of the MLE. i.e. treat  $\hat{\theta}$  as a random quantity by replacing  $x$  by  $X$  in  $\hat{\theta}(x)$ . i.e. look at  $X/n$  where  $X$  can take all possible values,  $0, 1, \dots, n$ .

$X \sim \text{Binomial}(n, \theta)$  with the probability model (1). Noting that the variance of such an  $X$  is  $n\theta(1 - \theta)$ , one obtains the variance of  $X/n$  to be  $\theta(1 - \theta)/n$ , which can be estimated by  $\hat{\theta}(1 - \hat{\theta})/n$ . A measure of estimation error of  $\hat{\theta}$  is the estimated standard deviation of  $X/n$ , namely,  $\sqrt{\hat{\theta}(1 - \hat{\theta})/n}$ . For further development we need large  $n$ , so that we can apply the *Law of Large Numbers* and the *Central Limit Theorem* to  $X/n$ . Then, the estimator will be close to the true  $\theta$  probabilistically and also, it is approximately distributed like a Gaussian random variable with mean  $\theta$  and variance  $\theta(1 - \theta)/n$ .

### Confidence Statements

Specifically, for large  $n$ , approximately

$$\frac{\hat{\theta} - \theta}{\sqrt{\theta(1 - \theta)/n}} \sim N(0, 1),$$

or

$$\frac{\hat{\theta} - \theta}{\sqrt{\hat{\theta}(1 - \hat{\theta})/n}} \sim N(0, 1). \quad (2)$$

From (2), an approximate 95% confidence interval for  $\theta$  (when  $n$  is large) is

$$\hat{\theta} \pm 2\sqrt{\hat{\theta}(1 - \hat{\theta})/n}.$$

### What Does This Mean?

Simply, if we sample again and again, in about 19 cases out of 20 this random interval

$$\left( \hat{\theta}(X) - 2\sqrt{\hat{\theta}(X)(1 - \hat{\theta}(X))/n}, \hat{\theta}(X) + 2\sqrt{\hat{\theta}(X)(1 - \hat{\theta}(X))/n} \right)$$

will contain the true unknown value of  $\theta$ .

Fine, but what can we say about the one interval that we can construct for the given sample or data  $x$ ?

Nothing; either  $\theta$  is inside  $(0.3 - 2\sqrt{0.3 \times 0.7/10}, 0.3 + 2\sqrt{0.3 \times 0.7/10})$  or it is outside.

Can we say  $0.3 - 2\sqrt{0.3 \times 0.7/10} \leq \theta \leq 0.3 + 2\sqrt{0.3 \times 0.7/10}$  with 95% chance?

Not in this approach. If  $\theta$  is treated as fixed unknown constant, conditioning on the given data  $X = x$  is meaningless.

## 3 Conditioning on Data

- What other approach is possible, then?
- How does one condition on data?
- How does one talk about probability of a model or a hypothesis?

**Example 3.**(not from physics but medicine) Consider a blood test for a certain disease; result is *positive* ( $x = 1$ ) or *negative* ( $x = 0$ ). Suppose  $\theta_1$  denotes *disease is present*,  $\theta_2$  *disease not present*.

Test is not confirmatory. Instead the probability distribution of  $X$  for different  $\theta$  is:

	$x = 0$	$x = 1$	What does it say?
$\theta_1$	0.2	0.8	Test is +ve 80% of time if 'disease present'
$\theta_2$	0.7	0.3	Test is -ve 70% of time if 'disease not present'

If for a particular patient the test result comes out to be 'positive', what should the doctor conclude?

### What is the Question?

What is to be answered is ‘what are the chances that the *disease is present* given that the test is positive?’ i.e.,  $P(\theta = \theta_1 | X = 1)$ .

What we have is  $P(X = 1 | \theta = \theta_1)$  and  $P(X = 1 | \theta = \theta_2)$ .

We have the ‘wrong’ conditional probabilities. They need to be ‘reversed’. But how?

## 4 The Bayesian Recipe

Recall Bayes Theorem: If  $A$  and  $B$  are two events,

$$P(A|B) = \frac{P(A \text{ and } B)}{P(B)}$$

assuming  $P(B) > 0$ . Therefore,  $P(A \text{ and } B) = P(A|B)P(B)$ , and by symmetry  $P(A \text{ and } B) = P(B|A)P(A)$ . Consequently, if  $P(B|A)$  is given and  $P(A|B)$  is desired, note

$$P(A|B) = \frac{P(A \text{ and } B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}.$$

Rule of total probability says,

$$\begin{aligned} P(B) = P(B \text{ and } \Omega) &= P(B \text{ and } A) + P(B \text{ and } A^c) \\ &= P(B|A)P(A) + P(B|A^c)(1 - P(A)), \text{ so} \end{aligned}$$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|A^c)(1 - P(A))} \quad (3)$$

*Bayes Theorem* allows one to invert a certain conditional probability to get a certain other conditional probability. How does this help us?

In our example we want  $P(\theta = \theta_1 | X = 1)$ . From (3),

$$\begin{aligned} &P(\theta = \theta_1 | X = 1) \\ &= \frac{P(X = 1 | \theta = \theta_1)P(\theta = \theta_1)}{P(X = 1 | \theta_1)P(\theta = \theta_1) + P(X = 1 | \theta_2)P(\theta = \theta_2)}. \end{aligned} \quad (4)$$

So, all we need is  $P(\theta = \theta_1)$ , which is simply the probability that a randomly chosen person has this disease, or just the ‘prevalence’ of this disease in the concerned population. The good doctor most likely has this information from his experience in the field. But this is not part of the experimental data. This is pre-experimental information or *prior* information. If we have this, and are willing to incorporate it in the analysis, we get the post-experimental information or *posterior* information in the form of  $P(\theta|X = x)$ .

In our example, if we take  $P(\theta = \theta_1) = 0.05$  or 5%, we get

$$P(\theta = \theta_1 | X = 1) = \frac{0.8 \times 0.05}{0.8 \times 0.05 + 0.3 \times 0.95} = \frac{0.04}{0.325} = 0.123$$

which is only 12.3% and  $P(\theta = \theta_2 | X = 1) = 0.877$  or 87.7%.

Formula (4) which shows how to ‘invert’ the given conditional probabilities,  $P(X = x | \theta)$  into the conditional probabilities of interest,  $P(\theta | X = x)$  is an instance of the **Bayes Theorem**, and hence the *Theory of Inverse Probability* (usage at the time of Bayes and Laplace, late eighteenth century and even by Jeffreys), is known these days as *Bayesian inference*.

Ingredients of Bayesian inference:

**likelihood function**,  $l(\theta|x)$ ;  $\theta$  can be a parameter vector

**prior probability**,  $\pi(\theta)$

Combining the two, one gets the **posterior probability** density or mass function

$$\pi(\theta | x) = \begin{cases} \frac{\pi(\theta)l(\theta|x)}{\sum_j \pi(\theta_j)l(\theta_j|x)} & \text{if } \theta \text{ is discrete;} \\ \frac{\pi(\theta)l(\theta|x)}{\int \pi(u)l(u|x) du} & \text{if } \theta \text{ is continuous.} \end{cases} \quad (5)$$

## 5 Inference for Binomial proportion

**Example 2 contd.** Suppose we have no special information available on  $\theta$ . Then assume  $\theta$  is uniformly distributed on the interval  $(0, 1)$ . i.e., the prior density is  $\pi(\theta) = 1$ ,  $0 < \theta < 1$ .

This is a choice of *non-informative* or *vague* or *reference* prior. Often, Bayesian inference from such a prior coincides with classical inference.

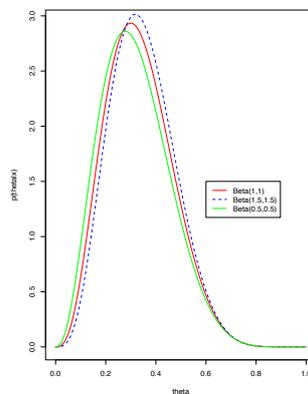
In the Example then the **posterior density of  $\theta$  given  $x$  is**

$$\begin{aligned}\pi(\theta|x) &= \frac{\pi(\theta)l(\theta|x)}{\int \pi(u)l(u|x) du} \\ &= \frac{(n+1)!}{x!(n-x)!} \theta^x (1-\theta)^{n-x}, \quad 0 < \theta < 1.\end{aligned}$$

As a function of  $\theta$ , this is the same as the likelihood function  $l(\theta|x) \propto \theta^x (1-\theta)^{n-x}$ , and so maximizing the posterior probability density will give the same estimate as the maximum likelihood estimate!

### Influence of the Prior

If we had some knowledge about  $\theta$  which can be summarized in the form of a Beta prior distribution with parameters  $\alpha$  and  $\gamma$ , the posterior will also be Beta with parameters  $x + \alpha$  and  $n - x + \gamma$ . Such priors which result in posteriors from the same ‘family’ are called ‘natural conjugate priors’. Robustness?



### Objective Bayesian Analysis:

Invariant priors: Jeffreys

Reference priors: Bernardo, Jeffreys

Maximum entropy priors: Jaynes

In Example 2, what  $\pi(\theta|x)$  says is that the uncertainty in  $\theta$  can now be described in terms of an actual probability distribution concentrated around the maximum likelihood estimate  $\hat{\theta} = x/n$ . However, the interpretation of  $\hat{\theta}$  as an estimate of  $\theta$  is quite different. It is the most probable value of the unknown parameter  $\theta$  conditional on the sample data  $x$ ; it is called the ‘maximum a posteriori estimate (MAP)’ or the ‘highest posterior density estimate (HPD)’.

There is no need to mimic the MLE anymore. We have a genuine probability distribution, namely, the posterior distribution to quantify our post-experimental knowledge about  $\theta$ . Indeed the usual Bayes estimate is the mean of the posterior distribution which minimizes the posterior dispersion:

$$E[(\theta - \hat{\theta}_B)^2|x] = \min_a E[(\theta - a)^2|x],$$

when  $\hat{\theta}_B = E(\theta|x)$ .

If we choose  $\hat{\theta}_B$  as the estimate of  $\theta$ , we get a natural measure of variability of this estimate in the form of the posterior variance:  $E[(\theta - E(\theta|x))^2|x]$ . Therefore the posterior standard deviation is a natural measure of estimation error. i.e., our estimate is  $\hat{\theta}_B \pm \sqrt{E[(\theta - E(\theta|x))^2|x]}$ .

In fact, we can say much more. For any interval around  $\hat{\theta}$  we can compute the (posterior) probability of it containing the true parameter  $\theta$ . In other words, a statement such as

$$P(\hat{\theta}_B - k_1 \leq \theta \leq \hat{\theta}_B + k_2|x) = 0.95$$

is perfectly meaningful.

All these inferences are conditional on the given data.

In Example 2, if the prior is a Beta distribution with parameters  $\alpha$  and  $\gamma$ , then  $\theta|x$  will have a Beta( $x + \alpha, n - x + \gamma$ ) distribution, so the Bayes estimate of  $\theta$  will be

$$\hat{\theta}_B = \frac{(x + \alpha)}{(n + \alpha + \gamma)} = \frac{n}{n + \alpha + \gamma} \frac{x}{n} + \frac{\alpha + \gamma}{n + \alpha + \gamma} \frac{\alpha}{\alpha + \gamma}.$$

This is a convex combination of sample mean and prior mean, with the weights depending upon the sample size and the strength of the prior information as measured by the values of  $\alpha$  and  $\gamma$ .

Bayesian inference relies on the conditional probability language to revise one's knowledge. In the above example, prior to the collection of sample data one had some (vague, perhaps) information on  $\theta$ . Then came the sample data. Combining the model density of this data with the prior density one gets the posterior density, the conditional density of  $\theta$  given the data. From now on until further data is available, this posterior distribution of  $\theta$  is the only relevant information as far as  $\theta$  is concerned.

## 6 Inference With Normals/Gaussians

**Gaussian PDF:**

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad \text{over } [-\infty, \infty] \quad (6)$$

Common abbreviated notation:  $X \sim N(\mu, \sigma^2)$

**Parameters:**

$$\begin{aligned} \mu &= E(X) \equiv \langle X \rangle \equiv \int x f(x|\mu, \sigma^2) dx \\ \sigma^2 &= E(X - \mu)^2 \equiv \langle (X - \mu)^2 \rangle \equiv \int (x - \mu)^2 f(x|\mu, \sigma^2) dx \end{aligned}$$

### Inference About a Normal Mean

**Example 4.** Fit a normal/Gaussian model to the ‘globular cluster luminosity functions’ data. The set-up is as follows.

Our data consist of  $n$  measurements,  $X_i = \mu + \epsilon_i$ .

Suppose the noise contributions are independent, and  $\epsilon_i \sim N(0, \sigma^2)$ . Denoting by  $\mathbf{x}$ , the random sample  $(x_1, \dots, x_n)$ ,

$$\begin{aligned} f(\mathbf{x}|\mu, \sigma^2) &= \prod_i f(x_i|\mu, \sigma^2) \\ &= \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x_i-\mu)^2} \\ &= (2\pi\sigma^2)^{-n/2} e^{-\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i-\mu)^2} \\ &= (2\pi\sigma^2)^{-n/2} e^{-\frac{1}{2\sigma^2} [\sum_{i=1}^n (x_i-\bar{x})^2 + n(\bar{x}-\mu)^2]}. \end{aligned}$$

Note  $(\bar{X}, s^2 = \sum_{i=1}^n (X_i - \bar{X})^2 / (n-1))$  is *sufficient* for the parameters  $(\mu, \sigma^2)$ . This is a very substantial data compression.

### Inference About a Normal Mean, $\sigma^2$ known

(Not useful, but easy to understand.)

$$l(\mu|\mathbf{x}) \propto f(\mathbf{x}|\mu, \sigma^2) \propto e^{-\frac{n}{2\sigma^2}(\mu - \bar{x})^2},$$

so that  $\bar{X}$  is sufficient. Also,  $\bar{X}|\mu \sim N(\mu, \sigma^2/n)$ . If an informative prior,  $\mu \sim N(\mu_0, \tau^2)$  is chosen for  $\mu$ ,

$$\begin{aligned} \pi(\mu|\mathbf{x}) &\propto l(\mu|\mathbf{x})\pi(\mu) \\ &\propto e^{-\frac{1}{2}\left[\frac{n(\mu-\bar{x})^2}{\sigma^2} + \frac{(\mu-\mu_0)^2}{\tau^2}\right]} \\ &\propto e^{-\frac{\tau^2 + \sigma^2/n}{2\tau^2\sigma^2/n}\left(\mu - \frac{\tau^2\sigma^2/n}{\tau^2 + \sigma^2/n}\left(\frac{\mu_0}{\tau^2} + \frac{n\bar{x}}{\sigma^2}\right)\right)^2}. \end{aligned}$$

i.e.,  $\mu|\mathbf{x} \sim N(\hat{\mu}, \delta^2)$ :

$$\begin{aligned} \hat{\mu} &= \frac{\tau^2\sigma^2/n}{\tau^2 + \sigma^2/n}\left(\frac{\mu_0}{\tau^2} + \frac{n\bar{x}}{\sigma^2}\right) \\ &= \frac{\tau^2}{\tau^2 + \sigma^2/n}\bar{x} + \frac{\sigma^2/n}{\tau^2 + \sigma^2/n}\mu_0. \end{aligned}$$

$\hat{\mu}$  is the Bayes estimate of  $\mu$ , which is just a weighted average of sample mean  $\bar{x}$  and prior mean  $\mu_0$ .

$\delta^2$  is the posterior variance of  $\mu$  and

$$\delta^2 = \frac{\tau^2\sigma^2/n}{\tau^2 + \sigma^2/n} = \frac{\sigma^2}{n} \frac{\tau^2}{\tau^2 + \sigma^2/n}.$$

Therefore  $\hat{\mu} \pm \delta$  is our estimate for  $\mu$  and  $\hat{\mu} \pm 2\delta$  is a 95% HPD (Bayesian) credible interval for  $\mu$ .

What happens as  $\tau^2 \rightarrow \infty$ , or as the prior becomes more and more flat?

$$\hat{\mu} \rightarrow \bar{x}, \quad \delta \rightarrow \frac{\sigma}{\sqrt{n}}$$

i.e., Jeffreys' prior  $\pi(\mu) = C$  reproduces frequentist inference.

### Inference About a Normal Mean, $\sigma^2$ unknown

Our observations  $X_1, \dots, X_n$  is a random sample from a Gaussian population with both mean  $\mu$  and variance  $\sigma^2$  unknown.

We are only interested in  $\mu$ .

How do we get rid of the nuisance parameter  $\sigma^2$ ?

Bayesian inference uses posterior distribution which is a probability distribution, so  $\sigma^2$  should be integrated out from the joint posterior distribution of  $\mu$  and  $\sigma^2$ .

$$l(\mu, \sigma^2 | \mathbf{x}) = (2\pi\sigma^2)^{-n/2} e^{-\frac{1}{2\sigma^2} [\sum_{i=1}^n (x_i - \bar{x})^2 + n(\mu - \bar{x})^2]}.$$

Start with  $\pi(\mu, \sigma^2)$  and get

$$\pi(\mu, \sigma^2 | \mathbf{x}) \propto \pi(\mu, \sigma^2) l(\mu, \sigma^2 | \mathbf{x})$$

and then get

$$\pi(\mu | \mathbf{x}) = \int_0^\infty \pi(\mu, \sigma^2 | \mathbf{x}) d\sigma^2.$$

Use Jeffreys' prior  $\pi(\mu, \sigma^2) \propto 1/\sigma^2$ : Flat prior for  $\mu$  which is a location or translation parameter, and an independent flat prior for  $\log(\sigma)$  which is again a location parameter, being the log of a scale parameter.

$$\pi(\mu, \sigma^2 | \mathbf{x}) \propto \frac{1}{\sigma^2} l(\mu, \sigma^2 | \mathbf{x})$$

$$\begin{aligned} \pi(\mu | \mathbf{x}) &\propto \int_0^\infty (\sigma^2)^{-(n+1)/2} e^{-\frac{1}{2\sigma^2} [\sum_{i=1}^n (x_i - \bar{x})^2 + n(\mu - \bar{x})^2]} d\sigma^2 \\ &\propto [(n-1)s^2 + n(\mu - \bar{x})^2]^{-n/2} \\ &\propto \left[ 1 + \frac{1}{n-1} \frac{n(\mu - \bar{x})^2}{s^2} \right]^{-n/2} \\ &\propto \text{density of Students } t_{n-1}. \end{aligned}$$

$$\frac{\sqrt{n}(\mu - \bar{x})}{s} \mid \text{data} \sim t_{n-1}$$

$$P\left(\bar{x} - t_{n-1}(0.975) \frac{s}{\sqrt{n}} \leq \mu \leq \bar{x} + t_{n-1}(0.975) \frac{s}{\sqrt{n}} \mid \text{data}\right) = 95\%$$

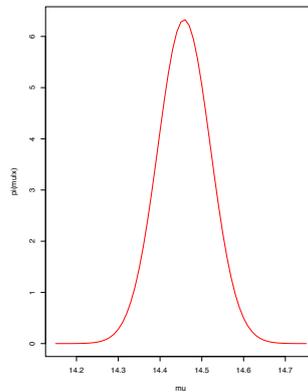
i.e., the **Jeffreys' translation-scale invariant prior** reproduces frequentist inference.

What if there are some constraints on  $\mu$  such as  $-A \leq \mu \leq B$ , for example,  $\mu > 0$ ? We will get a truncated  $t_{n-1}$  instead, but the procedure will go through with minimal change.

**Example 4 contd.** (GCL Data)  $n = 360$ ,  $\bar{x} = 14.46$ ,  $s = 1.19$ .

$$\frac{\sqrt{360}(\mu - 14.46)}{1.19} \mid \text{data} \sim t_{359}$$

$\mu \mid \text{data} \sim N(14.46, 0.063^2)$  approximately.



Estimate for mean GCL is  $14.46 \pm 0.063$  and 95% HPD credible interval is (14.33, 14.59).

### Comparing two Normal Means

**Example 5.** Check whether the mean distance indicators in the two populations of LMC datasets are different. Model as follows:

$X_1, \dots, X_{n_1}$  is a random sample from  $N(\mu_1, \sigma_1^2)$ .

$Y_1, \dots, Y_{n_2}$  is a random sample from  $N(\mu_2, \sigma_2^2)$ .

Samples are independent.

Unknown parameters:  $(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$

Quantity of interest:  $\eta = \mu_1 - \mu_2$

Nuisance parameters:  $\sigma_1^2$  and  $\sigma_2^2$

**Case 1.**  $\sigma_1^2 = \sigma_2^2$ . Then sufficient statistic for  $(\mu_1, \mu_2, \sigma^2)$  is  $(\bar{X}, \bar{Y}, s^2 = \frac{1}{n_1+n_2-2} (\sum_{i=1}^{n_1} (X_i - \bar{X})^2 + \sum_{j=1}^{n_2} (Y_j - \bar{Y})^2))$

$\bar{X}|\mu_1, \mu_2, \sigma^2 \sim N(\mu_1, \sigma^2/n_1)$ ,  $\bar{Y}|\mu_1, \mu_2, \sigma^2 \sim N(\mu_2, \sigma^2/n_2)$ ,  $(n_1+n_2-2)s^2|\mu_1, \mu_2, \sigma^2 \sim \sigma^2 \chi_{n_1+n_2-2}^2$ .

These three are independently distributed.

$\bar{X} - \bar{Y}|\mu_1, \mu_2, \sigma^2 \sim N(\eta, \sigma^2(\frac{1}{n_1} + \frac{1}{n_2}))$ ,  $\eta = \mu_1 - \mu_2$

Use Jeffreys' location-scale invariant prior  $\pi(\mu_1, \mu_2, \sigma^2) \propto 1/\sigma^2$

$\eta|\sigma^2, \mathbf{x}, \mathbf{y} \sim N(\bar{x} - \bar{y}, \sigma^2(\frac{1}{n_1} + \frac{1}{n_2}))$ , and

$$\pi(\eta, \sigma^2|\mathbf{x}, \mathbf{y}) \propto \pi(\eta|\sigma^2, \mathbf{x}, \mathbf{y})\pi(\sigma^2|s^2), \quad (7)$$

Integrate out  $\sigma^2$  from (7) as in the previous example to get

$$\frac{\eta - (\bar{x} - \bar{y})}{s\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} | \mathbf{x}, \mathbf{y} \sim t_{n_1+n_2-2}.$$

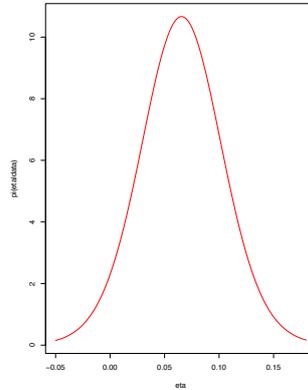
95% HPD credible interval for  $\eta = \mu_1 - \mu_2$  is

$$\bar{x} - \bar{y} \pm t_{n_1+n_2-2}(0.975)s\sqrt{\frac{1}{n_1} + \frac{1}{n_2}},$$

same as frequentist *t*-interval.

**Example 5 contd.** We have  $\bar{x} = 18.539$ ,  $\bar{y} = 18.473$ ,  $n_1 = 13$ ,  $n_2 = 12$  and  $s^2 = 0.0085$ .  $\hat{\eta} = \bar{x} - \bar{y} = 0.066$ ,  $s\sqrt{\frac{1}{n_1} + \frac{1}{n_2}} = 0.037$ ,  $t_{23}(0.975) = 2.069$ .

95% HPD credible interval for  $\eta = \mu_1 - \mu_2$ :  $(0.066 - 2.069 \times 0.037, 0.066 + 2.069 \times 0.037) = (-0.011, 0.142)$ .



**Case 2.**  $\sigma_1^2$  and  $\sigma_2^2$  are not known to be equal.

From the one-sample normal example, note that  $(\bar{X}, s_X^2 = \frac{1}{n_1-1} \sum_{i=1}^{n_1} (X_i - \bar{X})^2)$  sufficient for  $(\mu_1, \sigma_1^2)$ , and  $(\bar{Y}, s_Y^2 = \frac{1}{n_2-1} \sum_{j=1}^{n_2} (Y_j - \bar{Y})^2)$  sufficient for  $(\mu_2, \sigma_2^2)$ .

Making inference on  $\eta = \mu_1 - \mu_2$  when  $\sigma_1^2$  and  $\sigma_2^2$  are not assumed to be equal is called the Behrens-Fisher problem for which the frequentist solution is not very straight forward, but the Bayes solution is.

$\bar{X} | \mu_1, \sigma_1^2 \sim N(\mu_1, \sigma_1^2/n_1)$ ,  $(n_1 - 1)s_X^2 | \mu_1, \sigma_1^2 \sim \sigma_1^2 \chi_{n_1-1}^2$ , and are independently distributed.

$\bar{Y} | \mu_2, \sigma_2^2 \sim N(\mu_2, \sigma_2^2/n_2)$ ,  $(n_2 - 1)s_Y^2 | \mu_2, \sigma_2^2 \sim \sigma_2^2 \chi_{n_2-1}^2$ , and are independently distributed.

$\mathbf{X}$  and  $\mathbf{Y}$  samples are independent.

Use Jeffreys' prior  $\pi(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2) \propto 1/\sigma_1^2 \times 1/\sigma_2^2$

Calculations similar to those in one-sample case give:

$$\begin{aligned} \frac{\sqrt{n_1}(\mu_1 - \bar{x})}{s_X} | \text{data} &\sim t_{n_1-1}, \\ \frac{\sqrt{n_2}(\mu_2 - \bar{y})}{s_Y} | \text{data} &\sim t_{n_2-1}, \end{aligned} \quad (8)$$

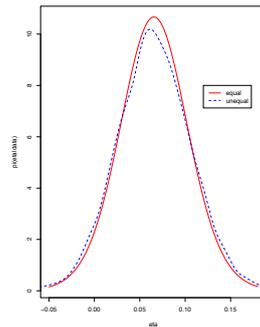
and these two are independent.

Posterior distribution of  $\eta = \mu_1 - \mu_2$  given the data is non-standard (differ-

ence of two independent  $t$  variables) but not difficult to get.

**Use Monte-Carlo Sampling:** Simply generate  $(\mu_1, \mu_2)$  repeatedly from (8) and construct a histogram for  $\eta = \mu_1 - \mu_2$

**Example 5 (LMC) contd.** Looks slightly different.



Posterior mean of  $\eta = \mu_1 - \mu_2$  is

$$\hat{\eta} = E(\mu_1 - \mu_2 | \text{data}) = \begin{cases} 0.0656 & \text{equal variance;} \\ 0.0657 & \text{unequal variance.} \end{cases} \quad (9)$$

95% HPD credible interval for  $\eta = \mu_1 - \mu_2$  is

$$= \begin{cases} (-0.011, 0.142) & \text{equal variance;} \\ (-0.014, 0.147) & \text{unequal variance.} \end{cases} \quad (10)$$

## 7 Bayesian Computations

Bayesian analysis requires computation of expectations and quantiles of probability distributions (posterior distributions). Most often posterior distributions will not be standard distributions. Then posterior quantities of inferential interest cannot be computed in closed form. Special techniques are needed.

**Example M1.** Suppose  $X_1, X_2, \dots, X_k$  are observed number of certain type of stars in  $k$  similar regions. Model them as independent Poisson counts:

$X_i \sim \text{Poisson}(\theta_i)$ .  $\theta_i$  are *a priori* considered related.  $\nu_i = \log(\theta_i)$  is the  $i$ th element of  $\boldsymbol{\nu}$  and suppose

$$\boldsymbol{\nu} \sim N_k(\mu\mathbf{1}, \tau^2 \{(1-\rho)I_k + \rho\mathbf{1}\mathbf{1}'\}),$$

where  $\mathbf{1}$  is the  $k$ -vector with all elements being 1, and  $\mu$ ,  $\tau^2$  and  $\rho$  are known constants. Then

$$f(\mathbf{x}|\boldsymbol{\nu}) = \exp\left(-\sum_{i=1}^k \{e^{\nu_i} - \nu_i x_i\}\right) / \prod_{i=1}^k x_i!$$

$$\pi(\boldsymbol{\nu}) \propto \exp\left(-\frac{1}{2\tau^2}(\boldsymbol{\nu} - \mu\mathbf{1})'((1-\rho)I_k + \rho\mathbf{1}\mathbf{1}')^{-1}(\boldsymbol{\nu} - \mu\mathbf{1})\right)$$

$$\pi(\boldsymbol{\nu}|\mathbf{x}) \propto \exp\left\{-\sum_{i=1}^k \{e^{\nu_i} - \nu_i x_i\} - \frac{(\boldsymbol{\nu} - \mu\mathbf{1})'((1-\rho)I_k + \rho\mathbf{1}\mathbf{1}')^{-1}(\boldsymbol{\nu} - \mu\mathbf{1})}{2\tau^2}\right\}.$$

To obtain the posterior mean of  $\theta_j$ , compute

$$E^\pi(\theta_j|x) = E^\pi(\exp(\nu_j)|x) = \frac{\int_{\mathcal{R}^k} \exp(\nu_j)g(\boldsymbol{\nu}|\mathbf{x}) d\boldsymbol{\nu}}{\int_{\mathcal{R}^k} g(\boldsymbol{\nu}|\mathbf{x}) d\boldsymbol{\nu}},$$

$$\text{where } g(\boldsymbol{\nu}|\mathbf{x}) = \exp\left\{-\sum_{i=1}^k \{e^{\nu_i} - \nu_i x_i\} - \frac{(\boldsymbol{\nu} - \mu\mathbf{1})'((1-\rho)I_k + \rho\mathbf{1}\mathbf{1}')^{-1}(\boldsymbol{\nu} - \mu\mathbf{1})}{2\tau^2}\right\}.$$

This is a ratio of two  $k$ -dimensional integrals, and as  $k$  grows, the integrals become less and less easy to work with. Numerical integration techniques fail to be an efficient technique in this case. This problem, known as the *curse of dimensionality*, is due to the fact that the size of the part of the space that is not relevant for the computation of the integral grows very fast with the dimension. Consequently, the *error in approximation associated with this numerical method increases as the power of the dimension  $k$* , making the technique inefficient.

The recent popularity of Bayesian approach to statistical applications is mainly due to advances in statistical computing. These include the E-M algorithm and the Markov chain Monte Carlo (MCMC) sampling techniques.

## 8 Monte Carlo Sampling

Consider an expectation that is not available in closed form. *To estimate a population mean*, gather a large sample from this population and consider

the corresponding **sample mean**. The **Law of Large Numbers** guarantees that the estimate will be *good* provided the sample is large enough. Specifically, let  $f$  be a probability density function (or a mass function) and suppose the quantity of interest is a finite expectation of the form

$$E_f h(\mathbf{X}) = \int_{\mathcal{X}} h(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \quad (11)$$

(or the corresponding sum in the discrete case). If i.i.d. observations  $\mathbf{X}_1, \mathbf{X}_2, \dots$  can be generated from the density  $f$ , then

$$\bar{h}_m = \frac{1}{m} \sum_{i=1}^m h(\mathbf{X}_i) \quad (12)$$

converges in probability to  $E_f h(\mathbf{X})$ . This justifies using  $\bar{h}_m$  as an approximation for  $E_f h(\mathbf{X})$  for large  $m$ .

To provide a measure of accuracy or the extent of error in the approximation, compute the standard error. If  $\text{Var}_f h(\mathbf{X})$  is finite, then  $\text{Var}_f(\bar{h}_m) = \text{Var}_f h(\mathbf{X})/m$ . Further,  $\text{Var}_f h(\mathbf{X}) = E_f h^2(\mathbf{X}) - (E_f h(\mathbf{X}))^2$  can be estimated by

$$s_m^2 = \frac{1}{m} \sum_{i=1}^m (h(\mathbf{X}_i) - \bar{h}_m)^2,$$

and hence the **standard error of  $\bar{h}_m$**  can be estimated by

$$\frac{1}{\sqrt{m}} s_m = \frac{1}{m} \left( \sum_{i=1}^m (h(\mathbf{X}_i) - \bar{h}_m)^2 \right)^{1/2}.$$

Confidence intervals for  $E_f h(\mathbf{X})$ : Using CLT

$$\frac{\sqrt{m} (\bar{h}_m - E_f h(\mathbf{X}))}{s_m} \xrightarrow{m \rightarrow \infty} N(0, 1), \text{ so}$$

$(\bar{h}_m - z_{\alpha/2} s_m / \sqrt{m}, \bar{h}_m + z_{\alpha/2} s_m / \sqrt{m})$  can be used as an approximate  $100(1 - \alpha)\%$  confidence interval for  $E_f h(\mathbf{X})$ , with  $z_{\alpha/2}$  denoting the  $100(1 - \alpha/2)\%$  quantile of standard normal.

If we want to approximate the posterior mean, try to generate i.i.d. observations from the posterior distribution and consider the mean of this sample. This is rarely useful because most often the posterior distribution will be a non-standard distribution which may not easily allow sampling from it.

**What are some other possibilities?**

**Example M2.** Suppose  $X$  is  $N(\theta, \sigma^2)$  with known  $\sigma^2$  and a Cauchy( $\mu, \tau$ ) prior on  $\theta$  is considered appropriate. Then

$$\pi(\theta|x) \propto \exp(-(\theta-x)^2/(2\sigma^2)) (\tau^2 + (\theta-\mu)^2)^{-1},$$

and hence the posterior mean is

$$\begin{aligned} E^\pi(\theta|x) &= \frac{\int_{-\infty}^{\infty} \theta \exp\left(-\frac{(\theta-x)^2}{2\sigma^2}\right) (\tau^2 + (\theta-\mu)^2)^{-1} d\theta}{\int_{-\infty}^{\infty} \exp\left(-\frac{(\theta-x)^2}{2\sigma^2}\right) (\tau^2 + (\theta-\mu)^2)^{-1} d\theta} \\ &= \frac{\int_{-\infty}^{\infty} \theta \left\{ \frac{1}{\sigma} \phi\left(\frac{\theta-x}{\sigma}\right) \right\} (\tau^2 + (\theta-\mu)^2)^{-1} d\theta}{\int_{-\infty}^{\infty} \left\{ \frac{1}{\sigma} \phi\left(\frac{\theta-x}{\sigma}\right) \right\} (\tau^2 + (\theta-\mu)^2)^{-1} d\theta}, \end{aligned}$$

where  $\phi$  denotes the density of standard normal.

$E^\pi(\theta|x)$  is the ratio of expectation of  $h(\theta) = \theta/(\tau^2 + (\theta-\mu)^2)$  to that of  $h(\theta) = 1/(\tau^2 + (\theta-\mu)^2)$ , both expectations being with respect to the  $N(x, \sigma^2)$  distribution. Therefore, we simply sample  $\theta_1, \theta_2, \dots$  from  $N(x, \sigma^2)$  and use

$$\widehat{E^\pi(\theta|x)} = \frac{\sum_{i=1}^m \theta_i (\tau^2 + (\theta_i - \mu)^2)^{-1}}{\sum_{i=1}^m (\tau^2 + (\theta_i - \mu)^2)^{-1}}$$

as our Monte Carlo estimate of  $E^\pi(\theta|x)$ . Note that (11) and (12) are applied separately to both the numerator and denominator, but using the same sample of  $\theta$ 's. It is unwise to assume that the problem has been completely solved. The sample of  $\theta$ 's generated from  $N(x, \sigma^2)$  will tend to concentrate around  $x$ , whereas to satisfactorily account for the contribution of the Cauchy prior to the posterior mean, a significant portion of the  $\theta$ 's should come from the tails of the posterior distribution.

Why not express the posterior mean in the form

$$E^\pi(\theta|x) = \frac{\int_{-\infty}^{\infty} \theta \exp\left(-\frac{(\theta-x)^2}{2\sigma^2}\right) \pi(\theta) d\theta}{\int_{-\infty}^{\infty} \exp\left(-\frac{(\theta-x)^2}{2\sigma^2}\right) \pi(\theta) d\theta},$$

and then sample  $\theta$ 's from Cauchy( $\mu, \tau$ ) and use the approximation

$$\widehat{E^\pi(\theta|x)} = \frac{\sum_{i=1}^m \theta_i \exp\left(-\frac{(\theta_i-x)^2}{2\sigma^2}\right)}{\sum_{i=1}^m \exp\left(-\frac{(\theta_i-x)^2}{2\sigma^2}\right)}?$$

However, this is also not satisfactory because the tails of the posterior distribution are not as heavy as those of the Cauchy prior, and there will be excess sampling from the tails relative to the center. So the convergence of the approximation will be slower resulting in a larger error in approximation (for a fixed  $m$ ). Ideally, therefore, sampling should be from the posterior distribution itself. With this view in mind, a variation of the above theme, called Monte Carlo importance sampling has been developed.

Consider (11) again. Suppose that it is difficult or expensive to sample directly from  $f$ , but there exists a probability density  $u$  that is very close to  $f$  from which it is easy to sample. Then we can rewrite (11) as

$$\begin{aligned} E_f h(\mathbf{X}) &= \int_{\mathcal{X}} h(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} = \int_{\mathcal{X}} h(\mathbf{x}) \frac{f(\mathbf{x})}{u(\mathbf{x})} u(\mathbf{x}) d\mathbf{x} \\ &= \int_{\mathcal{X}} \{h(\mathbf{x}) w(\mathbf{x})\} u(\mathbf{x}) d\mathbf{x} = E_u \{h(\mathbf{X}) w(\mathbf{X})\}, \end{aligned}$$

where  $w(\mathbf{x}) = f(\mathbf{x})/u(\mathbf{x})$ . Now apply (12) with  $f$  replaced by  $u$  and  $h$  replaced by  $hw$ . In other words, generate i.i.d. observations  $\mathbf{X}_1, \mathbf{X}_2, \dots$  from the density  $u$  and compute

$$\overline{hw}_m = \frac{1}{m} \sum_{i=1}^m h(\mathbf{X}_i) w(\mathbf{X}_i).$$

The sampling density  $u$  is called the *importance* function.<sup>1</sup>

## 9 Markov Chain Monte Carlo Methods

A severe drawback of the standard Monte Carlo sampling/ importance sampling: complete determination of the functional form of the posterior density is needed for implementation.

Situations where posterior distributions are incompletely specified or are specified indirectly cannot be handled: joint posterior distribution of the vector of parameters is specified in terms of several conditional and marginal distributions, but not directly.

This covers a large range of Bayesian analysis because a lot of Bayesian modeling is hierarchical so that the joint posterior is difficult to calculate but

<sup>1</sup>Rest of these notes was not covered in the lectures and may be omitted at first reading.

the conditional posteriors given parameters at different levels of hierarchy are easier to write down (and hence sample from).

**Markov Chains.** A sequence of random variables  $\{X_n\}_{n \geq 0}$  is a *Markov chain* if for any  $n$ , given the current value,  $X_n$ , the *past*  $\{X_j, j \leq n-1\}$  and the *future*  $\{X_j : j \geq n+1\}$  are *independent*. In other words,

$$P(A \cap B | X_n) = P(A | X_n)P(B | X_n), \quad (13)$$

where  $A$  and  $B$  are events defined respectively in terms of the past and the future.

Important subclass: Markov chains with time homogeneous or *stationary transition probabilities*: the probability distribution of  $X_{n+1}$  given  $X_n = x$ , and the past,  $X_j : j \leq n-1$  depends only on  $x$  and does not depend on the values of  $X_j : j \leq n-1$  or  $n$ .

If the set  $S$  of values  $\{X_n\}$  can take, known as the *state space*, is countable, this reduces to specifying the transition probability matrix  $P \equiv ((p_{ij}))$  where for any two values  $i, j$  in  $S$ ,  $p_{ij}$  is the probability that  $X_{n+1} = j$  given  $X_n = i$ , i.e., of moving from state  $i$  to state  $j$  in one time unit.

For state space  $S$  that is not countable, specify a *transition kernel* or *transition function*  $P(x, \cdot)$  where  $P(x, A)$  is the probability of moving from  $x$  into  $A$  in one step, i.e.,  $P(X_{n+1} \in A | X_n = x)$ .

Given the transition probability and the probability distribution of the initial value  $X_0$ , one can construct the joint probability distribution of  $\{X_j : 0 \leq j \leq n\}$  for any finite  $n$ . i.e.,

$$\begin{aligned} P(X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}, X_n = i_n) \\ &= P(X_n = i_n | X_0 = i_0, \dots, X_{n-1} = i_{n-1}) \\ &\quad \times P(X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}) \\ &= p_{i_{n-1}i_n} P(X_0 = i_0, \dots, X_{n-1} = i_{n-1}) \\ &= P(X_0 = i_0) p_{i_0i_1} p_{i_1i_2} \dots p_{i_{n-1}i_n}. \end{aligned}$$

A probability distribution  $\pi$  is called *stationary* or *invariant* for a transition probability  $P$  or the associated Markov chain  $\{X_n\}$  if it is the case that when the probability distribution of  $X_0$  is  $\pi$  then the same is true for  $X_n$  for all  $n \geq 1$ . Thus in the countable state space case a probability distribution  $\pi = \{\pi_i : i \in S\}$  is stationary for a transition probability matrix  $P$  if for

each  $j$  in  $S$ ,

$$\begin{aligned} P(X_1 = j) &= \sum_i P(X_1 = j | X_0 = i) P(X_0 = i) \\ &= \sum_i \pi_i p_{ij} = P(X_0 = j) = \pi_j. \end{aligned} \quad (14)$$

In vector notation it says  $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots)$  is a left eigenvector of the matrix  $P$  with eigenvalue 1 and

$$\boldsymbol{\pi} = \boldsymbol{\pi} P. \quad (15)$$

Similarly, if  $S$  is a continuum, a probability distribution  $\pi$  with density  $p(x)$  is *stationary* for the transition kernel  $P(\cdot, \cdot)$  if

$$\pi(A) = \int_A p(x) dx = \int_S P(x, A) p(x) dx$$

for all  $A \subset S$ .

A Markov chain  $\{X_n\}$  with a countable state space  $S$  and transition probability matrix  $P \equiv ((p_{ij}))$  is said to be *irreducible* if for any two states  $i$  and  $j$  the probability of the Markov chain visiting  $j$  starting from  $i$  is positive, i.e., for some  $n \geq 1$ ,  $p_{ij}^{(n)} \equiv P(X_n = j | X_0 = i) > 0$ .

A similar notion of *irreducibility*, known as Harris or Doeblin irreducibility exists for the general state space case also.

**Theorem (Law of Large Numbers for Markov Chains).**  $\{X_n\}_{n \geq 0}$  is a Markov chain with a countable state space  $S$  and a transition probability matrix  $P$ . Suppose it is *irreducible and has a stationary probability distribution*  $\boldsymbol{\pi} \equiv (\pi_i : i \in S)$  as defined in (14). Then, for any bounded function  $h : S \rightarrow R$  and for any initial distribution of  $X_0$

$$\frac{1}{n} \sum_{i=0}^{n-1} h(X_i) \rightarrow \sum_j h(j) \pi_j \quad (16)$$

in probability as  $n \rightarrow \infty$ .

A similar law of large numbers (LLN) holds when the state space  $S$  is not countable. The limit value in (16) will be the integral of  $h$  with respect to the stationary distribution  $\pi$ . A sufficient condition for the validity of this LLN

is that the Markov chain  $\{X_n\}$  be Harris irreducible and have a stationary distribution  $\pi$ .

### How is this Useful?

A probability distribution  $\pi$  on a set  $S$  is given. Want to compute the “integral of  $h$  with respect to  $\pi$ ”, which reduces to  $\sum_j h(j)\pi_j$  in the countable case.

Look for an irreducible Markov chain  $\{X_n\}$  with state space  $S$  and stationary distribution  $\pi$ . Starting from some initial value  $X_0$ , run the Markov chain  $\{X_j\}$  for a period of time, say  $0, 1, 2, \dots, n-1$  and consider as an estimate

$$\mu_n = \frac{1}{n} \sum_0^{n-1} h(X_j). \quad (17)$$

By the LLN (16),  $\mu_n$  will be close to  $\sum_j h(j)\pi_j$  for large  $n$ .

This technique is called *Markov chain Monte Carlo* (MCMC).

To approximate  $\pi(A) \equiv \sum_{j \in A} \pi_j$  for some  $A \subset S$  simply consider

$$\pi_n(A) \equiv \frac{1}{n} \sum_0^{n-1} I_A(X_j) \rightarrow \pi(A),$$

where  $I_A(X_j) = 1$  if  $X_j \in A$  and 0 otherwise.

An irreducible Markov chain  $\{X_n\}$  with a countable state space  $S$  is called *aperiodic* if for some  $i \in S$  the greatest common divisor, g.c.d.  $\{n : p_{ii}^{(n)} > 0\} = 1$ . Then, in addition to the LLN (16), the following result on the convergence of  $P(X_n = j)$  holds.

$$\sum_j |P(X_n = j) - \pi_j| \rightarrow 0 \quad (18)$$

as  $n \rightarrow \infty$ , for any initial distribution of  $X_0$ . In other words, for large  $n$  the probability distribution of  $X_n$  will be close to  $\pi$ . There exists a result similar to (18) for the general state space case also.

This suggests that instead of doing one run of length  $n$ , one could do  $N$  independent runs each of length  $m$  so that  $n = Nm$  and then from the  $i^{\text{th}}$

run use only the  $m^{\text{th}}$  observation, say,  $X_{m,i}$  and consider the estimate

$$\tilde{\mu}_{N,m} \equiv \frac{1}{N} \sum_{i=1}^N h(X_{m,i}). \quad (19)$$

### Metropolis-Hastings Algorithm

Very general MCMC method with wide applications. Idea is **not to directly simulate from the given target density (which may be computationally difficult), but to simulate an easy Markov chain that has this target density as the stationary distribution.**

Let  $\pi$  be the target probability distribution on  $S$ , a finite or countable set. Let  $Q \equiv ((q_{ij}))$  be a transition probability matrix such that for each  $i$ , it is computationally easy to generate a sample from the distribution  $\{q_{ij} : j \in S\}$ . Generate a Markov chain  $\{X_n\}$  as follows. **If  $X_n = i$ , first sample from the distribution  $\{q_{ij} : j \in S\}$  and denote that observation  $Y_n$ . Then, choose  $X_{n+1}$  from the two values  $X_n$  and  $Y_n$  according to**

$$P(X_{n+1} = Y_n | X_n, Y_n) = \rho(X_n, Y_n) = 1 - P(X_{n+1} = X_n | X_n, Y_n),$$

where the “acceptance probability”  $\rho(\cdot, \cdot)$  is given by

$$\rho(i, j) = \min \left\{ \frac{\pi_j q_{ji}}{\pi_i q_{ij}}, 1 \right\} \text{ for all } (i, j) \text{ such that } \pi_i q_{ij} > 0.$$

$\{X_n\}$  is a Markov chain with transition probability matrix  $P = ((p_{ij}))$  given by

$$p_{ij} = \begin{cases} q_{ij} \rho_{ij} & j \neq i, \\ 1 - \sum_{k \neq i} p_{ik}, & j = i. \end{cases} \quad (20)$$

$Q$  is called the “proposal transition probability” and  $\rho$  the “acceptance probability”. A significant feature of this transition mechanism  $P$  is that  $P$  and  $\pi$  satisfy

$$\pi_i p_{ij} = \pi_j p_{ji} \text{ for all } i, j. \quad (21)$$

This implies that for any  $j$

$$\sum_i \pi_i p_{ij} = \pi_j \sum_i p_{ji} = \pi_j, \quad (22)$$

or,  $\pi$  is a stationary probability distribution for  $P$ .

Suppose  $S$  is irreducible with respect to  $Q$  and  $\pi_i > 0$  for all  $i$  in  $S$ . It can then be shown that  $P$  is irreducible, and because it has a stationary distribution  $\pi$ , LLN (16) is available. This algorithm is thus a very flexible and useful one. The choice of  $Q$  is subject only to the condition that  $S$  is irreducible with respect to  $Q$ . A sufficient condition for the aperiodicity of  $P$  is that  $p_{ii} > 0$  for some  $i$  or equivalently

$$\sum_{j \neq i} q_{ij} \rho_{ij} < 1.$$

A sufficient condition for this is that there exists a pair  $(i, j)$  such that  $\pi_i q_{ij} > 0$  and  $\pi_j q_{ji} < \pi_i q_{ij}$ .

Recall that **if  $P$  is aperiodic, then both the LLN (16) and (18) hold.**

If  $S$  is not finite or countable but is a continuum and the target distribution  $\pi(\cdot)$  has a density  $p(\cdot)$ , then one proceeds as follows: Let  $Q$  be a transition function such that for each  $x$ ,  $Q(x, \cdot)$  has a density  $q(x, y)$ . Then proceed as in the discrete case but set the “acceptance probability”  $\rho(x, y)$  to be

$$\rho(x, y) = \min \left\{ \frac{p(y)q(y, x)}{p(x)q(x, y)}, 1 \right\}$$

for all  $(x, y)$  such that  $p(x)q(x, y) > 0$ .

A particularly useful feature of the above algorithm is that **it is enough to know  $p(\cdot)$  upto a multiplicative constant as the “acceptance probability”  $\rho(\cdot, \cdot)$  needs only the ratios  $p(y)/p(x)$  or  $\pi_i/\pi_j$ .**

This assures us that in Bayesian applications it is not necessary to have the normalizing constant of the posterior density available for computation of the posterior quantities of interest.

### Gibbs Sampling

Most of the new problems that Bayesians are asked to solve are high-dimensional: e.g. micro-arrays, image processing. Bayesian analysis of such problems involve target (posterior) distributions that are high-dimensional multivariate distributions.

In image processing, typically one has  $N \times N$  square grid of pixels with  $N = 256$  and each pixel has  $k \geq 2$  possible values. Each configuration has

$(256)^2$  components and the state space  $S$  has  $k^{(256)^2}$  configurations. How does one simulate a random configuration from a target distribution over such a large  $S$ ?

*Gibbs sampler* is a technique especially suitable for generating an irreducible aperiodic Markov chain that has as its stationary distribution a target distribution in a high-dimensional space having some special structure.

The most interesting aspect of this technique: **to run this Markov chain, it suffices to generate observations from univariate distributions.**

The *Gibbs sampler* in the context of a bivariate probability distribution can be described as follows. Let  $\pi$  be a target probability distribution of a bivariate random vector  $(X, Y)$ . For each  $x$ , let  $P(x, \cdot)$  be the conditional probability distribution of  $Y$  given  $X = x$ . Similarly, let  $Q(y, \cdot)$  be the conditional probability distribution of  $X$  given  $Y = y$ . Note that for each  $x$ ,  $P(x, \cdot)$  is a univariate distribution, and for each  $y$ ,  $Q(y, \cdot)$  is also a univariate distribution. Now generate a bivariate Markov chain  $Z_n = (X_n, Y_n)$  as follows:

Start with some  $X_0 = x_0$ . Generate an observation  $Y_0$  from the distribution  $P(x_0, \cdot)$ . Then generate an observation  $X_1$  from  $Q(Y_0, \cdot)$ . Next generate an observation  $Y_1$  from  $P(X_1, \cdot)$  and so on. At stage  $n$  if  $Z_n = (X_n, Y_n)$  is known, then generate  $X_{n+1}$  from  $Q(Y_n, \cdot)$  and  $Y_{n+1}$  from  $P(X_{n+1}, \cdot)$ .

If  $\pi$  is a discrete distribution concentrated on  $\{(x_i, y_j) : 1 \leq i \leq K, 1 \leq j \leq L\}$  and if  $\pi_{ij} = \pi(x_i, y_j)$  then  $P(x_i, y_j) = \pi_{ij}/\pi_i$  and  $Q(y_j, x_i) = \pi_{ij}/\pi_{.j}$ , where  $\pi_i = \sum_j \pi_{ij}$ ,  $\pi_{.j} = \sum_i \pi_{ij}$ . Thus the transition probability matrix  $R = ((r_{(ij),(k\ell)}))$  for the  $\{Z_n\}$  chain is given by

$$\begin{aligned} r_{(ij),(k\ell)} &= Q(y_j, x_k)P(x_k, y_\ell) \\ &= \frac{\pi_{kj} \pi_{k\ell}}{\pi_{.j} \pi_k}. \end{aligned}$$

Verify that this **chain is irreducible, aperiodic, and has  $\pi$  as its stationary distribution.** Thus LLN (16) and (18) hold in this case. Thus for large  $n$ ,  $Z_n$  can be viewed as a sample from a distribution that is close to  $\pi$  and one can approximate  $\sum_{i,j} h(i, j)\pi_{ij}$  by  $\sum_{i=1}^n h(X_i, Y_i)/n$ .

Illustration: Consider sampling from  $\begin{pmatrix} X \\ Y \end{pmatrix} \sim N_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right)$ . The

conditional distribution of  $X$  given  $Y = y$  and that of  $Y$  given  $X = x$  are

$$X|Y = y \sim N(\rho y, 1 - \rho^2) \text{ and } Y|X = x \sim N(\rho x, 1 - \rho^2). \quad (23)$$

Using this property, Gibbs sampling proceeds as follows: Generate  $(X_n, Y_n)$ ,  $n = 0, 1, 2, \dots$ , by starting from an arbitrary value  $x_0$  for  $X_0$ , and repeat the following steps for  $i = 0, 1, \dots, n$ .

1. Given  $x_i$  for  $X$ , draw a random deviate from  $N(\rho x_i, 1 - \rho^2)$  and denote it by  $Y_i$ .
2. Given  $y_i$  for  $Y$ , draw a random deviate from  $N(\rho y_i, 1 - \rho^2)$  and denote it by  $X_{i+1}$ .

The theory of Gibbs sampling tells us that if  $n$  is large, then  $(x_n, y_n)$  is a random draw from a distribution that is close to  $N_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right)$ .

Multivariate extension:  $\pi$  is a probability distribution of a  $k$ -dimensional random vector  $(X_1, X_2, \dots, X_k)$ . If  $\mathbf{u} = (u_1, u_2, \dots, u_k)$  is any  $k$ -vector, let  $\mathbf{u}_{-i} = (u_1, u_2, \dots, u_{i-1}, u_{i+1}, \dots, u_k)$  be the  $k - 1$  dimensional vector resulting by dropping the  $i$ th component  $u_i$ . Let  $\pi_i(\cdot|\mathbf{x}_{-i})$  denote the univariate conditional distribution of  $X_i$  given that  $\mathbf{X}_{-i} \equiv (X_1, X_2, X_{i-1}, X_{i+1}, \dots, X_k) = \mathbf{x}_{-i}$ . Starting with some initial value for  $\mathbf{X}_0 = (x_{01}, x_{02}, \dots, x_{0k})$  generate  $\mathbf{X}_1 = (X_{11}, X_{12}, \dots, X_{1k})$  sequentially by generating  $X_{11}$  according to the univariate distribution  $\pi_1(\cdot|x_{0_{-1}})$  and then generating  $X_{12}$  according to  $\pi_2(\cdot|(X_{11}, x_{03}, x_{04}, \dots, x_{0k}))$  and so on.

The most important feature to recognize here is that [all the univariate conditional distributions,  \$X\_i|\mathbf{X}\_{-i} = \mathbf{x}\_{-i}\$ , known as \*full conditionals\* should easily allow sampling](#) from them. This is the case in most hierarchical Bayes problems. Thus, the Gibbs sampler is particularly well adapted for Bayesian computations with hierarchical priors.

### Rao-Blackwellization

The variance reduction idea of the famous *Rao-Blackwell theorem* in the presence of auxiliary information can be used to provide improved estimators when MCMC procedures are adopted.

**Theorem (Rao-Blackwell)** Let  $\delta(X_1, X_2, \dots, X_n)$  be an estimator of  $\theta$  with finite variance. Suppose that  $T$  is sufficient for  $\theta$ , and let  $\delta^*(T)$ , defined

by  $\delta^*(t) = E(\delta(X_1, X_2, \dots, X_n)|T = t)$ , be the conditional expectation of  $\delta(X_1, X_2, \dots, X_n)$  given  $T = t$ . Then

$$E(\delta^*(T) - \theta)^2 \leq E(\delta(X_1, X_2, \dots, X_n) - \theta)^2.$$

The inequality is strict unless  $\delta = \delta^*$ , or equivalently,  $\delta$  is already a function of  $T$ .

By the property of iterated conditional expectation,

$$E(\delta^*(T)) = E[E(\delta(X_1, X_2, \dots, X_n)|T)] = E(\delta(X_1, X_2, \dots, X_n)).$$

Therefore, to compare the mean squared errors (MSE) of the two estimators, compare their variances only. Now,

$$\begin{aligned} \text{Var}(\delta(X_1, X_2, \dots, X_n)) &= \text{Var}[E(\delta|T)] + E[\text{Var}(\delta|T)] \\ &= \text{Var}(\delta^*) + E[\text{Var}(\delta|T)] > \text{Var}(\delta^*), \end{aligned}$$

unless  $\text{Var}(\delta|T) = 0$ , which is the case only if  $\delta$  is a function of  $T$ .

The Rao–Blackwell theorem involves two key steps: variance reduction by conditioning and conditioning by a sufficient statistic. The first step is based on the *analysis of variance* formula: For any two random variables  $S$  and  $T$ , because

$$\text{Var}(S) = \text{Var}(E(S|T)) + E(\text{Var}(S|T)),$$

one can reduce the variance of a random variable  $S$  by taking conditional expectation given some auxiliary information  $T$ . This can be exploited in MCMC.

$(X_j, Y_j), j = 1, 2, \dots, N$ : a single run of the Gibbs sampler algorithm with a target distribution of a bivariate random vector  $(X, Y)$ . Let  $h(X)$  be a function of the  $X$  component of  $(X, Y)$  and let its mean value be  $\mu$ . Goal is to estimate  $\mu$ . A first estimate is the sample mean of the  $h(X_j), j = 1, 2, \dots, N$ . From the MCMC theory, as  $N \rightarrow \infty$ , this estimate will converge to  $\mu$  in probability. The computation of variance of this estimator is not easy due to the (Markovian) dependence of the sequence  $\{X_j, j = 1, 2, \dots, N\}$ . Suppose we make  $n$  independent runs of Gibbs sampler and generate  $(X_{ij}, Y_{ij}), j = 1, 2, \dots, N; i = 1, 2, \dots, n$ . Suppose that  $N$  is sufficiently large so that  $(X_{iN}, Y_{iN})$  can be regarded as a sample from the limiting target distribution of the Gibbs sampling scheme. Thus  $(X_{iN}, Y_{iN}), i = 1, 2, \dots, n$  form a random sample from the target distribution. Consider a second estimate of  $\mu$ —the sample mean of  $h(X_{iN}), i = 1, 2, \dots, n$ .

This estimator ignores part of the MCMC data but has the advantage that the variables  $h(X_{iN})$ ,  $i = 1, 2, \dots, n$  are independent and hence the variance of their mean is of order  $n^{-1}$ . Now applying the variance reduction idea of the Rao-Blackwell theorem by using the auxiliary information  $Y_{iN}$ ,  $i = 1, 2, \dots, n$ , one can improve this estimator as follows:

Let  $k(y) = E(h(X)|Y = y)$ . Then for each  $i$ ,  $k(Y_{iN})$  has a smaller variance than  $h(X_{iN})$  and hence the following third estimator,

$$\frac{1}{n} \sum_{i=1}^n k(Y_{iN}),$$

has a smaller variance than the second one. A crucial fact to keep in mind here is that the exact functional form of  $k(y)$  be available for implementing this improvement.

**(Example M2 continued.)**  $X|\theta \sim N(\theta, \sigma^2)$  with known  $\sigma^2$  and  $\theta \sim \text{Cauchy}(\mu, \tau)$ . Simulate  $\theta$  from the posterior distribution, but sampling directly is difficult.

Gibbs sampling: Cauchy is a scale mixture of normal densities, with the scale parameter having a Gamma distribution.

$$\begin{aligned} \pi(\theta) &\propto (\tau^2 + (\theta - \mu)^2)^{-1} \\ &\propto \int_0^\infty \left(\frac{\lambda}{2\pi\tau^2}\right)^{1/2} \exp\left(-\frac{\lambda}{2\tau^2}(\theta - \mu)^2\right) \lambda^{1/2-1} \exp\left(-\frac{\lambda}{2}\right) d\lambda, \end{aligned}$$

so that  $\pi(\theta)$  may be considered the marginal prior density from the joint prior density of  $(\theta, \lambda)$  where

$$\theta|\lambda \sim N(\mu, \tau^2/\lambda) \text{ and } \lambda \sim \text{Gamma}(1/2, 1/2).$$

This implicit hierarchical prior structure implies:  $\pi(\theta|x)$  is the marginal density from  $\pi(\theta, \lambda|x)$ .

Full conditionals of  $\pi(\theta, \lambda|x)$  are standard distributions:

$$\theta|\lambda, x \sim N\left(\frac{\tau^2}{\tau^2 + \lambda\sigma^2}x + \frac{\lambda\sigma^2}{\tau^2 + \lambda\sigma^2}\mu, \frac{\tau^2\sigma^2}{\tau^2 + \lambda\sigma^2}\right), \quad (24)$$

$$\lambda|\theta, x \sim \lambda|\theta \sim \text{Exponential}\left(\frac{\tau^2 + (\theta - \mu)^2}{2\tau^2}\right). \quad (25)$$

Thus, the Gibbs sampler will use (24) and (25) to generate  $(\theta, \lambda)$  from  $\pi(\theta, \lambda|x)$ .

**Example M5.**  $X$  = number of defectives in the daily production of a product.  $(X | Y, \theta) \sim \text{binomial}(Y, \theta)$ , where  $Y$ , a day's production, is Poisson with known mean  $\lambda$ , and  $\theta$  is the probability that any product is defective. The difficulty is that  $Y$  is not observable, and inference has to be made on the basis of  $X$  only. Prior:  $(\theta | Y = y) \sim \text{Beta}(\alpha, \gamma)$ , with known  $\alpha$  and  $\gamma$  independent of  $Y$ . Bayesian analysis here is not difficult because the posterior distribution of  $\theta | X = x$  can be obtained as follows. First,  $X | \theta \sim \text{Poisson}(\lambda\theta)$ . Next,  $\theta \sim \text{Beta}(\alpha, \gamma)$ . Therefore,

$$\pi(\theta | X = x) \propto \exp(-\lambda\theta)\theta^{x+\alpha-1}(1-\theta)^{\gamma-1}, 0 < \theta < 1. \quad (26)$$

This is not a standard distribution, and hence posterior quantities cannot be obtained in closed form. Instead of focusing on  $\theta | X$  directly, view it as a marginal component of  $(Y, \theta | X)$ . Check that the full conditionals of this are given by

$Y | X = x, \theta \sim x + \text{Poisson}(\lambda(1 - \theta))$ , and  
 $\theta | X = x, Y = y \sim \text{Beta}(\alpha + x, \gamma + y - x)$   
 both of which are standard distributions.

**Example M5 continued.** It is actually possible here to sample from the posterior distribution using the *accept-reject* Monte Carlo method:

Let  $g(\mathbf{x})/K$  be the target density, where  $K$  is the possibly unknown normalizing constant of the unnormalized density  $g$ . Suppose  $h(\mathbf{x})$  is a density that can be simulated by a known method and is close to  $g$ , and suppose there exists a known constant  $c > 0$  such that  $g(\mathbf{x}) < ch(\mathbf{x})$  for all  $\mathbf{x}$ . Then, to simulate from the target density, the following two steps suffice. Step 1. Generate  $\mathbf{Y} \sim h$  and  $U \sim U(0, 1)$ ;

Step 2. Accept  $\mathbf{X} = \mathbf{Y}$  if  $U \leq g(\mathbf{Y})/\{ch(\mathbf{Y})\}$ ; return to Step 1 otherwise.

The optimal choice for  $c$  is  $\sup\{g(\mathbf{x})/h(\mathbf{x})\}$ .

In Example M5, from (26),

$$g(\theta) = \exp(-\lambda\theta)\theta^{x+\alpha-1}(1-\theta)^{\gamma-1}I\{0 \leq \theta \leq 1\},$$

so that  $h(\theta)$  may be chosen to be the density of  $\text{Beta}(x + \alpha, \gamma)$ . Then, with the above-mentioned choice for  $c$ , if  $\theta \sim \text{Beta}(x + \alpha, \gamma)$  is generated in Step 1, its 'acceptance probability' in Step 2 is simply  $\exp(-\lambda\theta)$ .

Even though this method works here, let us see how the Metropolis-Hastings algorithm can be applied.

The required Markov chain is generated by taking the transition density  $q(z, y) = q(y|z) = h(y)$ , independently of  $z$ . Then the acceptance probability

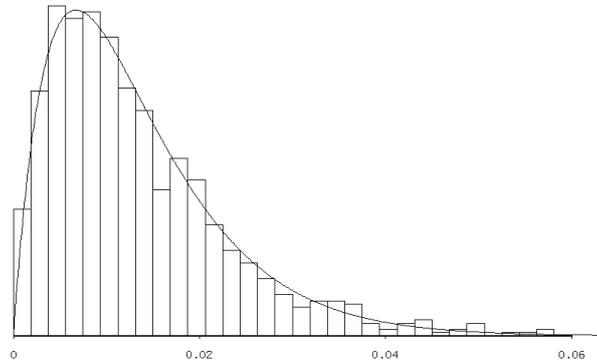


Figure 1: M-H frequency histogram and true posterior density.

is

$$\begin{aligned}\rho(z, y) &= \min \left\{ \frac{g(y)h(z)}{g(z)h(y)}, 1 \right\} \\ &= \min \{ \exp(-\lambda(y-z)), 1 \}.\end{aligned}$$

The steps involved in this “independent” M-H algorithm are:

Start at  $t = 0$  with a value  $x_0$  in the support of the target distribution; in this case,  $0 < x_0 < 1$ . Given  $x_t$ , generate the next value in the chain as given below.

(a) Draw  $Y_t$  from  $\text{Beta}(x + \alpha, \gamma)$ .

(b) Let

$$x_{(t+1)} = \begin{cases} Y_t & \text{with probability } \rho_t \\ x_t & \text{otherwise,} \end{cases}$$

where  $\rho_t = \min\{\exp(-\lambda(Y_t - x_t)), 1\}$ .

(c) Set  $t = t + 1$  and go to step (a).

Run this chain until  $t = n$ , a suitably chosen large integer. In our example, for  $x = 1$ ,  $\alpha = 1$ ,  $\gamma = 49$  and  $\lambda = 100$ , we simulated such a Markov chain. The resulting frequency histogram is shown in Figure below, with the true posterior density super-imposed on it.

## 10 Empirical Bayes Methods for High Dimensional Problems

This is becoming popular again, this time for ‘high dimensional’ problems. Astronomers routinely estimate characteristics of millions of similar astronomical objects – distance, radial velocity whatever. Consider the data:

$$(\mathbf{X}_1 = \begin{pmatrix} X_{11} \\ X_{12} \\ \vdots \\ X_{1n} \end{pmatrix}, \mathbf{X}_2 = \begin{pmatrix} X_{21} \\ X_{22} \\ \vdots \\ X_{2n} \end{pmatrix}, \dots, \mathbf{X}_p = \begin{pmatrix} X_{p1} \\ X_{p2} \\ \vdots \\ X_{pn} \end{pmatrix}).$$

$\mathbf{X}_j$  represents  $n$  repeated independent observations on the  $j$ th object,  $j = 1, 2, \dots, p$ . The important point is  $n$  is small, 2, 5, or 10, whereas  $p$  is large, such as a million.

Suppose  $X_{j1}, \dots, X_{jn}$  measure  $\mu_j$  with variability  $\sigma^2$ .

**Problem: Maximum likelihood can give wrong estimates**

Take  $n = 2$  and suppose

$$\begin{pmatrix} X_{j1} \\ X_{j2} \end{pmatrix} \sim N \left( \begin{pmatrix} \mu_j \\ \mu_j \end{pmatrix}, \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix} \right), \quad j = 1, 2, \dots, p.$$

i.e., we measure  $\mu_j$  with 2 independent measurements, each coming with a  $N(0, \sigma^2)$  error added to it; we do this for a very large number  $p$  of objects.

**What is the MLE of  $\sigma^2$ ?**

$$\begin{aligned} l(\mu_1, \dots, \mu_p; \sigma^2 | \mathbf{x}_1, \dots, \mathbf{x}_p) &= f(\mathbf{x}_1, \dots, \mathbf{x}_p | \mu_1, \dots, \mu_p; \sigma^2) \\ &= \prod_{j=1}^p \prod_{i=1}^2 f(x_{ji} | \mu_j, \sigma^2) \\ &= (2\pi\sigma^2)^{-p} \exp\left(-\frac{1}{2\sigma^2} \sum_{j=1}^p \sum_{i=1}^2 (x_{ji} - \mu_j)^2\right) \\ &= (2\pi\sigma^2)^{-p} \exp\left(-\frac{1}{2\sigma^2} \sum_{j=1}^p \left[ \sum_{i=1}^2 (x_{ji} - \bar{x}_j)^2 + 2(\bar{x}_j - \mu_j)^2 \right]\right). \end{aligned}$$

$\hat{\mu}_j = \bar{x}_j = (x_{j1} + x_{j2})/2$  and

$$\begin{aligned}\hat{\sigma}^2 &= \frac{1}{2p} \sum_{j=1}^p \sum_{i=1}^2 (x_{ji} - \bar{x}_j)^2 \\ &= \frac{1}{2p} \sum_{j=1}^p \left[ \left( x_{j1} - \frac{x_{j1} + x_{j2}}{2} \right)^2 + \left( x_{j2} - \frac{x_{j1} + x_{j2}}{2} \right)^2 \right] \\ &= \frac{1}{2p} \sum_{j=1}^p 2 \frac{(x_{j1} - x_{j2})^2}{4} = \frac{1}{4p} \sum_{j=1}^p (x_{j1} - x_{j2})^2.\end{aligned}$$

Since  $X_{j1} - X_{j2} \sim N(0, 2\sigma^2)$ ,  $j = 1, 2, \dots$ ,

$$\begin{aligned}\frac{1}{p} \sum_{j=1}^p (X_{j1} - X_{j2})^2 &\xrightarrow[p \rightarrow \infty]{P} 2\sigma^2, \text{ so that} \\ \hat{\sigma}^2 = \frac{1}{4p} \sum_{j=1}^p (X_{j1} - X_{j2})^2 &\xrightarrow[p \rightarrow \infty]{P} \frac{\sigma^2}{2}, \text{ and not } \sigma^2.\end{aligned}$$

Good estimates for  $\sigma^2$  do exist, for example,

$$\frac{1}{2p} \sum_{j=1}^p (X_{j1} - X_{j2})^2 \xrightarrow[p \rightarrow \infty]{P} 2\sigma^2.$$

**What is going wrong here?**

This is not a *small p, large n* problem, but a *small n, large p* problem. i.e. a high dimensional problem, so needs care!

As  $p \rightarrow \infty$ , there are too many parameters to estimate and the likelihood function is unable to see where information lies, so tries to distribute it everywhere.

**What is the way out?** Go Bayesian!

There is a lot of information available on  $\sigma^2$  (note  $\sum_{j=1}^p (X_{j1} - X_{j2})^2 \sim 2\sigma^2 \chi_p^2$ ) but very little on individual  $\mu_j$ . However, if  $\mu_j$  are ‘similar’, there is a lot of information on where they come from, because we get to see  $p$  samples,  $p$  large.

Suppose we are interested in  $\mu_j$ . How can we use the above information? Model as follows:

$\bar{X}_j | \mu_j, \sigma^2 \sim N(\mu_j, \sigma^2/2)$ ,  $j = 1, \dots, p$ , independent observations.

$\sigma^2$  may be assumed known, since a reliable estimate  $\hat{\sigma}^2 = \frac{1}{2p} \sum_{j=1}^p (X_{j1} - X_{j2})^2$  is available. Express the information that  $\mu_j$  are ‘similar’ in the form:  $\mu_j, j = 1, \dots, p$  is a random sample (collection) from  $N(\eta, \tau^2)$ . Where do we get the  $\eta$  and  $\tau^2$ , the prior mean and prior variance?

Marginally (or in predictive sense)  $\bar{X}_j, j = 1, \dots, p$  is a random sample from  $N(\mu_0, \tau^2 + \sigma^2/2)$ . Use this random sample.

Estimate  $\eta$  by  $\hat{\eta} = \bar{\bar{X}} = \frac{1}{p} \sum \bar{X}_j$  and  $\tau^2$  by  $\hat{\tau}^2 = \left( \frac{1}{p-1} \sum_{j=1}^p (\bar{X}_j - \bar{\bar{X}})^2 - \sigma^2/2 \right)^+$ .

Now one could pretend that the prior for  $(\mu_1, \dots, \mu_p)$  is  $N(\hat{\eta}, \hat{\tau}^2)$  and compute the Bayes estimates for  $\mu_j$ :

$$E(\mu_j | \mathbf{X}_1, \dots, \mathbf{X}_p) = (1 - \hat{B})\bar{X}_j + \hat{B}\bar{\bar{X}},$$

where  $\hat{B} = \frac{\sigma^2/2}{\sigma^2/2 + \hat{\tau}^2}$ . If instead of 2 observations, each sample has  $n$  observations, replace 2 by  $n$ . This is called *Empirical Bayes* since the prior is estimated using data. There is also a fully Bayesian counter-part called *Hierarchical Bayes*.

# Chapter 14

## BAYESIAN ANALYSIS

*Notes by Tom Lored*

# Introduction to Bayesian Inference

## Lecture 1: Fundamentals

Tom Loredo

Dept. of Astronomy, Cornell University

<http://www.astro.cornell.edu/staff/loredo/bayes/>  
Lectures <http://inference.astro.cornell.edu/INPE09/>

CASt Summer School — 11 June 2010

1 / 59

## Lecture 1: Fundamentals

- ① The big picture
- ② A flavor of Bayes:  $\chi^2$  confidence/credible regions
- ③ Foundations: Logic & probability theory
- ④ Probability theory for data analysis: Three theorems
- ⑤ Inference with parametric models
  - Parameter Estimation
  - Model Uncertainty

2 / 59

## Bayesian Fundamentals

- ① The big picture
- ② A flavor of Bayes:  $\chi^2$  confidence/credible regions
- ③ Foundations: Logic & probability theory
- ④ Probability theory for data analysis: Three theorems
- ⑤ Inference with parametric models
  - Parameter Estimation
  - Model Uncertainty

3 / 59

## Scientific Method

*Science is more than a body of knowledge; it is a way of thinking.  
The method of science, as stodgy and grumpy as it may seem,  
is far more important than the findings of science.*  
—Carl Sagan

Scientists *argue!*

Argument  $\equiv$  Collection of statements comprising an act of reasoning from *premises* to a *conclusion*

A key goal of science: Explain or predict *quantitative measurements* (data!)

*Data analysis*: Constructing and appraising arguments that reason from data to interesting scientific conclusions (explanations, predictions)

4 / 59

## The Role of Data

*Data do not speak for themselves!*

We don't just *tabulate* data, we *analyze* data.

We gather data so they may speak for or against existing hypotheses, and guide the formation of new hypotheses.

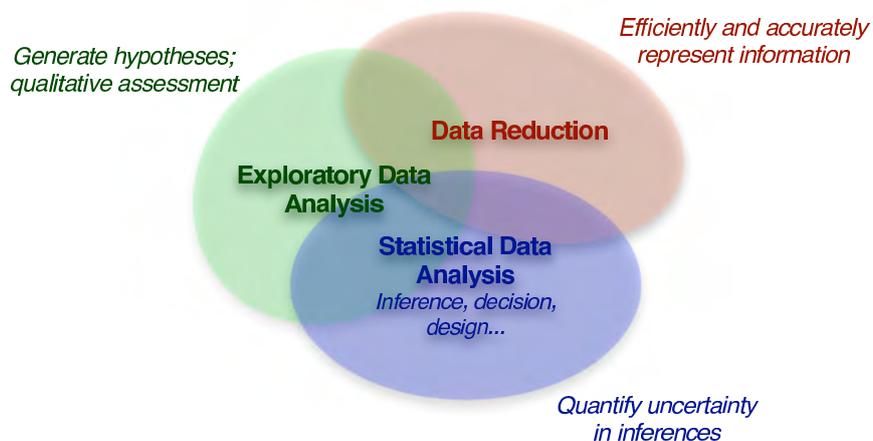
A key role of data in science is to be among the premises in scientific arguments.

5 / 59

## Data Analysis

*Building & Appraising Arguments Using Data*

### Modes of Data Analysis



*Statistical inference* is but one of several interacting modes of analyzing data.

6 / 59

## Bayesian Statistical Inference

- A different approach to *all* statistical inference problems (i.e., not just another method in the list: BLUE, maximum likelihood,  $\chi^2$  testing, ANOVA, survival analysis . . . )
- Foundation: Use probability theory to quantify the strength of arguments (i.e., a more abstract view than restricting PT to describe variability in repeated “random” experiments)
- Focuses on *deriving consequences of modeling assumptions* rather than *devising and calibrating procedures*

7 / 59

## Frequentist vs. Bayesian Statements

“I find conclusion  $C$  based on data  $D_{\text{obs}}$  . . . .”

### *Frequentist assessment*

“It was found with a procedure that’s right 95% of the time over the set  $\{D_{\text{hyp}}\}$  that includes  $D_{\text{obs}}$ .”

Probabilities are properties of *procedures*, not of particular results.

### *Bayesian assessment*

“The strength of the chain of reasoning from  $D_{\text{obs}}$  to  $C$  is 0.95, on a scale where 1= certainty.”

Probabilities are properties of *specific results*.

Long-run performance must be separately evaluated (and is typically good by frequentist criteria).

8 / 59

## Bayesian Fundamentals

- ① The big picture
- ② A flavor of Bayes:  $\chi^2$  confidence/credible regions
- ③ Foundations: Logic & probability theory
- ④ Probability theory for data analysis: Three theorems
- ⑤ Inference with parametric models
  - Parameter Estimation
  - Model Uncertainty

9 / 59

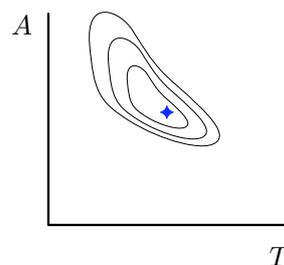
### Estimating Parameters Via $\chi^2$

Collect data  $D_{\text{obs}} = \{d_i, \sigma_i\}$ , fit with 2-parameter model via  $\chi^2$ :

$$\chi^2(A, T) = \sum_{i=1}^N \frac{[d_i - f_i(A, T)]^2}{\sigma_i^2}$$

Two classes of variables

- Data (samples)  $d_i$  — Known, define  $N$ -D *sample space*
- Parameters  $\theta = (A, T)$  — Unknown, define 2-D *parameter space*

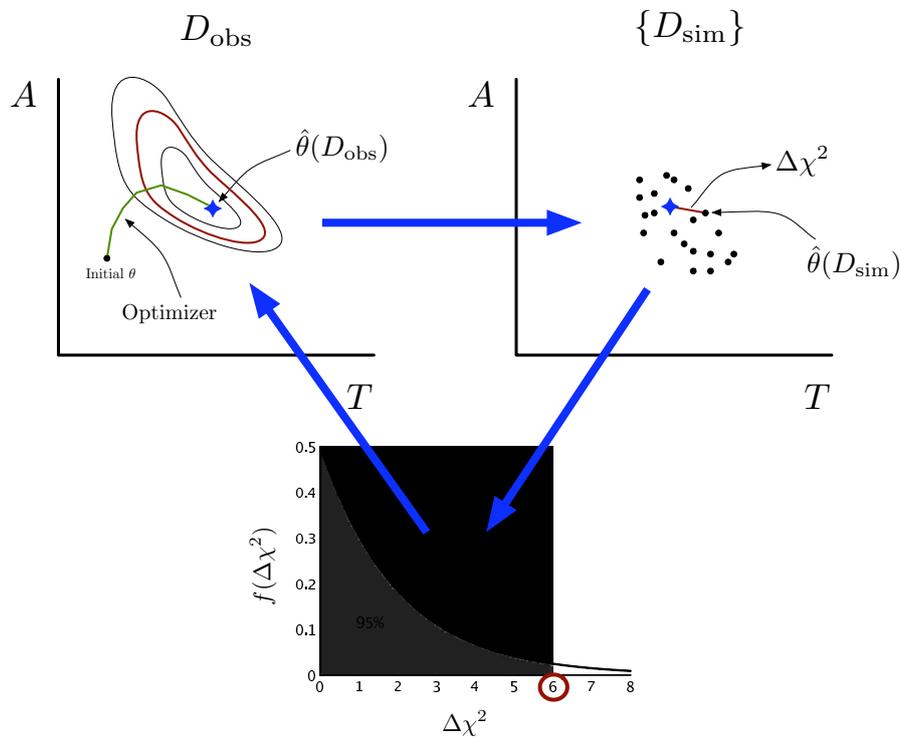


“Best fit”  $\hat{\theta} = \arg \min_{\theta} \chi^2(\theta)$

Report uncertainties via  $\chi^2$  contours, but how do we quantify uncertainty vs. contour level?

10 / 59

## Frequentist: Parametric Bootstrap



11 / 59

## Parametric Bootstrap Algorithm

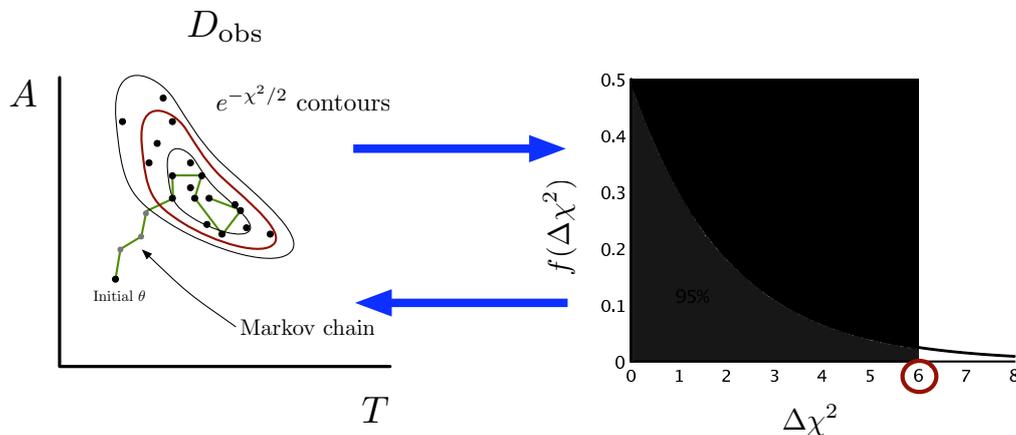
Monte Carlo algorithm for finding approximate confidence regions:

1. Pretend true params  $\theta^* = \hat{\theta}(D_{\text{obs}})$  ("plug-in approx'n")
2. Repeat  $N$  times:
  1. Simulate a dataset from  $p(D_{\text{sim}}|\theta^*)$   
 $\rightarrow \chi^2_{D_{\text{sim}}}(\theta)$
  2. Find min  $\chi^2$  estimate  $\hat{\theta}(D_{\text{sim}})$
  3. Calculate  $\Delta\chi^2 = \chi^2(\theta^*) - \chi^2(\hat{\theta}_{D_{\text{sim}}})$
4. Histogram the  $\Delta\chi^2$  values to find coverage vs.  $\Delta\chi^2$   
 (fraction of sim'ns with smaller  $\Delta\chi^2$ )

Result is approximate even for  $N \rightarrow \infty$  because  $\theta^* \neq \hat{\theta}(D_{\text{obs}})$ .

12 / 59

## Bayesian: Posterior Sampling Via MCMC



13 / 59

## Posterior Sampling Algorithm

Monte Carlo algorithm for finding credible regions:

1. Create a RNG that can sample  $\theta$  from  $p(\theta|D_{\text{obs}}) \propto e^{-\chi^2(\theta)/2}$
2. Draw  $N$  samples; record  $\theta_i$  and  $q_i = \chi^2(\theta_i)$
3. Sort the samples by the  $q_i$  values
4. A credible region of probability  $P$  is the  $\theta$  region spanned by the  $100P\%$  of samples with highest  $q_i$

Note that no dataset other than  $D_{\text{obs}}$  is ever considered.

The only approximation is the use of Monte Carlo.

These very different procedures produce the same regions for linear models with Gaussian error distributions.

These procedures produce *different* regions in general.

14 / 59

## Bayesian Fundamentals

- ① The big picture
- ② A flavor of Bayes:  $\chi^2$  confidence/credible regions
- ③ Foundations: Logic & probability theory
- ④ Probability theory for data analysis: Three theorems
- ⑤ Inference with parametric models
  - Parameter Estimation
  - Model Uncertainty

15 / 59

## Logic—Some Essentials

“Logic can be defined as *the analysis and appraisal of arguments*”  
 —Gensler, *Intro to Logic*

Build arguments with propositions and logical operators/connectives:

- *Propositions*: Statements that may be true or false

$\mathcal{P}$  : Universe can be modeled with  $\Lambda$ CDM

$A$  :  $\Omega_{\text{tot}} \in [0.9, 1.1]$

$B$  :  $\Omega_{\Lambda}$  is not 0

$\overline{B}$  : “not  $B$ ,” i.e.,  $\Omega_{\Lambda} = 0$

- *Connectives*:

$A \wedge B$  :  $A$  and  $B$  are both true

$A \vee B$  :  $A$  or  $B$  is true, or both are

16 / 59

## Arguments

Argument: Assertion that an *hypothesized conclusion*,  $H$ , follows from *premises*,  $\mathcal{P} = \{A, B, C, \dots\}$  (take “,” = “and”)

Notation:

$H|\mathcal{P}$  :      Premises  $\mathcal{P}$  imply  $H$   
                    $H$  may be deduced from  $\mathcal{P}$   
                    $H$  follows from  $\mathcal{P}$   
                    $H$  is true given that  $\mathcal{P}$  is true

Arguments are (compound) propositions.

Central role of arguments → special terminology for true/false:

- A true argument is *valid*
- A false argument is *invalid* or *fallacious*

17 / 59

## Valid vs. Sound Arguments

### *Content vs. form*

- An argument is *factually correct* iff all of its *premises are true* (it has “good content”).
- An argument is *valid* iff its conclusion *follows from* its premises (it has “good form”).
- An argument is *sound* iff it is both *factually correct and valid* (it has good form and content).

Deductive logic and probability theory address *validity*.

We want to make *sound* arguments. There is no formal approach for addressing factual correctness → there is always a subjective element to an argument.

18 / 59

## Factual Correctness

### *Passing the buck*

Although logic can teach us something about validity and invalidity, it can teach us very little about factual correctness. The question of the truth or falsity of individual statements is primarily the subject matter of the sciences.

— Hardegree, *Symbolic Logic*

### *An open issue*

To test the truth or falsehood of premisses is the task of science. . . . But as a matter of fact we are interested in, and must often depend upon, the correctness of arguments whose premisses are not known to be true.

— Copi, *Introduction to Logic*

19 / 59

## Premises

- *Facts* — Things known to be true, e.g. *observed data*
- “*Obvious*” *assumptions* — Axioms, postulates, e.g., Euclid’s first 4 postulates (line segment b/t 2 points; congruency of right angles . . . )
- “*Reasonable*” or “*working*” *assumptions* — E.g., Euclid’s fifth postulate (parallel lines)
- *Desperate presumption!*
- Conclusions from other arguments

Premises define a fixed *context* in which arguments may be assessed.

Premises are considered “given”—if only for the sake of the argument!

20 / 59

## Deductive and Inductive Inference

### *Deduction—Syllogism as prototype*

Premise 1:  $A$  implies  $H$

Premise 2:  $A$  is true

Deduction:  $\therefore H$  is true

$H|\mathcal{P}$  is valid

### *Induction—Analogy as prototype*

Premise 1:  $A, B, C, D, E$  all share properties  $x, y, z$

Premise 2:  $F$  has properties  $x, y$

Induction:  $F$  has property  $z$

" $F$  has  $z$ "  $|\mathcal{P}$  is not strictly valid, but may still be rational (likely, plausible, probable); some such arguments are stronger than others

*Boolean algebra* (and/or/not over  $\{0, 1\}$ ) quantifies deduction.

*Bayesian probability theory* (and/or/not over  $[0, 1]$ ) generalizes this to quantify the strength of inductive arguments.

21 / 59

## Deductive Logic

Assess arguments by decomposing them into parts via connectives, and assessing the parts

*Validity of  $A \wedge B|\mathcal{P}$*

	$A \mathcal{P}$	$\bar{A} \mathcal{P}$
$B \mathcal{P}$	valid	invalid
$\bar{B} \mathcal{P}$	invalid	invalid

*Validity of  $A \vee B|\mathcal{P}$*

	$A \mathcal{P}$	$\bar{A} \mathcal{P}$
$B \mathcal{P}$	valid	valid
$\bar{B} \mathcal{P}$	valid	invalid

22 / 59

## Representing Deduction With $\{0, 1\}$ Algebra

$V(H|\mathcal{P}) \equiv$  Validity of argument  $H|\mathcal{P}$ :

$$\begin{aligned} V &= 0 \rightarrow \text{Argument is } \textit{invalid} \\ &= 1 \rightarrow \text{Argument is } \textit{valid} \end{aligned}$$

Then deduction can be reduced to integer multiplication and addition over  $\{0, 1\}$  (as in a computer):

$$\begin{aligned} V(A \wedge B|\mathcal{P}) &= V(A|\mathcal{P}) V(B|\mathcal{P}) \\ V(A \vee B|\mathcal{P}) &= V(A|\mathcal{P}) + V(B|\mathcal{P}) - V(A \wedge B|\mathcal{P}) \\ V(\bar{A}|\mathcal{P}) &= 1 - V(A|\mathcal{P}) \end{aligned}$$

23 / 59

## Representing Induction With $[0, 1]$ Algebra

$P(H|\mathcal{P}) \equiv$  strength of argument  $H|\mathcal{P}$

$$\begin{aligned} P &= 0 \rightarrow \text{Argument is } \textit{invalid}; \text{ premises imply } \bar{H} \\ &= 1 \rightarrow \text{Argument is } \textit{valid} \\ &\in (0, 1) \rightarrow \text{Degree of deducibility} \end{aligned}$$

### *Mathematical model for induction*

$$\begin{aligned} \text{'AND' (product rule): } P(A \wedge B|\mathcal{P}) &= P(A|\mathcal{P}) P(B|A \wedge \mathcal{P}) \\ &= P(B|\mathcal{P}) P(A|B \wedge \mathcal{P}) \end{aligned}$$

$$\text{'OR' (sum rule): } P(A \vee B|\mathcal{P}) = P(A|\mathcal{P}) + P(B|\mathcal{P}) - P(A \wedge B|\mathcal{P})$$

$$\text{'NOT': } P(\bar{A}|\mathcal{P}) = 1 - P(A|\mathcal{P})$$

24 / 59

## The Product Rule

We simply promoted the  $V$  algebra to real numbers; the only thing changed is part of the product rule:

$$\begin{aligned} V(A \wedge B|\mathcal{P}) &= V(A|\mathcal{P}) V(B|\mathcal{P}) \\ P(A \wedge B|\mathcal{P}) &= P(A|\mathcal{P}) P(B|A, \mathcal{P}) \end{aligned}$$

Suppose  $A$  implies  $B$  (i.e.,  $B|A, \mathcal{P}$  is valid). Then we don't expect  $P(A \wedge B|\mathcal{P})$  to differ from  $P(A|\mathcal{P})$ .

In particular,  $P(A \wedge A|\mathcal{P})$  must equal  $P(A|\mathcal{P})$ !

Such qualitative reasoning satisfied early probabilists that the sum and product rules were worth considering as axioms for a theory of quantified induction.

25 / 59

## Firm Foundations

Today many different lines of argument *derive* induction-as-probability from various simple and appealing requirements:

- Consistency with logic + internal consistency (Cox; Jaynes)
- “Coherence” /optimal betting (Ramsey; DeFinetti; Wald; Savage)
- Algorithmic information theory (Rissanen; Wallace & Freeman)
- Optimal information processing (Zellner)
- Avoiding problems with frequentist methods:
  - Avoiding recognizable subsets (Cornfield)
  - Avoiding stopping rule problems → likelihood principle (Birnbaum; Berger & Wolpert)

26 / 59

## Interpreting Bayesian Probabilities

If we like there is no harm in saying that a probability expresses a degree of reasonable belief. . . . ‘Degree of confirmation’ has been used by Carnap, and possibly avoids some confusion. But whatever verbal expression we use to try to convey the primitive idea, this expression cannot amount to a definition. Essentially the notion *can only be described by reference to instances where it is used*. It is intended to express *a kind of relation between data and consequence* that habitually arises in science and in everyday life, and the reader should be able to recognize the relation from examples of the circumstances when it arises.

— Sir Harold Jeffreys, *Scientific Inference*

27 / 59

## More On Interpretation

Physics uses words drawn from ordinary language—mass, weight, momentum, force, temperature, heat, etc.—but their technical meaning is more abstract than their colloquial meaning. We can map between the colloquial and abstract meanings associated with specific values by using specific instances as “calibrators.”

### A Thermal Analogy

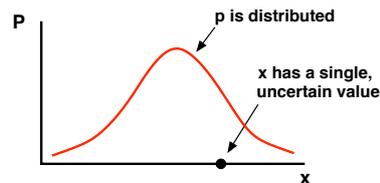
<i>Intuitive notion</i>	<i>Quantification</i>	<i>Calibration</i>
Hot, cold	Temperature, $T$	Cold as ice = 273K Boiling hot = 373K
uncertainty	Probability, $P$	Certainty = 0, 1 $p = 1/36$ : plausible as “snake’s eyes” $p = 1/1024$ : plausible as 10 heads

28 / 59

## A Bit More On Interpretation

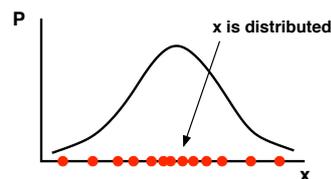
### Bayesian

Probability quantifies uncertainty in an inductive inference.  $p(x)$  describes how *probability* is distributed over the possible values  $x$  might have taken in the single case before us:



### Frequentist

Probabilities are always (limiting) rates/proportions/frequencies in a sequence of trials.  $p(x)$  describes variability, how the *values of  $x$*  would be distributed among infinitely many trials:



29 / 59

## Bayesian Fundamentals

- ① The big picture
- ② A flavor of Bayes:  $\chi^2$  confidence/credible regions
- ③ Foundations: Logic & probability theory
- ④ Probability theory for data analysis: Three theorems
- ⑤ Inference with parametric models
  - Parameter Estimation
  - Model Uncertainty

30 / 59

## Arguments Relating Hypotheses, Data, and Models

We seek to appraise scientific hypotheses in light of observed data and modeling assumptions.

Consider the data and modeling assumptions to be the premises of an argument with each of various hypotheses,  $H_i$ , as conclusions:  $H_i|D_{\text{obs}}, I$ . ( $I$  = "background information," everything deemed relevant besides the observed data)

$P(H_i|D_{\text{obs}}, I)$  measures the degree to which  $(D_{\text{obs}}, I)$  allow one to deduce  $H_i$ . It provides an ordering among arguments for various  $H_i$  that share common premises.

Probability theory tells us how to analyze and appraise the argument, i.e., how to calculate  $P(H_i|D_{\text{obs}}, I)$  from simpler, hopefully more accessible probabilities.

31 / 59

## The Bayesian Recipe

Assess hypotheses by calculating their probabilities  $p(H_i|\dots)$  conditional on known and/or presumed information using the rules of probability theory.

*Probability Theory Axioms:*

$$\text{'OR' (sum rule): } P(H_1 \vee H_2|I) = P(H_1|I) + P(H_2|I) - P(H_1, H_2|I)$$

$$\begin{aligned} \text{'AND' (product rule): } P(H_1, D|I) &= P(H_1|I) P(D|H_1, I) \\ &= P(D|I) P(H_1|D, I) \end{aligned}$$

$$\text{'NOT': } P(\overline{H_1}|I) = 1 - P(H_1|I)$$

32 / 59

## Three Important Theorems

### Bayes's Theorem (BT)

Consider  $P(H_i, D_{\text{obs}}|I)$  using the product rule:

$$\begin{aligned} P(H_i, D_{\text{obs}}|I) &= P(H_i|I) P(D_{\text{obs}}|H_i, I) \\ &= P(D_{\text{obs}}|I) P(H_i|D_{\text{obs}}, I) \end{aligned}$$

Solve for the *posterior probability*:

$$P(H_i|D_{\text{obs}}, I) = P(H_i|I) \frac{P(D_{\text{obs}}|H_i, I)}{P(D_{\text{obs}}|I)}$$

Theorem holds for any propositions, but for hypotheses & data the factors have names:

$$\text{posterior} \propto \text{prior} \times \text{likelihood}$$

$$\text{norm. const. } P(D_{\text{obs}}|I) = \text{prior predictive}$$

33 / 59

### Law of Total Probability (LTP)

Consider exclusive, exhaustive  $\{B_i\}$  ( $I$  asserts one of them must be true),

$$\begin{aligned} \sum_i P(A, B_i|I) &= \sum_i P(B_i|A, I)P(A|I) = P(A|I) \\ &= \sum_i P(B_i|I)P(A|B_i, I) \end{aligned}$$

If we do not see how to get  $P(A|I)$  directly, we can find a set  $\{B_i\}$  and use it as a “basis”—*extend the conversation*:

$$P(A|I) = \sum_i P(B_i|I)P(A|B_i, I)$$

If our problem already has  $B_i$  in it, we can use LTP to get  $P(A|I)$  from the joint probabilities—*marginalization*:

$$P(A|I) = \sum_i P(A, B_i|I)$$

34 / 59

Example: Take  $A = D_{\text{obs}}$ ,  $B_i = H_i$ ; then

$$\begin{aligned} P(D_{\text{obs}}|I) &= \sum_i P(D_{\text{obs}}, H_i|I) \\ &= \sum_i P(H_i|I)P(D_{\text{obs}}|H_i, I) \end{aligned}$$

prior predictive for  $D_{\text{obs}} =$  Average likelihood for  $H_i$   
(a.k.a. *marginal likelihood*)

### Normalization

For *exclusive, exhaustive*  $H_i$ ,

$$\sum_i P(H_i|\dots) = 1$$

35 / 59

## Well-Posed Problems

The rules express desired probabilities in terms of other probabilities.

To get a numerical value *out*, at some point we have to put numerical values *in*.

*Direct probabilities* are probabilities with numerical values determined directly by premises (via modeling assumptions, symmetry arguments, previous calculations, desperate presumption . . . ).

An inference problem is *well posed* only if all the needed probabilities are assignable based on the premises. We may need to add new assumptions as we see what needs to be assigned. We may not be entirely comfortable with what we need to assume! (Remember Euclid's fifth postulate!)

Should explore how results depend on uncomfortable assumptions ("robustness").

36 / 59

## Visualizing Bayesian Inference

Simplest case: Binary classification

- 2 hypotheses:  $\{H, C\}$
- 2 possible data values:  $\{-, +\}$

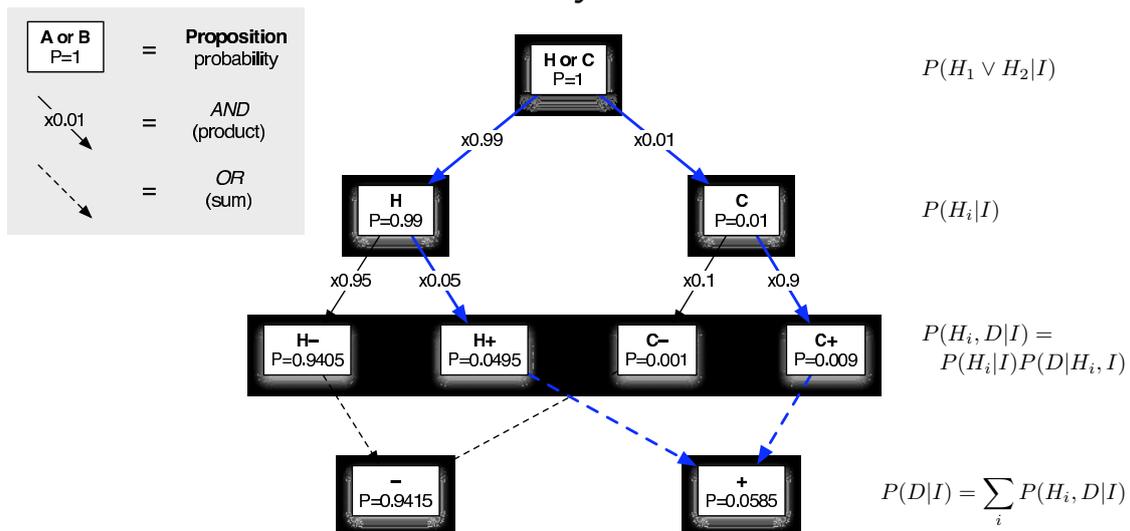
Concrete example: You test positive (+) for a medical condition. Do you have the condition (C) or not (H, "healthy")?

- Prior: Prevalence of the condition in your population is 0.1%
- Likelihood:
  - Test is 90% accurate if you have the condition:  
 $P(+|C, I) = 0.9$  ("sensitivity")
  - Test is 95% accurate if you are healthy:  
 $P(-|H, I) = 0.95$  ("specificity")

*Numbers roughly correspond to breast cancer in asymptomatic women aged 40–50, and mammography screening*  
[Gigerenzer, *Calculated Risks* (2002)]

37 / 59

### Probability Lattice



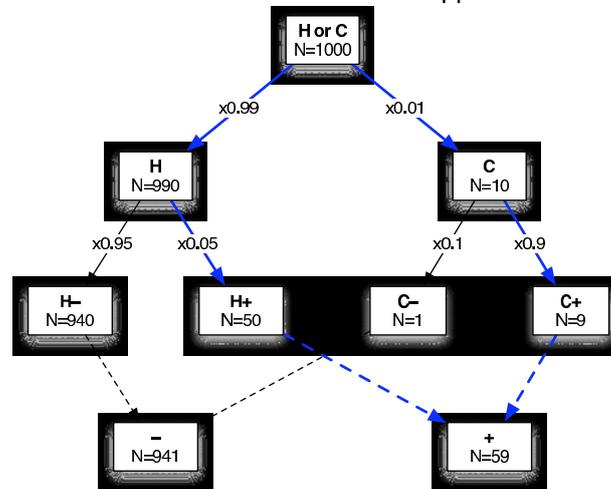
$$P(C|+, I) = \frac{0.009}{0.0585} \approx 0.15$$

38 / 59

## Count Lattice

Integers are easier than reals!

Create a large ensemble of cases so ratios of counts approximate the probabilities.



$$P(C|+, I) = \frac{9}{59} \approx 0.15$$

Of the 59 cases with positive test results, only 9 have the condition. The prevalence is so low that when there is a positive result, it's more likely to have been a mistake than accurate.

39 / 59

## Recap

### Bayesian inference is more than BT

Bayesian inference quantifies uncertainty by reporting probabilities for things we are uncertain of, given specified premises.

It uses *all* of probability theory, not just (or even primarily) Bayes's theorem.

### The Rules in Plain English

- Ground rule: Specify premises that include everything relevant that you know or are willing to presume to be true (for the sake of the argument!).
- BT: To adjust your appraisal when new evidence becomes available, add the evidence to your initial premises.
- LTP: If the premises allow multiple arguments for a hypothesis, its appraisal must account for all of them.

40 / 59

## Bayesian Fundamentals

- ① The big picture
- ② A flavor of Bayes:  $\chi^2$  confidence/credible regions
- ③ Foundations: Logic & probability theory
- ④ Probability theory for data analysis: Three theorems
- ⑤ Inference with parametric models
  - Parameter Estimation
  - Model Uncertainty

41 / 59

## Inference With Parametric Models

Models  $M_i$  ( $i = 1$  to  $N$ ), each with parameters  $\theta_i$ , each imply a *sampling dist'n* (conditional predictive dist'n for possible data):

$$p(D|\theta_i, M_i)$$

The  $\theta_i$  dependence when we fix attention on the **observed** data is the *likelihood function*:

$$\mathcal{L}_i(\theta_i) \equiv p(D_{\text{obs}}|\theta_i, M_i)$$

We may be uncertain about  $i$  (model uncertainty) or  $\theta_i$  (parameter uncertainty).

*Henceforth we will only consider the actually observed data, so we drop the cumbersome subscript:  $D = D_{\text{obs}}$ .*

42 / 59

## Three Classes of Problems

### Parameter Estimation

Premise = choice of model (pick specific  $i$ )  
 → What can we say about  $\theta_i$ ?

### Model Assessment

- Model comparison: Premise =  $\{M_i\}$   
 → What can we say about  $i$ ?
- Model adequacy/GoF: Premise =  $M_1 \vee$  “all” alternatives  
 → Is  $M_1$  adequate?

### Model Averaging

Models share some common params:  $\theta_i = \{\phi, \eta_i\}$   
 → What can we say about  $\phi$  w/o committing to one model?  
 (Examples: systematic error, prediction)

43 / 59

## Parameter Estimation

### Problem statement

$I$  = Model  $M$  with parameters  $\theta$  (+ any add'l info)  
 $H_i$  = statements about  $\theta$ ; e.g. “ $\theta \in [2.5, 3.5]$ ,” or “ $\theta > 0$ ”  
 Probability for any such statement can be found using a  
*probability density function* (PDF) for  $\theta$ :

$$\begin{aligned} P(\theta \in [\theta, \theta + d\theta] | \dots) &= f(\theta) d\theta \\ &= p(\theta | \dots) d\theta \end{aligned}$$

### Posterior probability density

$$p(\theta | D, M) = \frac{p(\theta | M) \mathcal{L}(\theta)}{\int d\theta p(\theta | M) \mathcal{L}(\theta)}$$

44 / 59

### Summaries of posterior

- “Best fit” values:
  - Mode,  $\hat{\theta}$ , maximizes  $p(\theta|D, M)$
  - Posterior mean,  $\langle \theta \rangle = \int d\theta \theta p(\theta|D, M)$
- Uncertainties:
  - Credible region  $\Delta$  of probability  $C$ :  
 $C = P(\theta \in \Delta|D, M) = \int_{\Delta} d\theta p(\theta|D, M)$   
 Highest Posterior Density (HPD) region has  $p(\theta|D, M)$  higher inside than outside
  - Posterior standard deviation, variance, covariances
- Marginal distributions
  - Interesting parameters  $\phi$ , nuisance parameters  $\eta$
  - Marginal dist'n for  $\phi$ :  $p(\phi|D, M) = \int d\eta p(\phi, \eta|D, M)$

45 / 59

## Nuisance Parameters and Marginalization

To model most data, we need to introduce parameters besides those of ultimate interest: *nuisance parameters*.

### Example

We have data from measuring a rate  $r = s + b$  that is a sum of an interesting signal  $s$  and a background  $b$ .

We have additional data just about  $b$ .

What do the data tell us about  $s$ ?

46 / 59

## Marginal posterior distribution

$$\begin{aligned}
 p(s|D, M) &= \int db p(s, b|D, M) \\
 &\propto p(s|M) \int db p(b|s) \mathcal{L}(s, b) \\
 &\equiv p(s|M) \mathcal{L}_m(s)
 \end{aligned}$$

with  $\mathcal{L}_m(s)$  the *marginal likelihood* for  $s$ . For broad prior,

$$\mathcal{L}_m(s) \approx p(\hat{b}_s|s) \mathcal{L}(s, \hat{b}_s) \delta b_s$$

best  $b$  given  $s$   
 $b$  uncertainty given  $s$

Profile likelihood  $\mathcal{L}_p(s) \equiv \mathcal{L}(s, \hat{b}_s)$  gets weighted by a [parameter space volume factor](#)

E.g., Gaussians:  $\hat{s} = \hat{r} - \hat{b}$ ,  $\sigma_s^2 = \sigma_r^2 + \sigma_b^2$

Background *subtraction* is a special case of background *marginalization*.

47 / 59

## Marginalization vs. Profiling

Marginal distribution for signal  $s$ , eliminating background  $b$ :

$$p(s|D, M) \propto p(s|M) \mathcal{L}_m(s)$$

with  $\mathcal{L}_m(s)$  the *marginal likelihood* for  $s$ ,

$$\mathcal{L}_m(s) \equiv \int db p(b|s) \mathcal{L}(s, b)$$

*For insight:* Suppose for a fixed  $s$ , we can accurately estimate  $b$  with max likelihood  $\hat{b}_s$ , with small uncertainty  $\delta b_s$ .

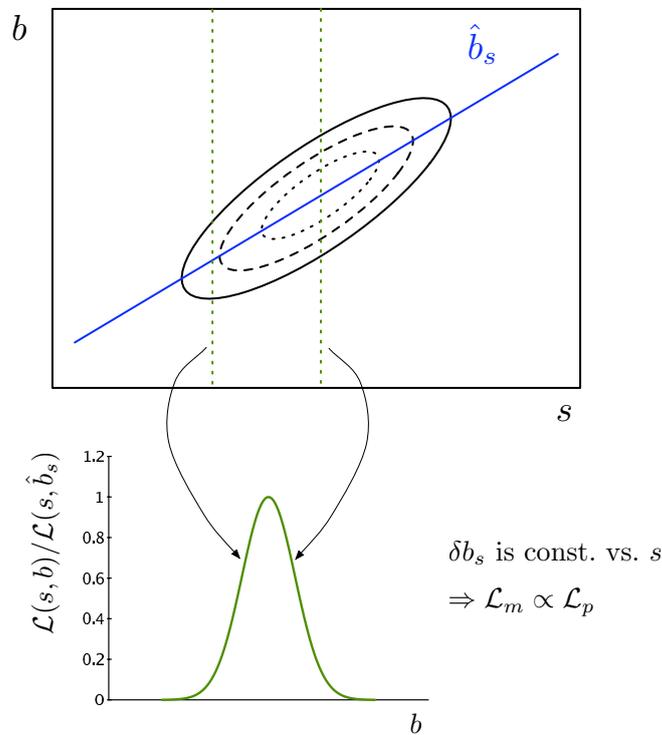
$$\begin{aligned}
 \mathcal{L}_m(s) &\equiv \int db p(b|s) \mathcal{L}(s, b) \\
 &\approx p(\hat{b}_s|s) \mathcal{L}(s, \hat{b}_s) \delta b_s
 \end{aligned}$$

best  $b$  given  $s$   
 $b$  uncertainty given  $s$

Profile likelihood  $\mathcal{L}_p(s) \equiv \mathcal{L}(s, \hat{b}_s)$  gets weighted by a [parameter space volume factor](#)

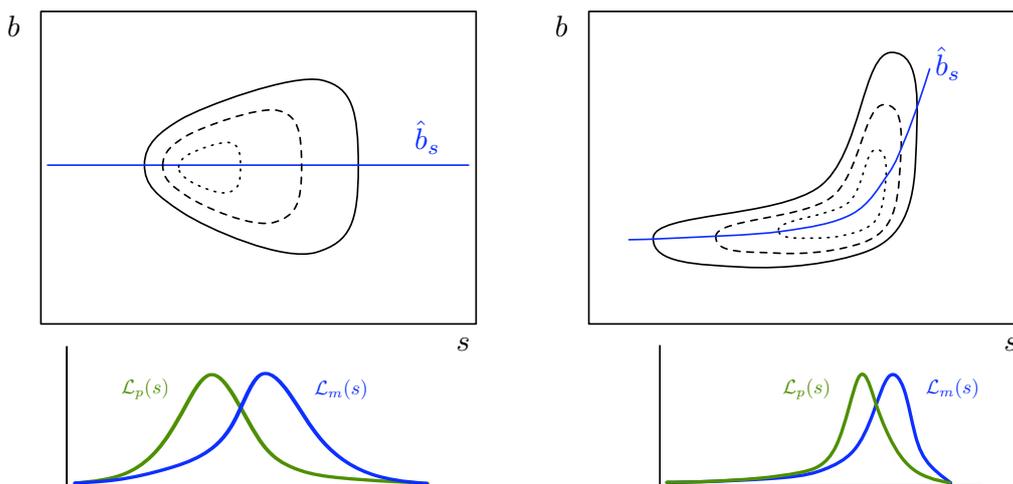
48 / 59

Bivariate normals:  $\mathcal{L}_m \propto \mathcal{L}_p$



49 / 59

Flared/skewed/banana-shaped:  $\mathcal{L}_m$  and  $\mathcal{L}_p$  differ



General result: For a linear (in params) model sampled with Gaussian noise, and flat priors,  $\mathcal{L}_m \propto \mathcal{L}_p$ . Otherwise, they will likely differ.

In *measurement error problems* (future lecture!) the difference can be dramatic.

50 / 59

## Many Roles for Marginalization

### Eliminate nuisance parameters

$$p(\phi|D, M) = \int d\eta p(\phi, \eta|D, M)$$

### Propagate uncertainty

Model has parameters  $\theta$ ; what can we infer about  $F = f(\theta)$ ?

$$\begin{aligned} p(F|D, M) &= \int d\theta p(F, \theta|D, M) = \int d\theta p(\theta|D, M) p(F|\theta, M) \\ &= \int d\theta p(\theta|D, M) \delta[F - f(\theta)] \quad [\text{single-valued case}] \end{aligned}$$

### Prediction

Given a model with parameters  $\theta$  and present data  $D$ , predict future data  $D'$  (e.g., for *experimental design*):

$$p(D'|D, M) = \int d\theta p(D', \theta|D, M) = \int d\theta p(\theta|D, M) p(D'|\theta, M)$$

### Model comparison...

51 / 59

## Model Comparison

### Problem statement

$I = (M_1 \vee M_2 \vee \dots)$  — Specify a set of models.

$H_i = M_i$  — Hypothesis chooses a model.

### Posterior probability for a model

$$\begin{aligned} p(M_i|D, I) &= p(M_i|I) \frac{p(D|M_i, I)}{p(D|I)} \\ &\propto p(M_i|I) \mathcal{L}(M_i) \end{aligned}$$

But  $\mathcal{L}(M_i) = p(D|M_i) = \int d\theta_i p(\theta_i|M_i) p(D|\theta_i, M_i)$ .

Likelihood for model = Average likelihood for its parameters

$$\mathcal{L}(M_i) = \langle \mathcal{L}(\theta_i) \rangle$$

Varied terminology: Prior predictive = Average likelihood = Global likelihood = Marginal likelihood = (Weight of) Evidence for model

## Odds and Bayes factors

A ratio of probabilities for two propositions using the same premises is called the *odds* favoring one over the other:

$$\begin{aligned} O_{ij} &\equiv \frac{p(M_i|D, I)}{p(M_j|D, I)} \\ &= \frac{p(M_i|I)}{p(M_j|I)} \times \frac{p(D|M_i, I)}{p(D|M_j, I)} \end{aligned}$$

The data-dependent part is called the *Bayes factor*:

$$B_{ij} \equiv \frac{p(D|M_i, I)}{p(D|M_j, I)}$$

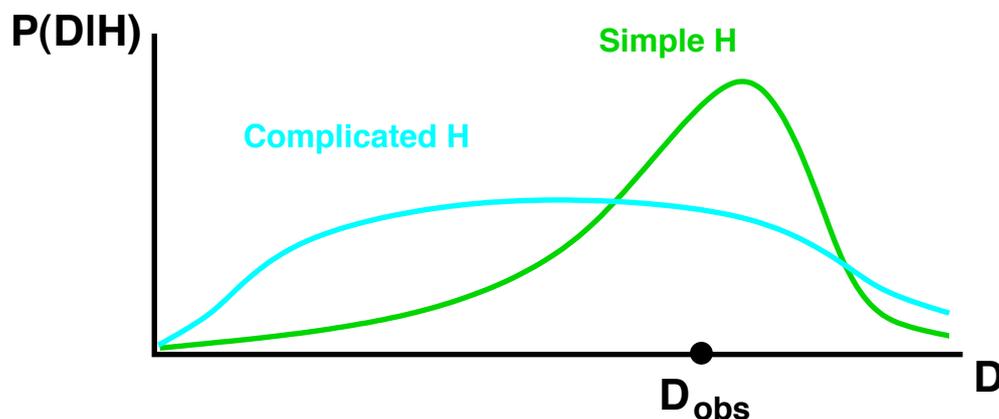
It is a *likelihood ratio*; the BF terminology is usually reserved for cases when the likelihoods are marginal/average likelihoods.

53 / 59

## An Automatic Occam's Razor

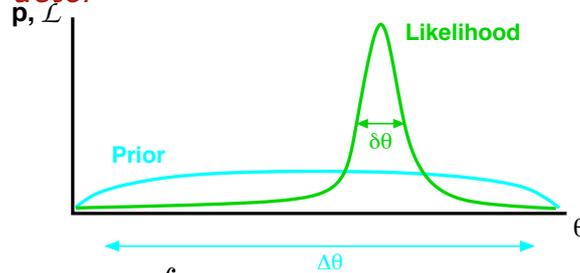
*Predictive probabilities can favor simpler models*

$$p(D|M_i) = \int d\theta_i p(\theta_i|M) \mathcal{L}(\theta_i)$$



54 / 59

### The Occam Factor



$$\begin{aligned}
 p(D|M_i) &= \int d\theta_i p(\theta_i|M) \mathcal{L}(\theta_i) \approx p(\hat{\theta}_i|M) \mathcal{L}(\hat{\theta}_i) \delta\theta_i \\
 &\approx \mathcal{L}(\hat{\theta}_i) \frac{\delta\theta_i}{\Delta\theta_i} \\
 &= \text{Maximum Likelihood} \times \text{Occam Factor}
 \end{aligned}$$

Models with more parameters often make the data more probable — *for the best fit*

Occam factor penalizes models for “wasted” **volume of parameter space**

Quantifies intuition that models shouldn't require fine-tuning

55 / 59

## Model Averaging

### Problem statement

$I = (M_1 \vee M_2 \vee \dots)$  — Specify a set of models

Models all share a set of “interesting” parameters,  $\phi$

Each has different set of nuisance parameters  $\eta_i$  (or different prior info about them)

$H_i$  = statements about  $\phi$

### Model averaging

Calculate posterior PDF for  $\phi$ :

$$\begin{aligned}
 p(\phi|D, I) &= \sum_i p(M_i|D, I) p(\phi|D, M_i) \\
 &\propto \sum_i \mathcal{L}(M_i) \int d\eta_i p(\phi, \eta_i|D, M_i)
 \end{aligned}$$

The model choice is a (discrete) nuisance parameter here.

56 / 59

## Theme: Parameter Space Volume

*Bayesian calculations sum/integrate over parameter/hypothesis space!*

(Frequentist calculations average over *sample* space & typically *optimize* over parameter space.)

- Marginalization weights the profile likelihood by a volume factor for the nuisance parameters.
- Model likelihoods have Occam factors resulting from parameter space volume factors.

Many virtues of Bayesian methods can be attributed to this accounting for the “size” of parameter space. This idea does not arise naturally in frequentist statistics (but it can be added “by hand”).

57 / 59

## Roles of the Prior

*Prior has two roles*

- Incorporate any relevant prior information
- Convert likelihood from “intensity” to “measure”  
→ Accounts for *size of hypothesis space*

*Physical analogy*

$$\text{Heat: } Q = \int dV c_v(\mathbf{r}) T(\mathbf{r})$$

$$\text{Probability: } P \propto \int d\theta p(\theta|I) \mathcal{L}(\theta)$$

Maximum likelihood focuses on the “hottest” hypotheses.

Bayes focuses on the hypotheses with the most “heat.”

A high- $T$  region may contain little heat if its  $c_v$  is low or if its volume is small.

A high- $\mathcal{L}$  region may contain little probability if its prior is low or if its volume is small.

58 / 59

## Recap of Key Ideas

- Probability as generalized logic for appraising arguments
- Three theorems: BT, LTP, Normalization
- Calculations characterized by parameter space integrals
  - Credible regions, posterior expectations
  - Marginalization over nuisance parameters
  - Occam's razor via marginal likelihoods
  - Do not integrate/average over hypothetical data



## Chapter 15

# JACKKNIFE & BOOTSTRAP

*Notes by Jogesh Babu*

## Jackknife and Bootstrap

G. Jogesh Babu

### 1 Introduction

The classical statistical methods of earlier sections concentrated mainly on the statistical properties of the estimators that have a simple closed form and which can be analyzed mathematically. Except for a few important but simple statistics, these methods involve often unrealistic model assumptions. It is often relatively simple to devise a statistic that measures the property of interest, but is almost always difficult or impossible to determine the distribution of that statistic. These limitations have been overcome in the last two decades of the 20th Century with advances in electronic computers. A class of computationally intensive procedures known as *resampling methods* provide inference on a wide range of statistics under very general conditions. Resampling methods involve constructing hypothetical ‘populations’ derived from the observations, each of which can be analyzed in the same way to see how the statistics depend on plausible random variations in the observations. Resampling the original data preserves whatever distributions are truly present, including selection effects such as truncation and censoring.

Perhaps the *half-sample method* is the oldest resampling method, where one repeatedly chooses at random half of the data points, and estimates the statistic for each resample. The inference on the parameter can be based on the histogram of the resampled statistics. It was used by Mahalanobis in 1946 under the name *interpenetrating samples*. An important variant is the Quenouille–Tukey *jackknife method*. For a dataset with  $n$  data points, one constructs exactly  $n$  hypothetical datasets each with  $n - 1$  points, each one omitting a different point. The most important of resampling methods is called the *bootstrap*. Bradley Efron introduced the bootstrap method, also known as resampling with replacement, in 1979. Here one generates a large number of datasets, each with  $n$  data points randomly drawn from the original data. The constraint is that each drawing is made from the entire dataset, so a simulated dataset is likely to miss some points and have duplicates or triplicates of others. Thus, bootstrap can be viewed as a *Monte Carlo method* to simulate from an existing data, without any assumption on the underlying population.

### 2 Jackknife

Jackknife method was introduced by Quenouille (1949) to estimate the bias of an estimator. The method is later shown to be useful in reducing the bias as well as in estimating the variance of an estimator. Let  $\hat{\theta}_n$  be an estimator of  $\theta$  based on  $n$  i.i.d. random vectors  $X_1, \dots, X_n$ , i.e.,  $\hat{\theta}_n = f_n(X_1, \dots, X_n)$ , for some function  $f_n$ . Let

$$\hat{\theta}_{n,-i} = f_{n-1}(X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n)$$

be the corresponding recomputed statistic based on all but the  $i$ -th observation. The jackknife estimator of bias  $E(\hat{\theta}_n) - \theta$  is given by

$$bias_J = \frac{(n-1)}{n} \sum_{i=1}^n (\hat{\theta}_{n,-i} - \hat{\theta}_n). \quad (1)$$

Jackknife estimator  $\theta_J$  of  $\theta$  is given by

$$\theta_J = \hat{\theta}_n - bias_J = \frac{1}{n} \sum_{i=1}^n (n\hat{\theta}_n - (n-1)\hat{\theta}_{n,-i}). \quad (2)$$

Such a bias corrected estimator hopefully reduces the over all bias. The summands above

$$\theta_{n,i} = n\hat{\theta}_n - (n-1)\hat{\theta}_{n,-i}, \quad i = 1, \dots, n$$

are called *pseudo-values*.

## 2.1 Bias Reduction

Jackknifing, indeed, helps in reducing bias of an estimator in many cases. Suppose the expected value of the estimator  $\hat{\theta}_n$  is of the form

$$E(\hat{\theta}_n) = \theta + \frac{a}{n} + \frac{b}{n^2},$$

then clearly,

$$E(\hat{\theta}_{n,i}) = \theta - \frac{b}{n(n-1)}.$$

Consequently, the bias of the jackknife estimator is  $E(\theta_J) - \theta = O(n^{-2})$ , which is of lower order than the bias of  $\hat{\theta}_n$ .

## 2.2 Estimation of variance

In the case of the sample mean  $\hat{\theta}_n = \bar{X}_n$ , it is easy to check that the *pseudo-values* are simply,

$$\theta_{n,i} = n\hat{\theta}_n - (n-1)\hat{\theta}_{n,-i} = X_i, \quad i = 1, \dots, n.$$

This provides motivation for the jackknife estimator of variance of  $\hat{\theta}_n$ ,

$$\begin{aligned} var_J(\hat{\theta}_n) &= \frac{1}{n(n-1)} \sum_{i=1}^n (\theta_{n,i} - \theta_J)(\theta_{n,i} - \theta_J)' \\ &= \frac{n-1}{n} \sum_{i=1}^n (\hat{\theta}_{n,-i} - \bar{\theta}_n)(\hat{\theta}_{n,-i} - \bar{\theta}_n)', \end{aligned} \quad (3)$$

where  $\bar{\theta}_n = \frac{1}{n} \sum_{i=1}^n \hat{\theta}_{n,-i}$ . For most statistics, jackknife estimator of variance is consistent, i.e.,

$$Var_J(\hat{\theta}_n)/Var(\hat{\theta}_n) \rightarrow 1,$$

as  $n \rightarrow \infty$  almost surely. In particular, this holds for a **smooth functional model**. To describe this, let the statistic of interest  $\hat{\theta}_n$  based on  $n$  data points be defined by  $H(\bar{Z}_n)$ , where  $\bar{Z}_n$  is the sample mean of random vectors  $Z_1, \dots, Z_n$  and  $H$  is continuously differentiable in a neighborhood of  $E(\bar{Z}_n)$ . Many commonly occurring statistics, including: Sample Means, Sample Variances, Central and Non-central t-statistics (with possibly non-normal populations), Sample Coefficient of Variation, Maximum Likelihood Estimators, Least Squares Estimators, Correlation Coefficients, Regression Coefficients, Smooth transforms of these statistics, fall under this model.

However, consistency does not always hold; for example the jackknife method fails for non-smooth statistics, such as the sample median. If  $\hat{\theta}_n$  denotes the sample median in the univariate case, then in general,

$$\text{Var}_J(\hat{\theta}_n)/\text{Var}(\hat{\theta}_n) \rightarrow \left(\frac{1}{2}\chi_2^2\right)^2$$

in distribution, where  $\chi_2^2$  denotes a *chi-square* random variable with 2 degrees of freedom (see Efron 1982, §3.4). So in this case, the jackknife method does not lead to a consistent estimator of the variance. However, a resampling method called *bootstrap* discussed in the next section, would lead to a consistent estimator.

### 3 Bootstrap

Bootstrap resampling constructs datasets with  $n$  points (rather than  $n - 1$  for the jackknife) where each point was selected from the full dataset; that is, resampling with replacement. The importance of the bootstrap emerged during the 1980s when mathematical study demonstrated that it gives nearly optimal estimate of the distribution of many statistics under a wide range of circumstances. In several cases, the method yields better results than those obtained by the classical normal approximation theory. However, one should caution that bootstrap is not the solution for all problems. The theory developed in 1980s and 1990s, show that bootstrap fails in some ‘non-smooth’ situations. Hence, caution should be used and should resist the temptation to use the method inappropriately. Many of these methods work well in the case of **smooth functional model**. As described earlier, these estimators are smooth functions of sample mean of a random vectors. In view of this, the bootstrap method is first described here for special case of the sample mean.

#### 3.1 Description of the bootstrap method

Let  $\mathbf{X} = (X_1, \dots, X_n)$  be data drawn from an unknown population distribution  $F$ . Suppose  $\hat{\theta}_n$ , based on data  $\mathbf{X}$ , is a good estimator of  $\theta$ , a parameter of interest. The interest lies in assessing its accuracy in estimation. Determining the confidence intervals for  $\theta$  requires knowledge of the sampling distribution  $G_n$  of  $\hat{\theta}_n - \theta$ , *i.e.*  $G_n(x) = P(\hat{\theta}_n - \theta \leq x)$ , for all  $x$ .

For example, the sample mean  $\bar{X}_n = n^{-1} \sum_{i=1}^n X_i$  is a good estimator of the population mean  $\mu$ . To get the confidence interval for  $\mu$ , we must find the sampling distribution of  $\bar{X}_n - \mu$ , which depends on the shape and other characteristics of the unknown distribution  $F$ .



Table 1: Statistics and their bootstrap versions

Statistic	Bootstrap Version
Mean, $\bar{X}_n$	$\bar{X}_n^*$
Variance, $\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X}_n)^2$	$\frac{1}{n} \sum_{i=1}^n (X_i^* - \bar{X}_n^*)^2$
Ratio estimator, $\bar{X}_n/\bar{Y}_n$	$\bar{X}_n^*/\bar{Y}_n^*$
Correlation coefficient, $\frac{\sum_{i=1}^n (X_i Y_i - \bar{X}_n \bar{Y}_n)}{\sqrt{(\sum_{i=1}^n (X_i - \bar{X}_n)^2)(\sum_{i=1}^n (Y_i - \bar{Y}_n)^2)}}$	$\frac{\sum_{i=1}^n (X_i^* Y_i^* - \bar{X}_n^* \bar{Y}_n^*)}{\sqrt{(\sum_{i=1}^n (X_i^* - \bar{X}_n^*)^2)(\sum_{i=1}^n (Y_i^* - \bar{Y}_n^*)^2)}}$

distribution of the estimator for the original dataset is obtained from the histogram of the estimators obtained from the bootstrapped samples.

The most popular and simple bootstrap is the *nonparametric bootstrap*, where the re-sampling with replacement is based on the EDF of the original data. This gives equal weights to each of the original data points. Table 1 gives bootstrap versions of some commonly used statistics. In the case of ratio estimator and the correlation coefficient, the data pairs are resampled from the original data pairs  $(X_i, Y_i)$ .

### 3.2 Confidence intervals

Bootstrap resampling is also widely used deriving confidence intervals for parameters. However, unless the limiting distribution of the point estimator is free from the unknown parameters, one can not invert it to get confidence intervals. Such quantities, with distributions that are free from unknown parameters, are called ‘pivotal’ statistics. It is thus important to focus on pivotal or approximately pivotal quantities in order to get reliable confidence intervals for the parameter of interest. For example, if  $X_i \sim N(\mu, \sigma^2)$ , then  $\sqrt{n}(\bar{X} - \mu)/s_n$  has  $t$  distribution with  $n - 1$  degrees of freedom, and hence it is pivotal, where  $s_n^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$ . In non-normal case, it is approximately pivotal. To obtain bootstrap confidence interval for  $\mu$ , we compute  $\sqrt{n}(\bar{X}^{*(j)} - \bar{X})/s_n$  for  $N$  bootstrap samples, arrange the values in increasing order

$$h_1 < h_2 < \dots < h_N.$$

One can then read off from the histogram (say) the 90% confidence interval of the parameter. That is, the 90% confidence interval for  $\mu$  is given by

$$\bar{X} - h_m \frac{s_n}{\sqrt{n}} \leq \mu < \bar{X} - h_k \frac{s_n}{\sqrt{n}},$$

where  $k = [0.05N]$  and  $m = [0.95N]$ . Babu & Singh (1983) have shown that  $N \sim n(\log n)^2$  bootstrap iterations would be sufficient.

It is important to note that even when  $\sigma$  is known the bootstrap version of  $\sqrt{n}(\bar{X} - \mu)/\sigma$  is  $\sqrt{n}(\bar{X}^* - \bar{X})/s_n$ . One should not replace  $\sqrt{n}(\bar{X}^* - \bar{X})/s_n$  by  $\sqrt{n}(\bar{X}^* - \bar{X})/\sigma$ .

### 3.3 Bootstrap at its best: Smooth function model

It is well established using Edgeworth expansions that the bootstrap provides a good approximation for a ‘Studentized smooth functional model’. A broad class of commonly used statistics, including least squares estimators and some maximum likelihood estimators, can be expressed as smooth function of multivariate means. The model is illustrated using Pearson’s well known estimator  $\hat{\rho}_n$  of correlation coefficient  $\rho$ . The sample correlation coefficient  $\hat{\rho}_n$  based on the data  $(X_i, Y_i), i = 1, \dots, n$ , can be expressed as  $\hat{\rho}_n = H(\bar{\mathbf{Z}}_n)$ , and  $\rho^* = H(\bar{\mathbf{Z}}_n^*)$ , where

$$\mathbf{Z}_i = (X_i Y_i, X_i^2, Y_i^2, X_i, Y_i), \quad \mathbf{Z}_i^* = (X_i^* Y_i^*, X_i^{*2}, Y_i^{*2}, X_i^*, Y_i^*)$$

and

$$H(a_1, a_2, a_3, a_4, a_5) = \frac{(a_1 - a_4 a_5)}{\sqrt{((a_2 - a_4^2)(a_3 - a_5^2))}}.$$

Note that  $H$  is a differentiable function.

In general, if the standard deviation of  $T_n(\mathbf{X}; F)$  is not known (which is often the case), the function may be divided by a good estimator of the standard deviation of the statistic. This makes it an ‘approximate pivotal’ quantity. Such a correction by a special type of estimator of standard deviation for the smooth function model refers to Studentization, as it is similar to the Student’s  $t$ -statistic. The empirical distribution of the data is used to estimate the standard deviation of the statistic in a special way, making it an ‘approximate pivotal’ quantity. For the smooth function model, a good estimator of the variance of  $\sqrt{n}H(\bar{\mathbf{Z}}_n)$  is given by  $\hat{\sigma}^2 = \ell^T(\bar{\mathbf{Z}}_n)\Sigma_n\ell(\bar{\mathbf{Z}}_n)$ , where  $\ell(\mathbf{x})$  denotes the vector of first order partial derivatives of  $H$  at  $\mathbf{x}$ ,  $^T$  denotes transpose, and  $\Sigma_n$  denotes the variance-covariance matrix computed from the  $\{\mathbf{Z}_i\}$ . That is,

$$\Sigma_n = \frac{1}{n} \sum_{i=1}^n (\mathbf{Z}_i - \bar{\mathbf{Z}}_n)(\mathbf{Z}_i - \bar{\mathbf{Z}}_n)^T. \quad (4)$$

This leads to Studentization or approximate pivotal function

$$t_n = \sqrt{n}(H(\bar{\mathbf{Z}}_n) - H(\mathbf{E}(\mathbf{Z}_1)))/\hat{\sigma} \quad (5)$$

Its bootstrap version is

$$t_n^* = \sqrt{n}(H(\bar{\mathbf{Z}}_n^*) - H(\bar{\mathbf{Z}}_n))/\sqrt{\ell^T(\bar{\mathbf{Z}}_n^*)\Sigma_n^*\ell(\bar{\mathbf{Z}}_n^*)}, \quad (6)$$

where  $\Sigma_n^*$  denotes the variance-covariance matrix computed from the bootstrap sample  $\{\mathbf{Z}_i^*\}$ , *i.e.*

$$\Sigma_n^* = \frac{1}{n} \sum_{i=1}^n (\mathbf{Z}_i^* - \bar{\mathbf{Z}}_n^*)(\mathbf{Z}_i^* - \bar{\mathbf{Z}}_n^*)^T. \quad (7)$$

If  $H(\bar{\mathbf{Z}}_n)$  represents the sample mean  $\bar{X}_n$ , then  $\hat{\sigma}^2 = s_n^2$ , and if  $H(\bar{\mathbf{Z}}_n)$  represents the ratio statistic  $\hat{\theta} = \bar{X}_n/\bar{Y}_n$ , then  $\hat{\sigma}^2 = \bar{Y}^{-2}n^{-1} \sum_{i=1}^n (X_i - \hat{\theta}Y_i)^2$ .

Under very general conditions, if  $\ell(\mathbf{E}(\mathbf{Z}_1)) \neq 0$ , then the approximation of the sampling distribution of  $t_n$  by the bootstrap distribution (the distribution of  $t_n^*$ ) is better than the

classical normal approximation. This is mainly because the bootstrap automatically corrects for the skewness factor. This is established using Edgeworth expansion (see Babu & Singh (1983), and Babu & Singh (1984)):

$$P(t_n \leq x) = \Phi(x) + \frac{1}{\sqrt{n}}p(x)\phi(x) + \text{error}$$

$$P^*(t_n^* \leq x) = \Phi(x) + \frac{1}{\sqrt{n}}p_n(x)\phi(x) + \text{error}.$$

The ‘error’ terms are so small that

$$\sqrt{n}|P(t_n \leq x) - P^*(t_n^* \leq x)| \rightarrow 0.$$

The theory above is applicable in very general set up that includes the statistics: *Sample Means, Sample Variances, Central and Non-central t-statistics (with possibly non-normal populations), Sample Coefficient of Variation, Maximum Likelihood Estimators, Least Squares Estimators, Correlation Coefficients, Regression Coefficients*, and Smooth transforms of these statistics.

Thus the sampling distribution of several commonly occurring statistics are closer to the corresponding bootstrap distribution than the normal distribution. These conditional approximations are suggestive of the unconditional ones, though one cannot be derived from the other by elementary methods. Babu & Bose (1988) provide theoretical justification for the accuracy of the bootstrap confidence intervals both in terms of the actual coverage probability achieved and also the limits of the confidence interval.

In spite of these positive results, one should use caution in using bootstrap methods. It is not a ‘cure all’ solution. There are cases where bootstrap method fails. These include, non-smooth statistics such as  $\hat{\theta} = \max_{1 \leq i \leq n} X_i$  (see Bickel & Freedman (1981)), heavy tailed distributions,  $\hat{\theta} = \bar{X}$  and  $EX_1^2 = \infty$  (see Babu (1984) and Athreya (1987)), and asymptotically non-linear statistics such as,  $\hat{\theta} - \theta = H(\bar{\mathbf{Z}}_n) - H(E(\mathbf{Z}_1))$  when  $\partial H(E(\mathbf{Z}_1)) = 0$ . In the last case the limiting distribution is like that of linear combinations of Chi-squares, but here a modified version works (Babu (1984)).

### 3.4 Linear regression

Consider the simple linear regression model, where the data  $(X_1, Y_1), \dots, (X_n, Y_n)$  satisfies

$$Y_i = \alpha + \beta X_i + e_i, \quad (8)$$

where  $\alpha$  and  $\beta$  are unknown parameters,  $X_1, \dots, X_n$  are often called the design points. The error variables  $e_i$  need not be Gaussian, but are assumed to be independent with zero mean and standard deviation  $\sigma_i$ . This model is called homoscedastic if all the  $\sigma_i$  are identical. Otherwise, the model is known as heteroscedastic. In what follows, for any sequence of pairs  $\{(U_1, V_1), \dots, (U_n, V_n)\}$  of numbers, we use the notation

$$S_{UV} = \sum_{i=1}^n (U_i - \bar{U}_n)(V_i - \bar{V}_n) \quad \text{and} \quad \bar{U}_n = \frac{1}{n} \sum_{i=1}^n U_i. \quad (9)$$

There are two conceptually separate models to consider, random and fixed design models. In the first case, the pairs  $\{(X_1, Y_1), \dots, (X_n, Y_n)\}$  are assumed to be random data points and the conditional mean and variance of  $e_i$  given  $X_i$  are assumed to be zero and  $\sigma_i^2$ . In the latter case,  $X_1, \dots, X_n$  are assumed to be fixed numbers (fixed design). In both the cases, the least squares estimators  $\hat{\alpha}$  and  $\hat{\beta}$  of  $\alpha$  and  $\beta$  are given by

$$\hat{\beta} = S_{XY}/S_{XX} \quad \text{and} \quad \hat{\alpha} = \bar{Y}_n - \hat{\beta}\bar{X}_n. \quad (10)$$

However the variances of these estimators are different for a random and fixed designs, though the difference is very small for large  $n$ . We shall concentrate on the fixed design case here.

The variance of the slope  $\hat{\beta}$  is given by

$$\text{var}(\hat{\beta}) = \sum_{i=1}^n (X_i - \bar{X}_n)^2 \sigma_i^2 / S_{XX}^2, \quad (11)$$

and depends on the individual error deviations  $\sigma_i$ , which may or may not be known. Knowledge of  $\text{var}(\hat{\beta})$  provides the confidence intervals for  $\beta$ . Several resampling methods are available in the literature to estimate the sampling distribution and  $\text{var}(\hat{\beta})$ . We consider three bootstrap procedures: a) the classical bootstrap, b) the paired bootstrap.

#### *The classical bootstrap*

Let  $\hat{e}_i$  denote the residual of the  $i$ -th element of  $\hat{e}_i = Y_i - \hat{\alpha} - \hat{\beta}X_i$  and define  $\tilde{e}_i$  to be

$$\tilde{e}_i = \hat{e}_i - \frac{1}{n} \sum_{j=1}^n \hat{e}_j. \quad (12)$$

A bootstrap sample is obtained by randomly drawing  $e_1^*, \dots, e_n^*$  with replacement from  $\tilde{e}_1, \dots, \tilde{e}_n$ . The bootstrap estimators  $\beta^*$  and  $\alpha^*$  of the slope and the intercept are given by

$$\beta^* - \hat{\beta} = S_{Xe^*}/S_{XX} \quad \text{and} \quad \alpha^* - \hat{\alpha} = (\hat{\beta} - \beta^*)\bar{X}_n + \bar{e}_n^*. \quad (13)$$

To estimate the sampling distribution and variance, the procedure is repeated  $N$  times to obtain

$$\beta_1^*, \dots, \beta_N^* \quad \text{where} \quad N \sim n(\log n)^2. \quad (14)$$

The histogram of these  $\beta^*$ s give a good approximation to the sampling distribution of  $\hat{\beta}$  and the estimate of the variance  $\hat{\beta}$  is given by

$$\text{var}_{\text{Boot}} = \frac{1}{N} \sum_{j=1}^N (\beta_j^* - \hat{\beta})^2. \quad (15)$$

This variance estimator is the best among the two methods proposed here, if the residuals are homoscedastic; *i.e.* if the variances of the residuals  $E(\epsilon_i^2) = \sigma_i^2 = \sigma^2$  are all the same. However if they are not, then the bootstrap estimator of the variance is an inconsistent estimator, and does not approach the actual variance. The *paired bootstrap* is robust against

heteroscedasticity, giving consistent estimator of variance when the residuals have different standard deviations.

*The paired bootstrap*

The paired bootstrap are useful to handle heteroscedastic data. The paired bootstrap method treats the design points as random quantities. A simple random sample  $(\tilde{X}_1, \tilde{Y}_1), \dots, (\tilde{X}_n, \tilde{Y}_n)$  is drawn from  $(X_1, Y_1), \dots, (X_n, Y_n)$  and the paired bootstrap estimators of slope and intercept are constructed as

$$\tilde{\beta} = \frac{\sum_{i=1}^n (\tilde{X}_i - \bar{\tilde{X}})(\tilde{Y}_i - \bar{\tilde{Y}})}{\sum_{i=1}^n (\tilde{X}_i - \bar{\tilde{X}})^2}, \quad \text{and} \quad \tilde{\alpha} = \bar{\tilde{Y}} - \tilde{\beta} \bar{\tilde{X}}$$

The variance is obtained by repeating the resampling scheme  $N$  times and applying equation (15).

Figure 1 provides a simple FORTRAN code for jackknife and paired bootstrap resampling.

```

C      PAIRED BOOTSTRAP RESAMPLING
      NSIM = INT(N * ALOG(FLOAT(N))**2)
      DO 20 ISIM = 1, NSIM
      DO 10 I = 1, N
          J = INT(RANDOM * N + 1.0)
          XBOOT(I) = X(J)
10      YBOOT(I) = Y(J)
20      CONTINUE

C      JACKKNIFE RESAMPLING
      DO 40 NSIM = 1, N
      DO 30 I = 1, N-1
          IF(I.LT.NSIM)
              XJACK(I) = X(I)
              YJACK(I) = Y(I)
          ELSE
              XJACK(I) = X(I+1)
              YJACK(I) = Y(I+1)
          ENDELSE
30      CONTINUE
40      CONTINUE

```

Figure 1: FORTRAN code illustrating the paired bootstrap and jackknife resampling for a two dimensional dataset  $(x_i, y_i), i = 1, \dots, N$ .

The bootstrap methodology, mathematics and second order properties are reviewed in Babu & Rao (1993). A detailed account of second order asymptotics can be found in Hall

(1992). A less mathematical overview of the bootstrap is presented in Efron and Tibshirani (1993). The book by Zoubir & Iskander (2004) serves as a handbook on ‘bootstrap’ for engineers, to analyze complicated data with little or no model assumptions. Bootstrap has found many applications in engineering field including, artificial neural networks, biomedical engineering, environmental engineering, image processing, and Radar and sonar signal processing. Majority of the applications in the book are taken from signal processing literature.

## References

- [1] Athreya, K. B. (1987). Bootstrap of the mean in the infinite variance case. *Ann. Statist.*, **15**, no. 2, 724-731.
- [2] Babu, G. J. (1984). Bootstrapping statistics with linear combinations of chi-squares as weak limit. *Sankhya*, Series A, **46**, 85-93.
- [3] Babu, G. J. (1986). A note on bootstrapping the variance of the sample quantile. *Ann. Inst. Statist. Math.*, **38**, Part A, 439-443.
- [4] Babu, G. J., and Bose, A. (1988). Bootstrap confidence intervals. *Statistics and Probability Letters*, **7**, 151-160.
- [5] Babu, G. J., and Rao, C. R. (1993). Bootstrap methodology. In *Handbook of Statistics, Vol. 9* “Computational Statistics.” C. R. Rao (Ed.), Elsevier Science Publishers B. V., Amsterdam, 627-659.
- [6] Babu, G. J., and Singh, K. (1983). Inference on means using the bootstrap. *Annals of Statistics*, **11**, 999-1003.
- [7] Babu, G. J., and Singh, K. (1984a). On one term Edgeworth correction by Efron’s Bootstrap. *Sankhya*, Series A, **46**, 219-232.
- [8] Babu, G. J., and Singh, K. (1984b). Asymptotic representations related to jackknifing and bootstrapping L-statistics. *Sankhyā*, Series A, **46**, 195-206.
- [9] Bickel, P. J., and Freedman, D. A. Some asymptotic theory for the bootstrap. *Ann. Statist.*, **9**, no. 6, 1196–1217.
- [10] Chernick, M. R. (2007). *Bootstrap Methods - A guide for Practitioners and Researchers*, (2nd Ed.) Wiley Inter-Science.
- [11] Chernick, M. R., and LaBudde, R. A. (2011). *An Introduction to Bootstrap Methods with Applications to R*, Wiley.
- [12] Efron, B. (1979). Bootstrap methods: another look at the jackknife. *Ann. Statist.*, **7**, no. 1, 1-26.
- [13] Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans*. CBMS-NSF Regional Conference Series in Applied Mathematics, 38. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa.
- [14] Efron, B., and Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Monographs on Statistics and Applied Probability, 57. Chapman and Hall, New York.
- [15] Hall, P. (1992). *The bootstrap and Edgeworth expansion*. Springer Series in Statistics. Springer-Verlag, New York.

- [16] Quenouille, M. (1949). Approximate tests of correlation in time series. *J. Roy. Statist. Soc., Ser. B*, **11**, pp. 1884.
- [17] Zoubir, A. M., and Iskander, D. R. (2004). *Bootstrap Techniques for Signal Processing*. Cambridge University Press, Cambridge, U.K.

## Chapter 16

# MODEL SELECTION & EVALUATION, GOODNESS-OF-FIT

*Notes by Rajeeva Karandikar & Jogesh Babu*

# Model Selection and Goodness of Fit

G. Jogesh Babu  
Center for Astrostatistics  
The Pennsylvania State University

## 1 Introduction

The aim of model fitting is to provide most parsimonious ‘best’ fit of a parametric model to data. It might be a simple, heuristic model to phenomenological relationships between observed properties in a sample of astronomical objects. Examples include characterizing the Fundamental Plane of elliptical galaxies or the power law index of solar flare energies. Perhaps more important are complex nonlinear models based on our astrophysical understanding of the observed phenomenon. Here, if the model family truly represents the underlying phenomenon, the fitted parameters give insights into sizes, masses, compositions, temperatures, geometries, and evolution of astronomical objects. Examples of astrophysical modeling include:

- Interpreting the spectrum of an accreting black hole such as a quasar. Is it a nonthermal power law, a sum of featureless blackbodies, and/or a thermal gas with atomic emission and absorption lines?
- Interpreting the radial velocity variations of a large sample of solar-like stars. This can lead to discovery of orbiting systems such as binary stars and exoplanets, giving insights into star and planet formation.
- Interpreting the spatial fluctuations in the cosmic microwave background radiation. What are the best fit combinations of baryonic, Dark Matter and Dark Energy components? Are Big Bang models with quintessence or cosmic strings excluded?

The mathematical procedures used to link data with astrophysical models fall into the realm of statistics. The relevant methods fall under the rubrics of statistical model selection, regression, and goodness-of-fit. Astronomers’ understanding of such methods are often rather simplistic, and we seek here to develop increased sophistication in some aspects of the methodological issues. We discuss the advantages and limitations of some traditional model fitting methods and suggest new procedures when these methods are inadequate. In particular, we discuss some recently developed procedures based on nonparametric resampling designed for model selection and goodness-of-fit when the astronomer not only seeks the best parameters of the model, but wishes to consider entirely different families of parametric models.

## 2 Challenges of Model Selection and Fitting

Consider the astronomical spectrum illustrated in Figure 1a where flux from a source is plotted against energy of light received by an X-ray telescope. Here the photons are shown collected into constant-width bins, and the measured flux value  $F$  is accompanied by an

error bar  $\sigma$  representing the uncertainty of the intensity at each energy based on the square-root of the number of counts in the bin. The dataset shown happens to be a low-resolution spectrum from the *Chandra* Orion Ultradeep Project (COUP) where NASA's *Chandra* X-ray Observatory observed about 1400 pre-main sequence stars in the Orion Nebula region for 13 days (Getman *et al.* 2005). But it could easily be an optical spectrum of a high-redshift starburst galaxy, or a millimeter spectrum of a collapsing molecular cloud core, or the spectrum of a gamma-ray burst at the birth of a black hole.

The histogram in Figure 1a shows the best-fit astrophysical model assuming a plausible emission mechanism: a single-temperature thermal plasma with solar abundances of elements. This model  $M$  has three free parameters – plasma temperature, line-of-sight absorption, and normalization – which we denote by the vector  $\theta$ . The astrophysical model has been convolved with complicated functions representing the sensitivity of the telescope and detector. The model is fitted by minimizing chi-square with an iterative procedure. That is

$$\hat{\theta} = \arg \min_{\theta} \chi^2(\theta) = \arg \min_{\theta} \sum_{i=1}^N \left( \frac{y_i - M_i(\theta)}{\sigma_i} \right)^2.$$

*Chi-square minimization* is a misnomer. It is known as parameter estimation by *weighted least squares*. Confidence intervals on best-fit parameter values are obtained using a  $\chi^2_{min}$ -plus-constant criterion. These procedures are familiar in the astronomical community (*e.g.* Bevington 1969).

There are important limitations to  $\chi^2$  minimization for use in astronomical model selection and fitting. The procedure depends strongly on Gaussian assumptions. It fails when the errors are non-Gaussian (*e.g.* small- $N$  problems with Poissonian errors). It does not provide clear procedures for adjudicating between models with different numbers of parameters (*e.g.* one- vs. two-temperature models) or between different acceptable models (*e.g.* local minima in  $\chi^2(\theta)$  space). It can be difficult to obtain confidence intervals on parameters when complex correlations between the estimators of parameters are present (*e.g.* non-parabolic shape near the minimum in  $\chi^2(\theta)$  space).

Figure 1b shows an important alternative approach to the model fitting and goodness-of-fit problem. Here the energies of photons of observed spectrum are shown individually rather than in a binned histogram. In statistical parlance, this is called the empirical distribution function (EDF), and is advantageous over the binned histogram because the exact measured values are used. This avoids the often arbitrary choices of bin width(s) and starting point in histograms, and the sometimes-inaccurate assumption of  $\sqrt{n}$  error bars on binned values. There is a large statistical literature on the difficulty of choosing bin widths, and indeed on choosing between histograms and other data smoothing procedures. Narrow bins or smoothing kernels can be dominated by noise while wide bins can miss physically important structure.

Figure 1c illustrates another major astrostatistical question: When a “good” model is found with parameters  $\theta_0$ , what is an acceptable range of parameter values around  $\theta_0$  consistent with the data? In the example shown, we might ask: “What is the confidence interval of absorption consistent with the data at 99% significance?” This question is not simple to answer. The scientist must specify in advance whether the parameter of interest is considered in isolation or in consort with other parameters, whether the statistical treatment

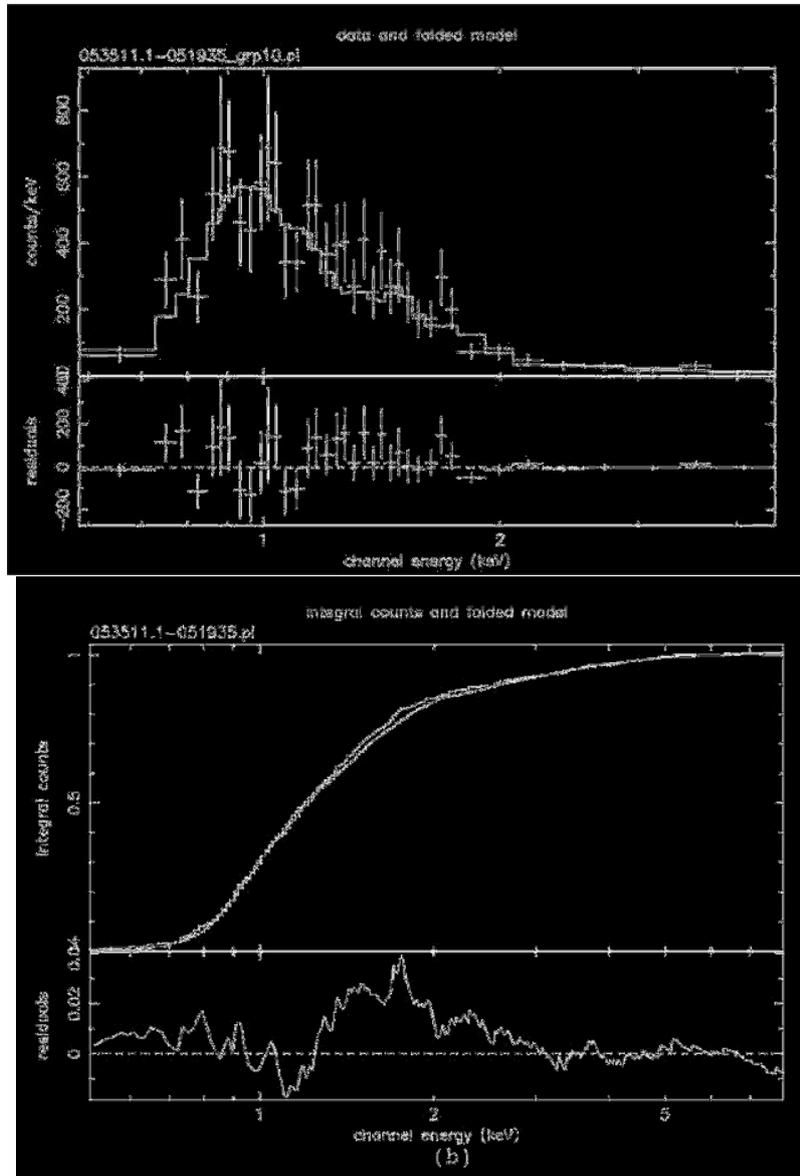


Figure 1: An example of astrophysical model fitting using a spectrum with 264 photons from the *Chandra* X-ray Observatory. (a) Best-fit thermal model (histogram) to differential binned data (separated points with error bars) obtained by minimum- $\chi^2$ . Here the absorption parameter has value  $A_V \sim 1$  mag. Data-minus-residuals appear in the bottom plot. (b) Thermal model (smooth curve) obtained by minimizing the K-S statistic, its distance to the empirical distribution (step) function. The resulting parameters are very similar to the  $\chi^2$  fit.

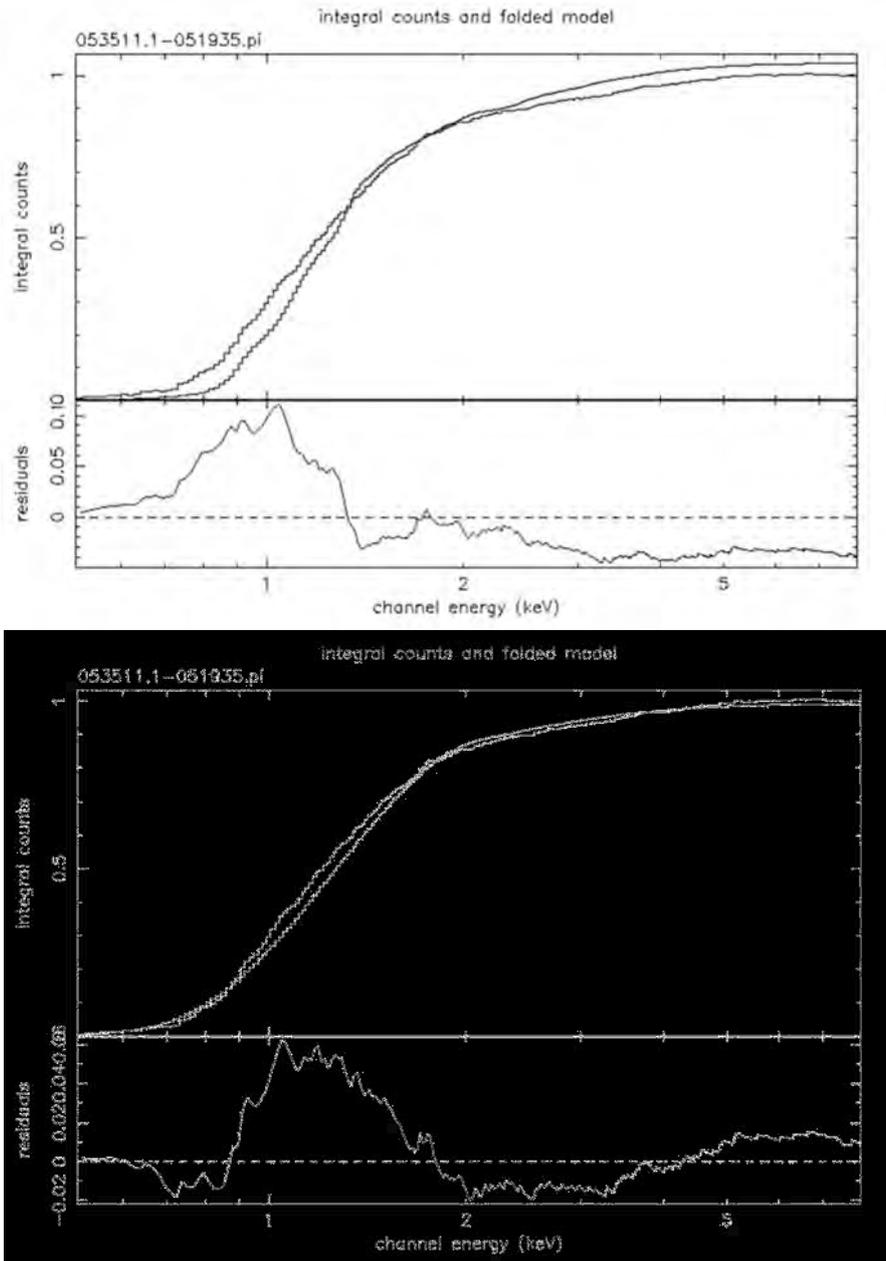


Figure 1: Continued. (c) An example of the correct model family but incorrect parameter value: thermal model with absorption set at  $A_V = 10$  mag. (d) An example of an incorrect model family: best-fit powerlaw model with absorption  $A_V \sim 1$  mag.

involves binned histograms or EDFs, and whether 67% ( $1\sigma$  equivalent), 90% or 99.7% ( $3\sigma$  equivalent) values should be reported. The statistician must decide which statistic to use, whether normal approximations are valid, and how extraneous model parameters should be treated.

Finally, Figure 1d treats a broader scientific question: Are the data consistent with *different families* of astrophysical models, irrespective of the best-fit parameter values within a family? We illustrate this here by obtaining the best-fit model using a nonthermal power law X-ray spectrum rather than a thermal plasma X-ray spectrum. Among statisticians, these are called ‘non-nested’ models. Even decisions between nested models can be tricky; for example, should the dataset in Figure 1 be modeled with thermal models with arbitrary elemental abundances, or is the assumption of solar abundances adequate?

### 3 Model Selection

A good statistical model should be parsimonious (model simplicity), conform fitted model to the data (goodness of fit), and should be easily generalizable. Occam’s Razor, a philosophical principle credited to the English Philosopher William of Ockham (1285-1349), that essentially says that the simplest solution is usually the correct one, is the main guiding principle for statistical modeling. Occam’s Razor suggests that we leave off extraneous ideas to better reveal the truth. That is, select a model that adequately accommodates the data. It neither *underfits* that excludes key variables or effects, nor *overfits* that unnecessarily be complex by including extraneous explanatory variables or effects. Underfitting induces bias and overfitting induces high variability. A model selection criterion should balance the competing objectives of conformity to the data and parsimony.

Hypothesis testing is one of the criteria used for comparing two models. Classical hypothesis testing methods are generally used for nested models. However, it does not treat models symmetrically. To set up framework for general model selection, let  $D$  denote the observed data and let  $M_1, \dots, M_k$  denote the models for  $D$  under consideration. Each model  $M_j$ , let  $f(D|\theta_j; M_j)$  and  $\ell(\theta_j) = \log f(D|\theta_j; M_j)$  denote the likelihood and loglikelihood respectively,  $\theta_j$  is a  $p_j$  dimensional parameter vector. Here  $f(D|\theta_j; M_j)$  denotes the probability density function (in the continuous case) or probability mass function (in the discrete case) evaluated at the data  $D$ . Most of the methodology can be framed as a comparison between two models  $M_1$  and  $M_2$ .

#### 3.1 Special case of Nested Models

The model  $M_1$  is said to be nested in  $M_2$ , if some coordinates of  $\theta_1$  are fixed, *i.e.*  $\theta_2 = (\alpha, \gamma)$  and  $\theta_1 = (\alpha, \gamma_0)$ , where  $\gamma_0$  is some known fixed constant vector. In this case, comparison of  $M_1$  and  $M_2$  can be considered as a classical hypothesis testing problem of  $H_0 : \gamma = \gamma_0$ .

For example, the model  $M_2$  refers to normal with mean  $\mu$  and variance  $\sigma^2$ , while  $M_1$  refers to normal with mean 0 and variance  $\sigma^2$ . The model selection problem can thus be framed in terms of statistical hypothesis testing  $H_0 : \mu = 0$ , with free parameter  $\sigma$ . There are some objections to using hypothesis testing to decide between the two models  $M_1$  and  $M_2$ , as they are not treated symmetrically by the test in which the null hypothesis is  $M_1$ .

We cannot *accept*  $H_0$ , we can only reject or fail to reject  $H_0$ . As larger samples can detect the discrepancies, they tend to make it more likely to reject the null hypothesis.

We now look at three different ideas for testing  $H_0$ .

### 3.2 Three statistical hypotheses tests

The *Wald Test*, the *Likelihood Ratio Test*, and the *Rao's Score Test*, based on maximum likelihood estimators, are collectively referred to in statistical literature as the *Holy Trinity* of statistical hypotheses tests. These statistical hypotheses tests can be used to test linear and non-linear restrictions among parameters. The three tests are described below in the case of scalar (1-dimensional) parameter  $\theta$ .

To test the null hypothesis  $H_0 : \theta = \theta_0$ , the Wald Test uses  $W_n = (\hat{\theta}_n - \theta_0)^2 / \text{Var}(\hat{\theta}_n)$ , the standardized distance between  $\theta_0$  and the maximum likelihood estimator  $\hat{\theta}_n$  based on a data of size  $n$ . The distribution of  $W_n$  is approximately the Chi-square distribution with one degree of freedom. In general variance of  $\hat{\theta}_n$  is not known, however, a close approximation is  $1/I(\hat{\theta}_n)$ , where  $I(\theta) = E((f'(X; \theta)/f(X; \theta))^2)$  is the Fisher's information,  $f$  denotes the probability density function of the random variable  $X$ , and  $f'$  denotes the derivative of  $f$  with respect to  $\theta$ . Thus  $I(\hat{\theta}_n)(\hat{\theta}_n - \theta_0)^2$  has chi-square distribution in the limit, and the Wald test rejects the null hypothesis  $H_0$ , when  $I(\hat{\theta}_n)(\hat{\theta}_n - \theta_0)^2$  is large.

The Likelihood Ratio Test uses the logarithm of ratio of likelihoods,  $\ell(\hat{\theta}_n) - \ell(\theta_0)$ , where  $\ell(\theta)$  denotes the loglikelihood at  $\theta$ . While Rao's Score Test (also known as Lagrangian Multiplier Test) uses the statistic  $S(\theta_0) = (\ell'(\theta_0))^2 / (nI(\theta_0))$ , where  $\ell'$  denotes the derivative of  $\ell$ , and as before  $I$  denotes the Fisher's Information. That is, if  $X, X_1, \dots, X_n$  denote independent random variables from a common probability density function  $f(\cdot; \theta)$ , then  $\ell'(\theta_0) = \sum_{i=1}^n (f'(X_i; \theta_0)/f(X_i; \theta_0))$ . Hence

$$S(\theta_0) = \frac{1}{nI(\theta_0)} \left( \sum_{i=1}^n \frac{f'(X_i; \theta_0)}{f(X_i; \theta_0)} \right)^2.$$

For example, in the case of data from normal (Gaussian) distribution

$$f(y; (\mu, \sigma^2)) = \frac{1}{\sigma} \phi((y - \mu)/\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{1}{2\sigma^2}(y - \mu)^2 \right\},$$

where  $\phi$  denotes the standard normal probability density function.

The three tests are equivalent to the first order of asymptotics, but differ to some extent in the second order properties. No single test among these three is uniformly better than the others.

In the regression context with data  $y_1, \dots, y_n$  and Gaussian residuals, the loglikelihood  $\ell$  is given by

$$\ell(\beta) = \log \prod_{i=1}^n \frac{1}{\sigma} \phi((y_i - \mathbf{x}_i' \beta)/\sigma).$$

### 3.3 Information Criteria based model selection

If the model  $M_1$  happens to be nested in the model  $M_2$ , the largest likelihood achievable by  $M_2$  will *always* be larger than that achievable by  $M_1$ . It suggests adding a penalty on

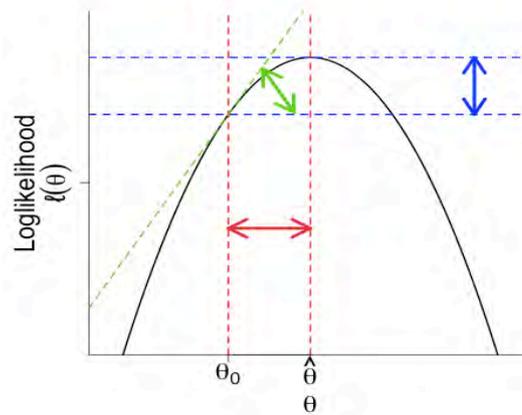


Figure 2: Wald Test is based on the distance between  $\hat{\theta}_n$  and  $\theta_0$ ; the Likelihood Ratio Test is based on the distance from  $\ell(\theta_0)$  to  $\ell(\hat{\theta}_n)$ , the loglikelihoods; the Rao's Score Test is based on the gradient of the loglikelihood at  $\theta_0$ .

“larger” models would achieve a balance between overfitting and underfitting. This leads to the so called *Penalized Likelihood approach*.

The traditional maximum likelihood paradigm, as applied to statistical modeling, provides a mechanism for estimating the unknown parameters of a model having a specified dimension and structure. Hirotugu Akaike extended this paradigm in [1] by considering a framework in which the model dimension is also unknown. He proposed a framework where both model estimation and selection could be simultaneously accomplished. Grounding in the concept of entropy, Akaike proposed an *information criterion* (AIC), which is now popularly known as *Akaike's Information Criterion*, and is defined for model  $M_j$ , as  $2\ell(\hat{\theta}_j) - 2p_j$ . The term  $2\ell(\hat{\theta}_j)$  is known as the *goodness of fit* term, and  $2p_j$  is known as the *penalty* term. This penalty term increase as the complexity of the model grows. AIC is generally regarded as the first model selection criterion, and it continues to be the most widely known and used model selection tool among practitioners.

One advantage of AIC is that it does not require the assumption that one of the candidate models is the “true” or “correct” model. It treats all the models symmetrically, unlike hypothesis testing. AIC can be used to compare nested as well as non-nested models. AIC can also be used to compare models based on different families of probability distributions. One of the disadvantages of AIC is the requirement of large samples especially in complex modeling frameworks. In addition, it is not *consistent*, in the sense that if  $p_0$  is the correct number of parameters, and  $\hat{p} = p_i$  ( $i = \arg \max_j 2\ell(\hat{\theta}_j) - 2p_j$ ), then  $\lim_{n \rightarrow \infty} P(\hat{p} > p_0) > 0$ . That is even if we have very large number of observations,  $\hat{p}$  does not approach the true value.

*Bayesian Information Criterion* (BIC), sometimes called the *Schwarz Bayesian Criterion* is another popular model selection criteria. Unlike AIC, BIC defined as

$$2\ell(\hat{\theta}_j) - p_j \log n$$

is consistent. Like AIC, the models need not be nested to be compared using BIC.

Conditions under which these two criteria are mathematically justified are often ignored in practice. Some practitioners apply them even in situations where they **should not be** applied. AIC penalizes free parameters less strongly than does the Schwartz's BIC. A note of caution: sometimes, these criteria are given a minus sign so the goal changes to finding the minimizer.

## 4 Inference for Statistics Based on the EDF

Among astronomers, the Kolmogorov-Smirnov (K-S) statistic is popular, although other EDF based statistics such as the Cramer-von Mises (C-vM) and Anderson-Darling (A-D) statistics have better sensitivity for some data-model differences. However, as we review in below, *the goodness-of-fit probabilities derived from the K-S or other EDF statistics are usually not correct when applied in model fitting situations with estimated parameters*. Astronomers are thus often making errors in EDF model fitting.

Figure 3a shows a hypothetical EDF, the cumulative frequency distribution function of the data. The three commonly used statistics, for inference on  $F$ , based on EDF mentioned above are:

Kolmogorov-Smirnov (K-S):  $\sup |F_n(x) - F(x)|$

Cramér-von Mises (C-vM):  $\int (F_n(x) - F(x))^2 dF(x)$ ,

and Anderson - Darling (A-D):  $\int \frac{(F_n(x) - F(x))^2}{F(x)(1 - F(x))} dF(x)$ .

Here  $F_n$  is the EDF,  $F$  is the model distribution function, and “sup” means the supremum. The K-S statistic is most sensitive to large-scale differences in location (*i.e.* median value) and shape between the model and data. The C-vM statistic is effective for both large-scale and small-scale differences in distribution shape. Both of these measures are relatively insensitive to differences near the ends of the distribution. This deficiency is addressed by the A-D statistic, a weighted version of the C-vM statistic to emphasize differences near the ends.

The power of these statistics is that they are distribution-free as long as  $F$  is continuous. That is, the probability distribution of these statistics is free from  $F$ . Consequently, the confidence bands for the ‘unknown’ distribution  $F$  can be obtained from standard tables of K-S, C-vM or A-D probabilities which depend only on the number of data points and the chosen significance level. A typical confidence band based on Kolmogorov-Smirnov test resembles Figure 3b.

But all these statistics are no longer distribution-free under two important and common situations: when the data are multivariate, or when the model parameters are estimated using the data. We address these situations here.

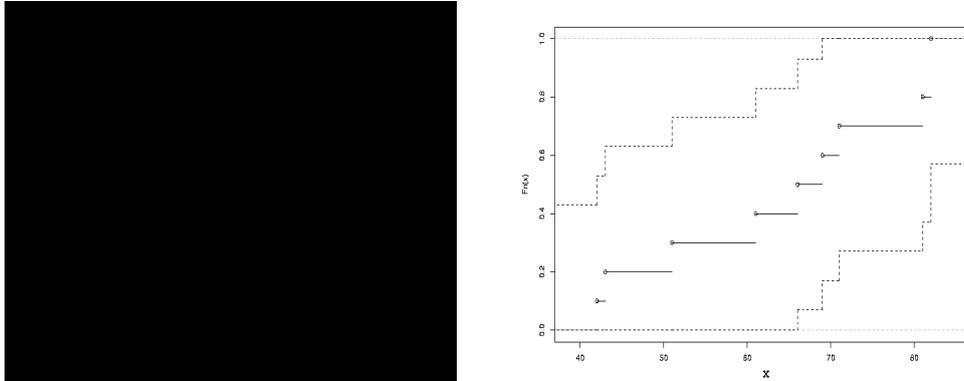


Figure 3: (a) A hypothetical EDF. (b) Confidence bands around the EDF based on the K-S statistic for 90% significance level.

#### 4.1 Failure of the multivariate case

Let  $(X_1, Y_1)$  be a data point from a bivariate distribution  $F$  on the unit square. Simpson (1951) shows that if  $F_1$  denotes the EDF of  $(X_1, Y_1)$ , then

$$P(|F_1(x, y) - F(x, y)| < .72, \text{ for all } x, y) \begin{cases} > 0.065 & \text{if } F(x, y) = xy^2 \\ < 0.058 & \text{if } F(x, y) = xy(x + y)/2. \end{cases}$$

Thus, the distribution of the K-S statistic varies with the unknown  $F$  and hence is not distribution-free when two or more dimensions are present. The K-S statistic still is a measure of “distance” between the data and model, but probabilities can not be assigned to a given value of the statistic without detailed calculation for each case under consideration. Several methodological studies in the astronomical literature discuss two-dimensional K-S tests. The results may be unreliable to degrees that can not readily be calculated.

#### 4.2 Failure when parameters are estimated from the data

The K-S statistic is also no longer distribution-free if some parameters are estimated from the dataset under consideration. For example, consider the question whether the illustrated X-ray spectrum supports a powerlaw in addition to a thermal model (Figure 1d). It may seem natural to find the best-fit powerlaw and best-fit thermal models by a procedure such as maximum likelihood, compute the K-S statistic for each case, and evaluate which model is acceptable using the probabilities in standard tables. But it has long been established that the K-S probabilities are incorrect in this circumstance (Lilliefors 1969). The K-S probabilities are only valid if the model being tested is derived independently of the dataset at hand; *e.g.* from some previous datasets or from prior astrophysical considerations.

## 5 Bootstrap resampling: A good solution

Fortunately, there is an alternative to the erroneous use of K-S procedure, although it requires a numerically intensive calculation for each dataset and model addressed. It is based

on bootstrap resampling, a data-based Monte Carlo method that has been mathematically shown to give valid estimates of goodness-of-fit probabilities under a very wide range of situations (Babu and Rao 1993).

We now outline the mathematics underlying bootstrap calculations. Let  $\{F(\cdot; \theta) : \theta \in \Theta\}$  be a family of continuous distributions parametrized by  $\theta$ . We want to test whether the univariate dataset  $X_1, \dots, X_n$  comes from  $F = F(\cdot; \theta)$  for some  $\theta = \theta_0$ . The K-S, C-vM and A-D statistics (and a few other goodness-of-fit tests) are continuous functionals of the process,  $Y_n(x; \hat{\theta}_n) = \sqrt{n}(F_n(x) - F(x; \hat{\theta}_n))$ . Here  $F_n$  denotes the EDF of  $X_1, \dots, X_n$ ,  $\hat{\theta}_n = \theta_n(X_1, \dots, X_n)$  is an estimator of  $\theta$  derived from the dataset, and  $F(x; \hat{\theta}_n)$  is the model being tested. For a simple example, if  $\{F(\cdot; \theta) : \theta \in \Theta\}$  denotes the Gaussian family with  $\theta = (\mu, \sigma^2)$ , then  $\hat{\theta}_n$  can be taken as  $(\bar{X}_n, s_n^2)$  where  $\bar{X}_n$  is the sample mean and  $s_n^2$  is the sample variance based on the data  $X_1, \dots, X_n$ . In the astrophysical example considered in §2,  $F$  is the family of thermal models with three parameters.

In the case of evaluating goodness-of-fit for a model where the parameters have been estimated from the data, the bootstrap can be computed in two different ways: the *parametric bootstrap* and the *nonparametric bootstrap*. The parametric bootstrap may be familiar to the astronomer as a well-established technique of creating fake datasets realizing the parametric model by Monte Carlo methods (*e.g.* Press et al. 1997). The actual values in the dataset under consideration are not used. The nonparametric bootstrap, in contrast, is a particular Monte Carlo realizations of the observed EDF using a “random selection with replacement” procedure.

We now outline the mathematics underlying these techniques. Let  $\hat{F}_n$  be an estimator of  $F$ , based on  $X_1, \dots, X_n$ . In order to bootstrap, we generate data  $X_1^*, \dots, X_n^*$  from the estimated population  $\hat{F}_n$  and then construct  $\hat{\theta}_n^* = \theta_n(X_1^*, \dots, X_n^*)$  using the same functional form. For example, if  $F(\cdot; \theta)$  is Gaussian with  $\theta = (\mu, \sigma^2)$  and if  $\hat{\theta}_n = (\bar{X}_n, s_n^2)$ , then  $\hat{\theta}_n^* = (\bar{X}_n^*, s_n^{*2})$ .

## 5.1 Parametric Bootstrap

The bootstrapping procedure is called parametric if  $\hat{F}_n = F(\cdot; \hat{\theta}_n)$ ; that is, we generate data  $X_1^*, \dots, X_n^*$  from the model assuming the estimated parameter values  $\hat{\theta}_n$ . The process  $Y_n^P(x) = \sqrt{n}(F_n^*(x) - F(x; \hat{\theta}_n^*))$  and the sample process  $Y_n(x) = \sqrt{n}(F_n(x) - F(x; \hat{\theta}_n))$  converge to the same Gaussian process  $Y$ . Consequently,  $L_n = \sqrt{n} \sup_x |F_n(x) - F(x; \hat{\theta}_n)|$  and  $L_n^* = \sqrt{n} \sup_x |F_n^*(x) - F(x; \hat{\theta}_n^*)|$  have the same limiting distribution. For the K-S statistic, the critical values of  $L_n$  can be derived as follows: construct  $B$  resamples based on the parametric model ( $B \sim 1000$  should suffice), arrange the resulting  $L_n^*$  values in increasing order to obtain 90 or 99 percentile points for getting 90% or 99% critical values. This procedure replaces the incorrect use of the standard probability tabulation.

## 5.2 Nonparametric Bootstrap

The nonparametric bootstrap involving resamples from the EDF;

$$\begin{aligned} Y_n^N(x) &= \sqrt{n}(F_n^*(x) - F(x; \hat{\theta}_n^*)) - B_n(x) \\ &= \sqrt{n}(F_n^*(x) - F_n(x) + F(x; \hat{\theta}_n) - F(x; \hat{\theta}_n^*)) \end{aligned}$$

is operationally easy to perform but requires an additional step of bias correction

$$B_n(x) = \sqrt{n}(F_n(x) - F(x; \hat{\theta}_n)).$$

The sample process  $Y_n$  and the bias corrected nonparametric process  $Y_n^N$  converge to the same Gaussian process  $Y$ . That is,  $L_n = \sqrt{n} \sup_x |F_n(x) - F(x; \hat{\theta}_n)|$  and  $J_n^* = \sup_x |\sqrt{n}(F_n^*(x) - F(x; \hat{\theta}_n^*)) - B_n(x)|$  have the same limiting distribution. The critical values of the distribution of  $L_n$  can then be derived as in the case of parametric bootstrap. For detailed understanding of the regularity conditions under which these results hold see Babu and Rao (2004).

## 6 Confidence Limits Under Misspecification of Model Family

We now address the more advanced problem of comparing best-fit models derived for non-nested model families; *e.g.* the powerlaw vs. thermal model fits in Figure 1. Essentially, we are asking ‘How far away’ is the unknown distribution underlying the observed dataset from the hypothesized family of models?

Let the original dataset  $X_1, \dots, X_n$  come from an unknown distribution  $H$ .  $H$  may or may not belong to the family  $\{F(\cdot; \theta) : \theta \in \Theta\}$ . Let  $F(\cdot, \theta_0)$  be the specific model in the family that is ‘closest’ to  $H$  where proximity is based on the Kullback-Leibler information,  $\int \log(h(x)/f(x; \theta)) dH(x) \geq 0$ , which arises naturally due to maximum likelihood arguments and has advantageous properties. Here  $h$  and  $f$  are the densities (*i.e.* derivatives) of  $H$  and  $F$ .

If the maximum likelihood estimator  $\hat{\theta}_n \rightarrow \theta_0$ , then  $U_n(x; \hat{\theta}_n) = \sqrt{n}(F_n(x) - F(x; \hat{\theta}_n)) - \sqrt{n}(H(x) - F(x; \theta_0))$  converges weakly to a Gaussian process  $U$  (Babu and Rao 2003). In this (nonparametric bootstrap) case,  $Y_n^N(x) = \sqrt{n}(F_n^*(x) - F(x; \hat{\theta}_n^*)) - \sqrt{n}(F_n(x) - F(x; \hat{\theta}_n))$ , and  $U_n$  converge to the same Gaussian process. For the K-S statistic, for any  $0 < \alpha < 1$ ,

$$P(\sqrt{n} \sup_x |F_n(x) - F(x; \hat{\theta}_n) - (H(x) - F(x; \theta_0))| \leq C_\alpha^*) - \alpha \rightarrow 0,$$

where  $C_\alpha^*$  is the  $\alpha$ -th quantile of  $\sup_x |\sqrt{n}(F_n^*(x) - F(x; \hat{\theta}_n^*)) - \sqrt{n}(F_n(x) - F(x; \hat{\theta}_n))|$ . This provides an estimate of the distance between the true distribution and the family of distributions under consideration (Babu and Bose 1988).

## References

- [1] Akaike, H. (1973). Information Theory and an Extension of the Maximum Likelihood Principle. In *Second International Symposium on Information Theory*, (B. N. Petrov and F. Csaki, Eds). Akademia Kiado, Budapest, 267-281.
- [2] Babu, G. J., and Bose, A. (1988). Bootstrap confidence intervals. *Statistics & Probability Letters*, **7**, 151-160.
- [3] Babu, G. J., and Rao, C. R. (1993). Bootstrap methodology. In *Computational statistics*, Handbook of Statistics **9**, C. R. Rao (Ed.), North-Holland, Amsterdam, 627-659.

- [4] Babu, G. J., and Rao, C. R. (2003). Confidence limits to the distance of the true distribution from a misspecified family by bootstrap. *J. Statistical Planning and Inference*, **115**, no. 2, 471-478.
- [5] Babu, G. J., and Rao, C. R. (2004). Goodness-of-fit tests when parameters are estimated. *Sankhyā*, **66**, no. 1, 63-74.
- [6] Bevington, P. R. (1969). *Data reduction and error analysis for the physical sciences*. McGraw-Hill.
- [7] Getman, K. V., and 23 others (2005). Chandra Orion Ultradeep Project: Observations and source lists. *Astrophys. J. Suppl.*, **160**, 319-352.
- [8] Lehmann, E. L. (1998). *Elements of Large-Sample Theory*. Springer.
- [9] Lilliefors, H. W. (1969). On the Kolmogorov-Smirnov Test for the Exponential Distribution with Mean Unknown. *Journal of the American Statistical Association*, **64**, No. 325, 387-389 .
- [10] Press, W. H. et al. (1997). *Numerical recipes in C: The art of scientific computing*. Cambridge Univ. Press.
- [11] Simpson, P. B. (1951). Note on the estimation of a bivariate distribution function. *Ann. Math. Stat.*, **22**, 476-478.



## Chapter 17

# SIMULATION AND BOOTSTRAPPING WITH R

*Notes by Arnab Chakraborty*

# Simulation and Bootstrapping

## Simulation

Many people think of a coin toss when they think about randomness. While everything in nature is random, a coin toss has the extra quality of being a completely known process. We do not know what the outcome of the next coin is going to be, but we know the average behavior quite well, *viz.*, the outcome will be either a Head or a Tail, each of which is equally likely. It is this perfect knowledge that leads us to use a coin toss when making a fair decision.

A coin toss is a random mechanism that we can repeat at our will to produce random quantities with a known behavior. Another such a device is a die. But such practical devices used to be limited in number and scope before the advent of high speed computers.

A remarkable fact about computers is that they can be instructed to produce random quantities whose general behavior is completely under control. We can, for example make the computer toss a coin with exactly a specified amount of bias for Head! We can turn the computer easily into a 100-faced die, something that is almost to construct in practice. In fact, computers can be made to perform more exotic (as well more useful) random computations that have started to play a dominant role in statistics. In this last tutorial we shall take a look at these.

## Putting it in perspective

Man has devised various techniques to deal with chance. These may be classified into roughly three categories:

1. **Wait and see:** Here we patiently wait until the end of the activity. This time-honored method remains the most frequently used one even today. Who can deny the usefulness of observing the sky carefully to learn about the random patterns? However, learning by just trial-and-error may prove too expensive. When we want to design a space station to maximize its chance of capturing some rare phenomenon, it would prove it bit too costly to simply rely on trial and error!
2. **Use probability models:** Often we have enough information (theoretical and/or based on prior observations) about the mechanism underlying a random phenomenon. For example, if a careful scrutiny of a die does not reveal any difference between the faces of a die, one may assume

that the die is fair. Such information can be expressed as a probability distribution:

The outcome  $X$  can take values 1,...,6 each with equal probability (taken to be 1/6 to keep the sum 1).

Using this mathematical formulation of our knowledge about the random phenomenon we can now analyze the distribution mathematically. For example, consider the following gambling game

A die is rolled, and the casino owner gives you Re 1 for each point shown. For example, you get Rs 6 if the die shows a 6. How much will you get on average per roll if you play this game 10000 times?

The brute force approach is to really play it 10000 times, and then divide the total amount by 10000. A smarter way is to use the probability distribution of the die. We know that on average we get 1 about 1/6 of the times, 2 about 1/6 of the times and so on. So on an average we shall get

$$(1+2+3+4+5+6)/6 = 3.5$$

per roll. Here the math turned out to be simple. But in more realistic models the math is often too tough to handle even with a computer!

3. **Simulate probability models:** Here also we start with a model. But instead of dealing with it mathematically, we make the computer to perform (or simulate) the random experiment following the model. This is possible because computers can generate random numbers ☆ Now it is just like the "wait and see" strategy, except that we do not have to wait long, since computers can perform many repetitions of a random experiment very fast. Also, no real monetary loss is involved.

☆Actually computers generate only pseudo-random numbers.

These are not random but only behave like random numbers.

However, we shall disregard the difference here.

Let us see the third approach in action for the gambling example before we move on to more serious applications.

```
values = 1:6 #the possible values
sample(values, 10, replace=T)
```

This last line asks R to **sample** 10 numbers from the vector `values`. The `replace=T` allows the same number to occur

multiple times. Run the last line repeatedly to see that how the output changes randomly. Now to solve the problem use

```
money = sample(values, 10000, replace=T)
avg = mean(money)
avg
```

This mean is indeed pretty close to the theoretical 3.5. But will it always be the case? After all it is random game. So it is a good idea to repeat the above simulation a number of times, say, 100 times. The blue lines below are as before. But these are now enveloped in some extra lines (shown in bold). This is an example of a **for loop**.

```
avgList = c() #an empty vector
for(i in 1:100) {
  money = sample(values, 10000, replace=T)
  avg = mean(money)
  avgList = c(avgList,avg) #add avg to avgList
}

mean(avgList)
var(avgList)
```

To repeat some commands for a fixed number of times (say 200) in R you put the commands inside a **for loop** like this



```
for(i in 1:200) {
  #Your commands
  #go here
}
```

For example, you may try

```
for(i in 1:5) {
  print("hello")
}
```

### Simulating from a distribution

The fundamental concept behind the above technique is that R can "roll a die" or, rather, "simulate a die". Statistically this means generating a random number from among the values 1,...,6 with probabilities 1/6 each. This is by no means the only distribution R can generate numbers from. We can for example, generate a random number from the **Uniform(0,1)** distribution.

Such a number can take any value between 0 and 1, each value being equally likely. Let us generate 100 such numbers.

```
data = runif(1000, min=0, max=1)
hist(data)
```

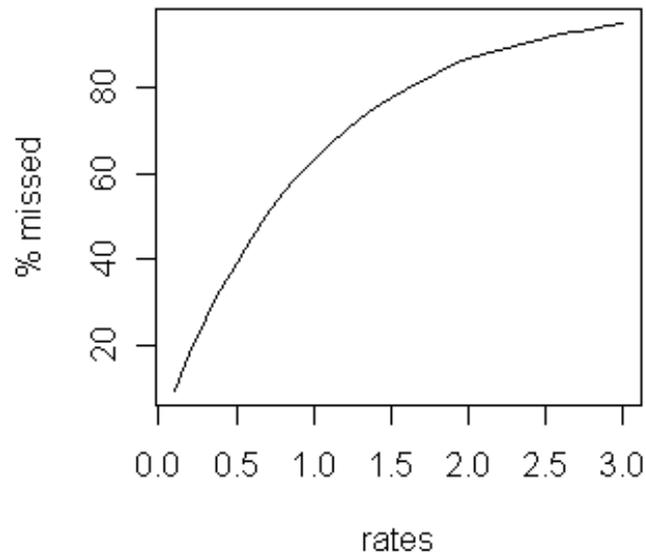
R has many probability distributions built into it, and it has ready-made functions (like **runif**) to generate data from them. The functions all share a common naming pattern. They start with the letter 'r' followed by an acronym for the distribution. Here are some examples.

```
data = rnorm(1000, mean=1, sd=2) #Normal(1,2)
hist(data)
data = rpois(1000, lambda=1) #Poisson(1)
hist(data)
data = rexp(1000, rate=2) #Exponential with mean 1/2
hist(data)
```

## Two realistic examples

### An imperfect particle counter

A huge radio active source emits particles at random time points. A particle detector is detecting them. However, the detector is imperfect and gets "jammed" for 1 sec every time a particle hits it. No further particles can be detected within this period. After the 1 sec is over, the detector again starts functioning normally. We want to know the fraction of particles that the detector is missing. Since this fraction may depend on the rate at which the source emits particles we want to get a plot like the following:



### Want a plot like this

Random emission of particles from a huge radio active source is well-studied process. A popular model is to assume that the time gaps between successive emissions are independent and have Exponential distributions.

```
gaps = rexp(999,rate=1)
```

The question now is to determine which of the particles will fail to be detected. A particle is missed if it comes within 1 sec of its predecessor. So the number of missed particles is precisely the number of gaps that are below 1.

```
miss = sum(gaps < 1)
miss
```

Now we want to repeat this experiment a number of times, say 50 times.

```
missList = c()
for(i in 1:50) {
  gaps = rexp(999,rate=1)
  miss = sum(gaps < 1)
  missList = c(missList,miss)
}
mean(missList)
var(missList)
```

All these are done for rate = 1. For other rates we need to repeat

the entire process afresh. So we shall use yet another `for` loop.

```

rates = seq(0.1,3,0.1)
mnList = c()
vrList = c()
for(lambda in rates) {
  missList = c()
  for(i in 1:50) {
    gaps = rexp(1000,rate=lambda)
    miss = sum(gaps < 1)
    missList = c(missList,miss)
  }

  mn = mean(missList)
  vr = var(missList)
  mnList = c(mnList,mn)
  vrList = c(vrList, vr)
}

```

Now we can finally make the plot.

```
plot(rates,mnList/10,ty="l",ylab="% missed")
```

We shall throw in two error bars for a good measure.

```

up = mnList + 2*sqrt(vrList)
lo = mnList - 2*sqrt(vrList)

lines(rates,up/10,col="red")
lines(rates,lo/10,col="blue")

```

**Exercise: (This exercise is difficult)** We can estimate the actual rate from the hitting times of the particles if the counter were perfect. The formula is to take the reciprocal of the average of the gaps. But if we apply the same formula for the imperfect counter, then we may get a wrong estimate of the true rate. We want to make a correction graph that may be used to correct the under estimate. Explain why the following R program achieves this. You will need to look up the online help of `cumsum` and `diff`.

```

rates = seq(0.1,3,0.1)
avgUnderEst = c()
for(lambda in rates) {
  underEst = c()
  for(i in 1:50) {
    gaps = rexp(1000,rate=lambda)
    hits = cumsum(gaps)
    obsHits = c(hits[1], hits[gaps>1])
    obsGaps = diff(obsHits)
    underEst = c(underEst,1/mean(obsGaps))
  }
  avgUnderEst = c(avgUnderEst,mean(underEst))
}

```

```
plot(avgUnderEst, rates, ty="l")
```

Can you interpret the plot?

### Simulating a galaxy

A galaxy is not a single homogeneous body and so it has different mass densities at different points. When we say that the mass density of a galaxy at a point  $(x,y,z)$  is  $f(x,y,z)$  we mean that for any region  $R$  in that galaxy the chance of detecting a body is

$$\frac{1}{M} \iiint_R f(x, y, z) dx dy dz,$$

where  $M$  is the total mass of the galaxy.

There are various models that specify different forms for  $f$ . Many of these consider the galaxy to be confined in some 2D plane, so that the density may be written as  $f(x,y)$ . The typical process for simulating a galaxy has two components:

1. First we simulate the initial state of the galaxy from some *probability* model.
2. Next, we let it evolve *deterministically* following appropriate laws of motion.

Such simulations are extremely computation intensive. We shall therefore confine ourselves to a very simple situation. To keep life simple we shall assume that the  $x$  and  $y$ -coordinates of the stars are independent normal random variables. We shall work with just 500 stars of equal masses. (It is easy to allow random masses, but in our simple implementation the simulation would then take awfully long to run!) Each star initially has tangential velocity proportional to the distance from the center.

```
x = rnorm(500, sd=4)
y = rnorm(500, sd=4)
vx = -0.5*y
vy = 0.5*x
```

Let us plot the initial stage of our galaxy.

```
oldpar = par(bg="black") #we want black background
plot(x, y, pch=".", col="white") #and white stars
par(oldpar) #restore the default (black on white)
```

This does not look much like a galaxy. We have to let it evolve over time. This basically means solving an  $n$ -body problem numerically. A popular method is the Hut-Barnes algorithm. But

we shall apply Newton's laws using R in a brute force (and slow) way. The commands are in the script file [newton.r](#).

```
|source("newton.r")
```

Here we shall perform the simulation for 100 time steps, updating the picture of the galaxy every 10 steps. With only 500 points it is difficult to see much pattern, but still one may see spiral tentacles forming.

The simulation here is admittedly very crude and inefficient. If you are interested, you may see more realistic examples (with references) at [this website](#).

## Bootstrapping

So far we are simulating from completely specified distributions. But suppose that someone gives some data and asks us to simulate from whatever is the distribution of the data. Since we do not have the distribution itself we cannot apply the above methods directly.

However, as it happens, there is an approximate way out that is simple to implement in R. Let us first see it using an artificial example.

Suppose someone has some data that was originally generated from  $N(0,1)$  distribution.

```
|origData = rnorm(1000,mean=0,sd=1)
```

Then we shall simply *resample* from this data set.

```
|newData = sample(origData,500, replace=T)
```

This is called (one form of) **bootstrapping**.

It is instructive to compare this with the true distribution.

```
|hist(newData,prob=T)
|x = seq(-3,3,0.1)
|lines(x,dnorm(x))
```

Now let us apply this on some real data.

```
|hip = read.table("HIP.dat",head=T)
|attach(hip)
```

We shall consider the `Vmag` variable. We want to generate 10 new values from its distribution (which is unknown to us).

```
|newVmag = sample(Vmag,10,replace=T)
```

Well, that was easily done, but the question is

What in the universe does this `newVmag` mean?

Does this mean we have generated 10 new galaxies? How can that be when we are just re-using the old values?

### Common sense, please!

First let us beware of the following mistaken notion.

 **Mistaken notion:** Bootstrapping is a way to increase the sample size without any extra real sampling.

If that were true, you could just keep on generating further samples from existing data and get away by *pretending* that they are new galaxies. Well, common sense tells us that this cannot be true.

 **Great lesson:** Never place statistics above common sense.

By repeatedly picking from the already available sample we are not adding anything to the information. In particular, if the original sample presents a biased view of the underlying population, then none of the resamples can cure that distortion.

### Why then should we do bootstrapping at all?

The next example explains why this is useful.

Astronomical data sets are often riddled with outliers (values that are far from the rest). To get rid of such outliers one sometimes ignores a few of the extreme values. One such method is the **trimmed mean**.

```
x = c(1,2,3,4,5,6,100,7,8,9)
mean(x)
mean(x,trim=0.1) #ignore the extreme 10% points in BOTH ends
mean(x[x!=1 & x!=100])
```

We shall compute 10%-trimmed mean of `Vmag` from the Hipparcos data set.

```
hip = read.table("HIP.dat",head=T)
attach(hip)
mean(Vmag,trim=0.1)
```

We want to estimate the standard error of this estimate. For ordinary mean we had a simple formula, but unfortunately such a

simple formula does not exist for the trimmed mean. So we shall use bootstrap here as follows. We shall generate 100 resamples each of same size as the original sample.

```
trmean = c()
for(i in 1:100) {
  resamp = sample(Vmag,length(Vmag),replace=T)
  trmean = c(trmean,mean(resamp,trim=0.1))
}
sd(trmean)
```



Bootstrapping is such a popular statistical technique that R has a package called `boot` to perform bootstrapping.

## Permutation test

Here is a variation of the same theme: **resampling** *i.e.*, randomly sampling from the original sample and pretending that it is a new sample.

The situation that we are discussing next involves two samples, say, the Hyades and the non-Hyades stars. Suppose that we want to see if their median colors are different or not. (By this, as usual, we mean to test if the medians of the *distributions underlying* the two samples are same or not.) Here we shall assume that the shapes of the two distributions are the same. Since the samples are pretty large, it is reasonable to look at the differences of the medians of the two *samples*.

```
source("hyad.r")
colH = B.V[HyadesFilter]
colnH = B.V[!HyadesFilter & !is.na(B.V)]
m = length(colH)
n = length(colnH)
median(colH)-median(colnH)
```

If the population medians are the same, then we should expect this difference to be near zero, as well. So if this difference is too large, then we have reason to suspect that the underlying distributions have different medians. The question therefore is how large is "too large"?

Sometimes statistical theory dictates a rule to decide this. But more often there is no theory to help. Then **permutation tests** provide an easy way out.

Imagine first what happens if the medians are the same. Then the two underlying distributions are actually the same (since their shapes already match by assumption). So the two samples together

is basically like a single large sample from this common distribution. If this is the case, calling a Hyades star as non-Hyades (or *vice versa*) would really not make any difference.

So we shall mix up all stars together, and then pick any  $m$  of them and call them Hyades, while the remaining  $n$  would be labeled nonHyades. This should be as good as the original sample *if the two distributions are indeed the same*.

```
pool = c(colH,colnH)
mixed = sample(pool) # sample generates a random permutation
newH = mixed[1:m]
newnH = mixed[(m+1):n]
median(newH) - median(newnH)
```

If the two distributions were really the same then this new difference should be close to the difference based on the original data.

Of course, this apparently takes us back to the original question: how close is "close enough"? But now it is easier to answer since we can repeat this random mixing many times.

```
pool = c(colH,colnH)
d = c()

for(i in 1:1000) {
  mixed = sample(pool)
  newH = mixed[1:m]
  newnH = mixed[(m+1):n]
  newDiff = median(newH) - median(newnH)
  d = c(d,newDiff)
}
```

Now that we have 1000 values of how the typical difference should look like if the distributions were the same, we can see where our original difference lies w.r.t. these values. First let us make a histogram of the typical values:

```
hist(d)
```

Well, the original value seems quite far from the range of typical values, right?

One way to make this idea precise is to compute the  $p$ -value, which is defined as the chance of observing a difference even more extreme than the original value. Here "more extreme" means "larger in absolute value".

```
orig = median(colH)-median(colnH)
sum(abs(d)>=abs(orig))/1000
```

A  $p$ -value smaller than 0.05, say, strongly indicates that the medians of the distributions are not the same.



# Chapter 18

## EM ALGORITHM

*Notes by Thriyambakam Krishnan*

EM Algorithm known to astronomers as

**Richardson-Lucy** Deconvolution or

Richardson Lucy Algorithm

E: Expectation;M: Maximization

**EM Algorithm**

- generic procedure for computing maximum Likelihood estimates (MLE) in awkward problems
- iterative procedure with E and M steps in each iteration cycle

W.H.Richardson (1972): Bayesian-based iterative method of image restoration. *Journal of Optical Society of America*, **62**, 55–59.

L.B.Lucy (1974): An iterative technique for the rectification of observed distributions. *Astronomical Journal*, **79**, 745–754

**Examples of Astronomy applications**

- image restoration
- classification, say of galaxies, gamma-ray bursts (GRB), etc.

---

### Example of Image Restoration

J.Núñez and J.Llacer (1998): Bayesian image reconstruction with space-invariant noise suppression. *Astronomy and Astrophysics Supplement Series*, **131**, 167–180.



Figure 8: Raw image of planet Saturn obtained with the WF/PC camera of the Hubble Space Telescope.

---

### Example of GRB Classification

L.Hováth, L.G.Balázs, Z.Bagoly, F.Ryde, and A.Mészáros (2006): A new definition of the

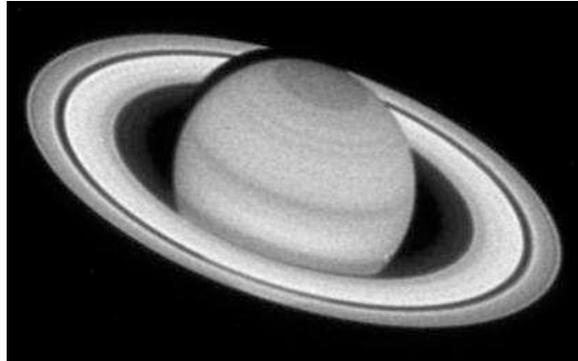


Figure 9: Reconstruction of the image of Saturn using the Richardson-Lucy algorithm.

intermediate group of gamma-ray bursts. *Astronomy & Astrophysics*.

---

#### A Bit of EM Algorithm History

- EM as a general method of ML estimation introduced by Dempster-Laird-Rubin in 1977

A.P.Dempster, N.M.Laird, and D.B.Rubin (1977): Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society*, **B 39**, 1–38.

- EM is a synthesis of innumerable similar algorithms like Richardson-Lucy

- Shepp and Vardi applied EM to image reconstruction—medical image—Positron Emission Tomography (PET)

L.A.Shepp and Y.Vardi (1982): Maximum likelihood reconstruction for emission tomography. *IEEE Transactions on Medical Imaging*, **1**, 113–122.

- Shepp-Vardi algorithm is identical to algorithms independently obtained by Richardson and Lucy in astronomy context

---

#### Linear Inverse Problems

“Incomplete-data problems” form a special case of a more general class of problems called “linear inverse problems”.

Linear Inverse Problems with positivity restrictions  
 statistical estimation problems from incomplete data  
 solve the equation

$$g(y) = \int_{D_{g_c}} h(x, y)g_c(x)dx$$

**Table 4.** Results of the EM algorithm in the  $\{\log T_{90}; \log H_{321}\}$  plane.  $k = 2$   $L_{max} = 920$ 

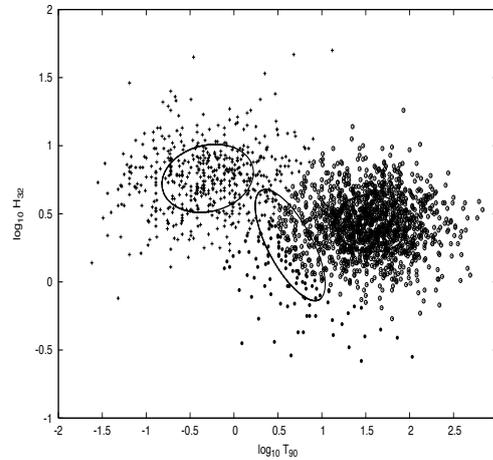
$l$	$p_l$	$a_x$	$a_y$	$\sigma_x$	$\sigma_y$	$r$
1	0.276	-0.251	0.544	0.531	0.256	0.016
2	0.725	1.479	0.132	0.479	0.287	0.123

**Table 5.** Results of the EM algorithm.  $k = 3$   $L_{max} = 980$ 

$l$	$p_l$	$a_x$	$a_y$	$\sigma_x$	$\sigma_y$	$r$
1	0.233	-0.354	0.560	0.486	0.237	0.082
2	0.154	0.722	0.057	0.480	0.432	-0.356
3	0.613	1.588	0.174	0.404	0.249	-0.048

**Table 6.** Results of the EM algorithm.  $k = 4$   $L_{max} = 982$ 

$l$	$p_l$	$a_x$	$a_y$	$\sigma_x$	$\sigma_y$	$r$
1	0.234	-0.354	0.559	0.485	0.238	0.078
2	0.148	0.704	0.062	0.447	0.432	-0.335
3	0.333	1.580	0.115	0.403	0.268	-0.141
4	0.284	1.600	0.236	0.400	0.214	0.064

**Fig. 1.** Distribution of  $N = 1956$  GRBs in the  $\{\log T_{90}; \log H_{32}\}$  plane. The  $1\sigma$  ellipses of the three Gaussian distributions are also shown, which were obtained in the ML procedure. The different symbols (crosses, filled circles and open circles) mark bursts belonging to the short, intermediate and long classes, respectively.

$D_{g_c}, D_g$ : Domains of the nonnegative real-valued functions  $g_c$  and  $g$

Image analysis:  $g_c$  true distorted image

$g$ : recorded blurred image

$g_c, g$ : grey-level intensities

function  $h(x, y)$ , which is assumed to be a bounded nonnegative function on  $D_{g_c} \times D_g$ :

characterizes the blurring mechanism

Examples: image reconstruction in PET/SPECT

traditional statistical estimation problems—grouped and truncated data

### About EM

- a method for computing MLE
- useful in many situations where direct maximization methods are tedious or impossible

### Examples of these situations

- Missing data
- Incomplete data
- Censored observations
- Difficult distributions
- Unsupervised data

- Blurred images
  - .....
- 

### **Introduction to EM**

- EM (Expectation–Maximization) algorithm
  - computing maximum likelihood estimates
  - “incomplete data problems”—nasty
  - “complete data problem”—easier MLE
  - “missing values” or “augmented data”
  - “statistically tuned” optimization method
  - finding the marginal posterior mode
- 

### **Informal Description of EM**

- formulate ‘nice’ complete-data problem
  - write down log-likelihood of complete-data problem
  - start with some initial estimates of parameters
  - **E-Step:** compute conditional expectation of log-likelihood of complete data problem given actual data, at current parameter values
  - **M-Step:** recompute parameter estimates using the simpler MLE for complete data problem
  - repeat E- and M-steps until convergence
- 

### **EXAMPLES OF EM ALGORITHM**

- Normal mixtures (Cluster Analysis; Classification)
- Missing data from bivariate normal

- Image Restoration: Tomography
- Hidden Markov models
- Neural Networks
- .....

Image restoration problem same in Astronomy and Medical Imaging

### Model for Image Restoration

Vector of emission densities (gray levels) (parameters to be estimated) at  $n$  pixels (locations) of true image:  $\Lambda = (\lambda_1, \dots, \lambda_n)^T$

Vector of the observations at  $d$  positions of device  $\mathbf{y} = (y_1, \dots, y_d)^T$

### Poisson model for counts

- Given  $\Lambda$ ,  $y_1, \dots, y_d$ , are conditionally independent Poisson

$$Y_j \sim P(\mu_j), \quad \mu_j = \sum_{i=1}^n \lambda_i p_{ij} \quad (j = 1, \dots, d),$$

- $p_{ij}$ : conditional probability that photon/positron is counted by  $j$ th detector given that it was emitted from  $i$ th pixel (in **PET**; known detector design parameters);
- $p_{ij}$ : known point spread function (fraction of light from location  $j$  observed at position  $i$ ) (**Image Processing**)

### Heuristic Solution for Image Restoration:

We use PET language, but solution is more generally valid under model

$z_{ij}$ : number of photons/positrons emitted by pixel  $i$  recorded at  $j^{\text{th}}$  detector

$$(i = 1, \dots, n; j = 1, \dots, d)$$

$$Z_{ij} \sim P(\lambda_i p_{ij}) \quad (i = 1, \dots, n; j = 1, \dots, d).$$

$$y_j = \sum_{i=1}^n z_{ij}, \quad (j = 1, \dots, d),$$

$$\lambda_i p_{ij} / \sum_{h=1}^n \lambda_h p_{hj} \quad (i = 1, \dots, n; j = 1, \dots, d)$$

is proportion of  $y_j$  emitted by  $i$ .

If we know  $\Lambda$ ,  $Z_{ij}$  estimated by

$$y_j \lambda_i p_{ij} / \sum_{h=1}^n \lambda_h p_{hj} \quad (E)$$

Then  $\lambda_i$  is estimated by

$$\sum_{i=1}^d z_{ij}$$

Hence the following iteration:

$$\lambda_i^{(k+1)} = \lambda_i^{(k)} \sum_{j=1}^d \left\{ y_j p_{ij} / \sum_{h=1}^n \lambda_h^{(k)} p_{hj} \right\} \\ (i = 1, \dots, n)$$

- iteration converges to MLE under Poisson Model
- this is Richardson-Lucy algorithm

### Above Heuristic Solution as EM Algorithm

Given problem: an incomplete-data problem

Consider  $z_{ij}$  as missing data

consistent with  $y_j = \sum_{i=1}^n z_{ij}$

Regard this as complete data

Complete-data log-likelihood:

$$\log L_c(\Lambda) = \sum_{i=1}^n \sum_{j=1}^d \{-\lambda_i p_{ij} + z_{ij} \log(\lambda_i p_{ij}) - \log z_{ij}!\}$$

leading to complete-data MLE of  $\lambda_i$  as

$$\frac{\sum_{i=1}^d z_{ij}}{\sum_{i=1}^d p_{ij}} \quad (M)$$

We exploit (E) and (M) in an iterative scheme

Let  $Z_{ij}$  be the random variable corresponding to observation  $z_{ij}$ . Given  $\mathbf{y}$  and  $\Lambda^{(k)}$

$$Z_{ij} \sim \text{Binomial}(y_j, \lambda_i^{(k)} p_{ij} / \sum_{h=1}^n \lambda_h^{(k)} p_{hj})$$

$$\mathbf{E}_{\Lambda^{(k)}}(Z_{ij} | \mathbf{y}) = z_{ij}^{(k)},$$

where

$$z_{ij}^{(k)} = y_j \lambda_i^{(k)} p_{ij} / \sum_{h=1}^n \lambda_h^{(k)} p_{hj} \quad (\mathbf{E} - \text{Step})$$

Replace  $z_{ij}$  by  $z_{ij}^{(k)}$  in (M) on the  $(k+1)^{\text{st}}$  iteration (**M-Step**)

$$\begin{aligned} \lambda_i^{(k+1)} &= q_i^{-1} \sum_{j=1}^d p_{ij} E_{\Lambda_i^{(k)}}(Z_{ij} | \mathbf{y}) \\ &= \lambda_i^{(k)} q_i^{-1} \sum_{j=1}^d \{y_j p_{ij} / \sum_{h=1}^n \lambda_h^{(k)} p_{hj}\} \end{aligned}$$

$$(i=1, \dots, n) \text{ where } q_i = \sum_{j=1}^d p_{ij}.$$

### Normal Mixtures:

Data:

3.54 3.90 3.93 5.19 3.58 4.60 3.85 4.69 4.29 4.067 3.77 3.45 5.36 2.62 4.80 4.65 3.65  
3.67 6.23 3.35 1.58 -0.19 -1.89 0.08 0.34 0.90 -0.03 0.55 -0.57 -1.20

Histogram of 30 observations

Suspected to be from a mixture of two normals

Let us model as a mixture of  $\mathcal{N}(0, 1), \mathcal{N}(\mu, 4)$

Mixture proportions  $1 - p, p, 0 < p < 1$

MLE of two parameters  $p, \mu$

### Mixture Density:

$$\phi(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2}$$

$$\phi(y - \mu) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(y-\mu)^2}$$

$\mathcal{N}(0, 1)$  and  $\mathcal{N}(\mu, 1)$  densities

Mixture of these two normal densities:

$$f(y; p, \mu) = \{p\phi(y - \mu) + (1 - p)\phi(y)\}$$

$p, \mu$  unknown,  $0 < p < 1$

Sample  $\mathbf{Y} = (Y_1, Y_2, \dots, Y_n)$  from  $f(y; p, \mu)$

To find MLE of  $p, \mu$

Mixture resolution;

Unsupervised learning;

Cluster Analysis

### Maximizing log-Likelihood:

Likelihood:

$$L_{\mathbf{y}}(p, \mu) = \prod_{i=1}^n [p\phi(y_i - \mu) + (1 - p)\phi(y_i)]$$

To maximize

1. Find  $\ell(p, \mu) = \log L_{\mathbf{y}}(p, \mu)$

2. Find  $\dot{\ell}(p, \mu) = \left( \frac{\partial \ell(p, \mu)}{\partial p}, \frac{\partial \ell(p, \mu)}{\partial \mu} \right)$

3. Solve  $\dot{\ell}(p, \mu) = 0$

4. Find

$$\ddot{\ell}(p, \mu) = \begin{bmatrix} \frac{\partial^2 \ell(p, \mu)}{\partial p^2} & \frac{\partial^2 \ell(p, \mu)}{\partial p \partial \mu} \\ \frac{\partial^2 \ell(p, \mu)}{\partial p \partial \mu} & \frac{\partial^2 \ell(p, \mu)}{\partial \mu^2} \end{bmatrix} = -\mathbf{I}(p, \mu; \mathbf{y})$$

called **Observed Information Matrix**

**Newton-Raphson:** Iterate:

$$(p^{(k+1)}, \mu^{(k+1)}) = \mathbf{I}^{-1}(p, \mu; \mathbf{y}) \dot{\ell}(p^{(k)}, \mu^{(k)})^T$$

**Fisher's Scoring Method:** replace

$\mathbf{I}$  by  $\mathbf{I}(p, \mu) = E(-\mathbf{I}(p, \mu; \mathbf{y}))$

called the **Expected Information Matrix**.

Both are possible, but messy.

### Heuristic Description of EM for this Problem:

- Consider the corresponding supervised estimation problem
- Supervised data identifies group of each case
- If model is correct, one group has mean 0 (group 0) and other group has mean  $\mu \neq 0$  (group 1)
- $\mu$  is estimated by sample mean of group 1
- $p$  can be estimated by the proportion in group 1
- But we do not have supervised data

- Make an initial guess of parameters, say  $\mu = 2, p = 0.75$
- **E-Step:** Using this find prob say  $\pi_i$  of case  $i$  from group 1
- This is exactly like posterior prob in discriminant analysis
- **M-Step:** Mean of  $\pi_i$  is an estimate of  $p$  for group 1  
Weighted mean of  $y_i$  with weights  $\pi_i$  is estimate of  $\mu$
- Iterate E-and M-steps until convergence
- Convergence test by say, successive parameter values
- This is EM algorithm

### Incomplete and Complete Data:

Two groups:

Group 1 with mean  $\mu$  (proportion  $p$ )

Group 0 with mean 0 (proportion  $1 - p$ )

**Pretend** for each  $i$ , we know the group, say  $z_i = 1$  or 0

**Supervised Learning Problem** (Discriminant Analysis)

$\mathbf{Z} = (Z_1, Z_2, \dots, Z_n)$  i.i.d. with

$$P(Z_i = 0) = 1 - p; P(Z_i = 1) = p$$

$$(Y_i|Z_i = 0) \sim \mathcal{N}(0, 1), (Y_i|Z_i = 1) \sim \mathcal{N}(\mu, 1)$$

Then  $(Z_i, Y_i), i = 1, 2, \dots, n$  called **Complete Data**

$(Y_i), i = 1, 2, \dots, n$  called **Incomplete Data**

### Complete Data Problem Solution:

Complete Data Likelihood:

$$L_{\mathbf{z}, \mathbf{y}}(p, \mu) = \prod_{i=1}^n p^{z_i} \phi(y_i - \mu)^{z_i} (1 - p)^{1 - z_i} \phi(y_i)^{1 - z_i}$$

$$= \text{constant} + p \sum z_i (1 - p)^{n - \sum z_i} \prod_{i=1}^n \phi(y_i - \mu)^{z_i}$$

$$\ell_{\mathbf{z}, \mathbf{y}}(p, \mu) = \log L_{\mathbf{z}, \mathbf{y}}(p, \mu) = \text{constant}$$

$$+ \log p \sum_{i=1}^n z_i + \log(1 - p)(n - \sum_{i=1}^n z_i) - \frac{1}{2} \sum_{i=1}^n z_i (y_i - \mu)^2 \quad (A)$$

$$\dot{\ell} = 0 \implies$$

$$\hat{p} = \frac{\sum_{i=1}^n z_i}{n}; \hat{\mu} = \frac{\sum_{i=1}^n z_i y_i}{\sum_{i=1}^n z_i} \quad (B)$$

MLE for Complete Data Problem simple  
EM exploits this simplicity in an iterative process

---

**E-Step:**

For iteration, initial values  $p^{(0)}, \mu^{(0)}$   
 $k^{\text{th}}$  iteration values  $p^{(k)}, \mu^{(k)}$   
Find surrogate for  $\ell_{\mathbf{z}, \mathbf{y}}(p, \mu)$  by taking

$$\begin{aligned} & E_{p^{(k)}, \mu^{(k)}}(\ell_{\mathbf{z}, \mathbf{y}}(p, \mu) | \mathbf{Y} = \mathbf{y}) \\ &= \log p \sum_{i=1}^n z_i^{(k+1)} + \log(1-p) \sum_{i=1}^n (n - z_i^{(k+1)}) \\ &\quad - \frac{1}{2} \sum_{i=1}^n z_i^{(k+1)} (y_i - \mu)^2 \end{aligned} \tag{C}$$

where

$$\begin{aligned} z_i^{(k+1)} &= E_{p^{(k)}, \mu^{(k)}}(Z_i | Y_i = y_i) \\ &= P_{p^{(k)}, \mu^{(k)}}(Z_i = 1 | Y_i = y_i) \end{aligned}$$


---

**M-Step:**

Equation (C) same form as (A); hence MLE same form as (B)

$$p^{(k+1)} = \frac{\sum_{i=1}^n z_i^{(k+1)}}{n}; \mu^{(k+1)} = \frac{\sum_{i=1}^n z_i^{(k+1)} y_i}{\sum_{i=1}^n z_i^{(k+1)}}$$

$$\begin{aligned} z_i^{(k+1)} &= E(Z_i | Y_i = y_i) = P(Z_i = 1 | Y_i = y_i) \\ &= \frac{p^{(k)} \phi(y_i - \mu^{(k)})}{p^{(k)} \phi(y_i - \mu^{(k)}) + (1 - p^{(k)}) \phi(y_i)} \end{aligned}$$

which is just the posterior probability (as in Discriminant Analysis)

Iterate E- and M-steps until convergence

---

Results of EM Algorithm (starting  $p = 0.6; \mu = 3.5$ ):

Iteration	$p$	$\mu$
0	0.6	3.5
1	0.68	4.1
2	0.67	4.15
3	0.67	4.15

Cluster Analysis using EM algorithm for Normal Mixtures will be discussed in the **Cluster Analysis** lecture.

### Example 2: Bivariate Normal Data with Missing Values: Computations

Variate 1: 8 11 16 18 6 4 20 25 9 13  
 Variate 2: 10 14 16 15 20 4 18 22 ? ?

Results of the EM Algorithm for Example 2.1 (Missing Data on One Variate).

Iteration	$\mu_1^{(k)}$	$\mu_2^{(k)}$	$\sigma_{11}^{(k)}$	$\sigma_{12}^{(k)}$	$\sigma_{22}^{(k)}$	$-2 \log L(\theta^{(k)})$
1	13	14.8750	40	32.3750	28.8593	1019.64
2	13	14.5528	40	21.2385	24.5787	210.930
3	13	14.5992	40	20.9241	26.2865	193.331
4	13	14.6116	40	20.8931	26.6607	190.550
5	13	14.6144	40	20.8869	26.7355	190.014
6	13	14.6150	40	20.8855	26.7503	189.908
7	13	14.6151	40	20.8852	26.7533	189.886
8	13	14.6152	40	20.8851	26.7538	189.882
9	13	14.6152	40	20.8851	26.7539	189.881
10	13	14.6152	40	20.8851	26.7540	189.881
11	13	14.6152	40	20.8851	26.7540	189.881
12	13	14.6152	40	20.8851	26.7540	189.881
$\infty$	13	14.6152	40	20.8851	26.7540	189.881

### THEORY AND METHODOLOGY OF EM

- Incomplete-data problems
- E- and M-steps
- Convergence of EM
- Rate of convergence of EM
- Standard error computation in EM

**Incomplete-Data Problems**

Incomplete-data problem; incomplete-data likelihood  $L$

Missing or latent or augmented data; missing data (conditional) distribution

Complete-data problem; complete-data likelihood

variety of statistical data models, including mixtures, convolutions, random effects, grouping, censoring, truncated and missing observations

observed data  $\mathbf{y}$ ; density  $g(\mathbf{y}|\theta)$ ; sample space  $\mathcal{Y}$ ; objective is to maximize  $\ell_{\mathbf{y}}(\theta) = \log(g(\mathbf{y}|\theta))$

Complete data  $\mathbf{x}$  density  $f(\mathbf{x}|\theta)$ ; sample space  $\mathcal{X}$

$$g(\mathbf{y}|\theta) = \int_{\mathbf{y}=\mathbf{y}(\mathbf{x})} f(\mathbf{x}|\theta) dx$$

$\mathbf{S}(\theta)$ : gradient vector (Fisher score vector)

$\mathbf{H}(\theta)$ : Hessian matrix of  $\ell_{\mathbf{y}}(\theta)$

$\mathbf{I}(\theta) = -\mathbf{H}(\theta)$ : observed information matrix

expected value of  $I(\theta) = \mathcal{I}(\theta)$ : expected information matrix

$\mathbf{S}(\theta) = \mathbf{O}$ : likelihood equations

$-\mathbf{H}^{-1}$ : estimate of asymptotic covariance matrix

$\mathcal{I}^{-1}(\hat{\theta})$  at  $\theta = \hat{\theta}$ : estimate of the asymptotic covariance matrix

**E- and M-Steps**

$$\ell_{\mathbf{y}}(\theta) = \log(g(\mathbf{y}|\theta))$$

$$\ell_{\mathbf{x}}(\theta) = \log(f(\mathbf{x}|\theta))$$

$$\ell_{\mathbf{x}|\mathbf{y}}(\theta) = \log(k(\mathbf{x}|\mathbf{y}, \theta))$$

$$k(\mathbf{x}|\mathbf{y}, \theta) = f(\mathbf{x}|\mathbf{y}, \theta)/g(\mathbf{y}|\theta)$$

$$\ell_{\mathbf{x}}(\theta) = \ell_{\mathbf{y}}(\theta) + \ell_{\mathbf{x}|\mathbf{y}}(\theta)$$

$$Q(\theta|\theta') = \ell_{\mathbf{y}}(\theta) + H(\theta|\theta')$$

**E-Step:** Compute

$$Q(\theta|\theta^{(k)}) = E(\log(f(\mathbf{x}|\theta)))$$

where the expectation is taken with respect to  $k(\mathbf{x}|\mathbf{y}, \theta^{(k)})$

**M-Step:** Maximize  $Q(\theta|\theta^{(k)})$  as a function of  $\theta$ , to obtain  $\theta^{(k+1)}$

---

Appealing properties:

1. It is numerically stable with each EM iteration increasing the likelihood.
2. Under fairly general conditions, it has reliable global convergence properties.
3. It is easily implemented, analytically and computationally.
4. It can be used to provide estimates of ‘missing data’.

Drawbacks:

1. It does not provide a natural covariance estimator for the MLE.
  2. It is sometimes very slow to converge.
- 

#### Standard Errors of EM Estimates

1. No natural way to compute covariance matrix
2. Augment EM computation with standard error computation
3. Exploit EM computations
4. Known methods based on observed information matrix, the expected information matrix or on resampling methods

numerically differentiate  $\ell_{\mathbf{y}}$  to obtain the Hessian. In a EM-aided differentiation approach, Meilijson suggests perturbation of the incomplete-data score vector to compute the observed information matrix.

---

Meng and Rubin: **Supplemented EM (SEM)** algorithm numerical techniques are used to compute the derivative of the EM operator  $\mathbf{M}$  and using this together with the complete-data observed information matrix in the equation

$$\mathbf{H} = \ddot{Q}(\mathbf{I} - \dot{\mathbf{M}})$$

the incomplete-data observed information matrix is computed.

Jamshidian and Jennrich: approximately obtains observed information matrix by numerical differentiation and suggest various alternatives to the SEM algorithm

# Chapter 19

## MULTIVARIATE ANALYSIS

*Notes by Thriyambakam Krishnan*

**Multivariate analysis:** The statistical analysis of data containing observations on two or more *variables* each measured on a set of *objects* or *cases*.

C. Wolf, K. Meisenheimer, M. Kleinheinrich, A. Borch, S. Dye, M. Gray, L. Wisotzki, E. F. Bell, H.-W. Rix, A. Cimatti, G. Hasinger, and G. Szokoly: "A catalogue of the Chandra Deep Field South with multi-colour classification and photometric redshifts from COMBO-17," *Astron. & Astrophys.*, 2004.

65 variables: Rmag, e.Rmag, ApDRmag, mumax, Mcz, e.Mcz, MCzml, . . . , IFD, e.IFD

63,501 objects: galaxies

<http://astrostatistics.psu.edu/datasets/COMBO17.dat>

---

Rmag	mumax	Mcz	MCzml	chi2red	UjMAG	BjMAG	VjMAG
2	5	6	8	9	10	12	14
24.995	24.214	0.832	1.400	0.64	-17.67	-17.54	-17.76
25.013	25.303	0.927	0.864	0.41	-18.28	-17.86	-18.20
24.246	23.511	1.202	1.217	0.92	-19.75	-19.91	-20.41
25.203	24.948	0.912	0.776	0.39	-17.83	-17.39	-17.67
25.504	24.934	0.848	1.330	1.45	-17.69	-18.40	-19.37
23.740	24.609	0.882	0.877	0.52	-19.22	-18.11	-18.70
25.706	25.271	0.896	0.870	1.31	-17.09	-16.06	-16.23
25.139	25.376	0.930	0.877	1.84	-16.87	-16.49	-17.01
24.699	24.611	0.774	0.821	1.03	-17.67	-17.68	-17.87
24.849	24.264	0.062	0.055	0.55	-11.63	-11.15	-11.32
25.309	25.598	0.874	0.878	1.14	-17.61	-16.90	-17.58
24.091	24.064	0.173	0.193	1.12	-13.76	-13.99	-14.41
25.219	25.050	1.109	1.400	1.76	-18.57	-18.49	-18.76
26.269	25.039	0.143	0.130	1.52	-10.95	-10.30	-11.82
23.596	23.885	0.626	0.680	0.78	-17.75	-18.21	-19.11
23.204	23.517	1.185	1.217	1.79	-20.50	-20.14	-20.30
25.161	25.189	0.921	0.947	1.68	-17.87	-16.13	-16.30
22.884	23.227	0.832	0.837	0.20	-19.81	-19.42	-19.64
24.346	24.589	0.793	0.757	1.86	-18.12	-18.11	-18.58
25.453	24.878	0.952	0.964	0.72	-17.77	-17.81	-18.06
25.911	24.994	0.921	0.890	0.96	-17.34	-17.59	-18.11
26.004	24.915	0.986	0.966	0.95	-17.38	-16.98	-17.30
26.803	25.232	1.044	1.400	0.78	-16.67	-18.17	-19.17
25.204	25.314	0.929	0.882	0.64	-18.05	-18.68	-19.63
25.357	24.735	0.901	0.875	1.69	-17.64	-17.48	-17.67
24.117	24.028	0.484	0.511	0.84	-16.64	-16.60	-16.83
26.108	25.342	0.763	1.400	1.07	-16.27	-16.39	-15.54
24.909	25.120	0.711	1.152	0.42	-17.09	-17.21	-17.85
24.474	24.681	1.044	1.096	0.69	-18.95	-18.95	-19.22
23.100	24.234	0.826	1.391	0.53	-19.61	-19.85	-20.28
22.009	22.633	0.340	0.323	2.88	-17.49	-17.64	-18.17

·  
·  
·

---

### The goals of multivariate analysis:

Generalize univariate statistical methods  
 Multivariate means, variances, and covariances  
 Multivariate probability distributions  
 Reduce the number of variables  
 Structural simplification  
 Linear functions of variables (principal components)  
 Investigate the dependence between variables      Canonical correlations  
 Statistical inference  
 Confidence regions  
 Multivariate regression  
 Hypothesis testing  
 Classify or cluster “similar” objects  
 Discriminant analysis  
 Cluster analysis  
 Prediction

---

### Organizing the data

$p$ : The number of variables

$n$ : The number of objects (cases) (the sample size)

$x_{ij}$ : the  $i^{\text{th}}$  observation on the  $j^{\text{th}}$  variable

Data array or data matrix

		Variables			
		1	2	...	$p$
Objects	1	$x_{11}$	$x_{12}$	...	$x_{1p}$
	2	$x_{21}$	$x_{22}$	...	$x_{2p}$
	$\vdots$	$\vdots$	$\vdots$		$\vdots$
	$n$	$x_{n1}$	$x_{n2}$	...	$x_{np}$

---

Data matrix

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

We write  $\mathbf{X}$  as  $n$  row or as  $p$  column vectors

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_p]$$

Matrix methods are essential to multivariate analysis

We will need only small amounts of matrix methods, e.g.,

$\mathbf{A}^T$ : The transpose of  $\mathbf{A}$

$|\mathbf{A}|$ : The determinant of  $\mathbf{A}$

$(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$

## Descriptive Statistics

The sample mean of the  $j^{\text{th}}$  variable:

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{ij}$$

The sample mean vector:

$$\bar{\mathbf{x}} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_p \end{bmatrix}$$

The sample variance of the  $j^{\text{th}}$  variable:

$$s_{jj} = \frac{1}{n-1} \sum_{k=1}^n (x_{kj} - \bar{x}_j)^2$$

The sample covariance of variables  $i$  and  $j$ :

$$s_{ij} = s_{ji} = \frac{1}{n-1} \sum_{k=1}^n (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)$$

[**Question:** Why do we divide by  $(n - 1)$  rather than  $n$ ?]

The sample covariance matrix:

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1p} \\ s_{21} & s_{22} & \cdots & s_{2p} \\ \vdots & \vdots & & \vdots \\ s_{p1} & s_{p2} & \cdots & s_{pp} \end{bmatrix}$$

The sample correlation coefficient of variables  $i$  and  $j$ :

$$r_{ij} = \frac{s_{ij}}{\sqrt{s_{ii}s_{jj}}}$$

Note that  $r_{ii} = 1$  and  $r_{ij} = r_{ji}$

The sample correlation matrix:

$$\mathbf{R} = \begin{bmatrix} 1 & r_{12} & \cdots & r_{1p} \\ r_{21} & 1 & \cdots & r_{2p} \\ \vdots & \vdots & & \vdots \\ r_{p1} & r_{p2} & \cdots & 1 \end{bmatrix}$$

$\mathbf{S}$  and  $\mathbf{R}$  are *symmetric*

$\mathbf{S}$  and  $\mathbf{R}$  are *positive semidefinite*:  $\mathbf{v}^T \mathbf{S} \mathbf{v} \geq 0$  for any vector  $\mathbf{v}$ .

Equivalently,

$$s_{11} \geq 0, \begin{vmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{vmatrix} \geq 0, \begin{vmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{vmatrix} \geq 0,$$

etc.

If  $\mathbf{S}$  is singular so is  $\mathbf{R}$  and conversely.

If  $n \leq p$  then  $\mathbf{S}$  and  $\mathbf{R}$  will be *singular*:

$$|\mathbf{S}| = 0 \text{ and } |\mathbf{R}| = 0$$

Which practical astrophysicist would attempt a statistical analysis with 65 variables and a sample size smaller than 65?

$\mathbf{v}^T \mathbf{S} \mathbf{v} > 0$  is the variance of  $\mathbf{v}^T \mathbf{X}$

If  $n > p$  then, generally (*but not always*),  $\mathbf{S}$  and  $\mathbf{R}$  are strictly *positive definite*:

Then  $\text{Var}(\mathbf{v}^T \mathbf{X}) = \mathbf{v}^T \mathbf{S} \mathbf{v} > 0$  for any non-zero vector  $\mathbf{v}$

Equivalently,

$$s_{11} > 0, \begin{vmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{vmatrix} > 0, \begin{vmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{vmatrix} > 0, \text{ etc.}$$

However, if  $n > p$  and  $|\mathbf{S}| = 0$  then for some  $\mathbf{v}$   $\text{Var}(\mathbf{v}^T \mathbf{X}) = 0$  implying  $\mathbf{v}^T \mathbf{X}$  is a constant and there is a linear relationship between the components of  $\mathbf{X}$

In this case, we can eliminate the dependent variables: **dimension reduction**

### The COMBO-17 data

Variables: Rmag,  $\mu$ max, Mcz, MCzml, chi2red, UjMAG, BjMAG, VjMAG

$p = 8$  and  $n = 3462$

The sample mean vector:

Rmag	mumax	Mcz	MCzml	chi2red	UjMAG	BjMAG	VjMAG
23.939	24.182	0.729	0.770	1.167	-17.866	-17.749	-18.113

The sample covariance matrix:

	Rmag	mumax	Mcz	MCzml	chi2red	UjMAG	BjMAG	VjMAG
Rmag	2.062	1.362	0.190	0.234	0.147	0.890	1.015	1.060
mumax	1.362	1.035	0.141	0.172	0.079	0.484	0.578	0.610
Mcz	0.190	0.141	0.102	0.105	-0.004	-0.438	-0.425	-0.428
MCzml	0.234	0.172	0.105	0.141	-0.009	-0.416	-0.414	-0.419
chi2red	0.147	0.079	-0.004	-0.009	0.466	0.201	0.204	0.221
UjMAG	0.890	0.484	-0.438	-0.416	0.201	3.863	3.890	3.946
BjMAG	1.015	0.578	-0.425	-0.414	0.204	3.890	4.500	4.219
VjMAG	1.060	0.610	-0.428	-0.419	0.221	3.946	4.219	4.375

### Advice given by some for Correlation Matrix:

- Use no more than two significant digits.
- Starting with the physically most important variable, reorder variables by descending correlations.
- Suppress diagonal entries to ease visual clutter.
- Suppress zeros before the decimal point.

COMBO-17's correlation matrix

	Rmag	mumax	Mcz	MCzml	chi2red	UjMAG	BjMAG	VjMAG
Rmag		.9	.4	.4	.2	.3	.3	.3
mumax	.9		.4	.5	.1	.2	.3	.3
Mcz	.4	.4		.9	-.0	-.7	-.6	-.6
MCzml	.4	.5	.9		-.0	-.6	-.5	-.5
chi2red	.2	.1	-.0	-.0		.2	.1	.2
UjMAG	.3	.2	-.7	-.6	.2		.9	1.0
BjMAG	.3	.3	-.6	-.5	.1	.9		1.0
VjMAG	.4	.3	-.6	-.5	.2	1.0	1.0	

**Reminder:** Correlations measure the strengths of linear relationships between variables *if* such relationships are valid

$\{UjMAG, BjMAG, VjMAG\}$  are highly correlated; perhaps, two of them can be eliminated. Similar remarks apply to  $\{Rmag, mumax\}$  and  $\{Mcz, Mczml\}$ .

$chi2red$  has small correlation with  $\{mumax, Mcz, Mczml\}$ ; we would retain  $chi2red$  in the subsequent analysis

### Multivariate probability distributions

Find the *probability* that a galaxy chosen *at random* from the population of *all* COMBO-17 type galaxies satisfies

$$4 * Rmag + 3 * mumax + |Mcz-MCzml| - chi2red + (UjMAG+BjMAG)^2 + VjMAG^2 < 70?$$

$X_1$ : Rmag  
 $X_2$ : mumax  
 ...  
 $X_7$ : BjMAG  
 $X_8$ : VjMAG

We wish to make probability statements about random *vectors*  
 $p$ -dimensional random vector:

$$\mathbf{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_p \end{bmatrix}$$

where  $X_1, \dots, X_p$  are random variables

$\mathbf{X}$  is a *continuous random vector* if  $X_1, \dots, X_p$  all are continuous random variables

We shall concentrate on continuous random vectors

Each nice  $\mathbf{X}$  has a prob. density function  $f$

Three important properties of the p.d.f.:

1.  $f(\mathbf{x}) \geq 0$  for all  $\mathbf{x} \in \mathbb{R}^p$
2. The total area below the graph of  $f$  is 1:

$$\int_{\mathbb{R}^p} f(\mathbf{x}) d\mathbf{x} = 1$$

3. For all  $t_1, \dots, t_p$ ,

$$P(X_1 \leq t_1, \dots, X_p \leq t_p) = \int_{-\infty}^{t_1} \cdots \int_{-\infty}^{t_p} f(\mathbf{x}) d\mathbf{x}$$

**Reminder:** “Expected value,” an average over the *entire* population

The *mean vector*:

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_p \end{bmatrix}$$

where

$$\mu_i = E(X_i) = \int_{\mathbb{R}^p} x_i f(\mathbf{x}) d\mathbf{x}$$

is the mean of the  $i$ th component of  $\mathbf{X}$

The *covariance* between  $X_i$  and  $X_j$ :

$$\begin{aligned} \sigma_{ij} &= E(X_i - \mu_i)(X_j - \mu_j) \\ &= E(X_i X_j) - \mu_i \mu_j \end{aligned}$$

The *variance* of each  $X_i$ :

$$\sigma_{ii} = E(X_i - \mu_i)^2 = E(X_i^2) - \mu_i^2$$

The *covariance matrix* of  $\mathbf{X}$ :

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1p} \\ \sigma_{21} & \cdots & \sigma_{2p} \\ \vdots & & \vdots \\ \sigma_{p1} & \cdots & \sigma_{pp} \end{bmatrix}$$

An easy result:

$$\boldsymbol{\Sigma} = E(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^T$$

Also,

$$\boldsymbol{\Sigma} = E(\mathbf{X}\mathbf{X}^T) - \boldsymbol{\mu}\boldsymbol{\mu}^T$$

To avoid pathological cases, we assume that  $\boldsymbol{\Sigma}$  is nonsingular

Theory *vs.* PracticePopulation *vs.* Random Sample

All galaxies of COMBO-17 type	A sample from the COMBO-17 data set
Random vector $\mathbf{X}$	Random sample $\mathbf{x}_1, \dots, \mathbf{x}_n$
Population Mean $\boldsymbol{\mu} = E(\mathbf{X})$	Sample mean $\bar{\mathbf{x}} = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k$
Popn. cov. matrix $\boldsymbol{\Sigma} =$ $E(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^T$	Sample cov. matrix, $S = \frac{1}{n-1}$ $\times \sum (\mathbf{x}_k - \bar{\mathbf{x}})(\mathbf{x}_k - \bar{\mathbf{x}})^T$

Laws of Large Numbers: In a technical sense,  $\bar{\mathbf{x}} \rightarrow \boldsymbol{\mu}$  and  $S \rightarrow \boldsymbol{\Sigma}$  as  $n \rightarrow \infty$

$\mathbf{X} = [X_1, \dots, X_p]^T$ : A random vector whose possible values range over all of  $\mathbb{R}^p$   
 $\mathbf{X}$  has a *multivariate normal distribution* if has a probability density function of the form

$$f(\mathbf{x}) = \text{const.} \times \exp \left[ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \right]$$

where

$$\text{const.} = \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}|^{1/2}}$$

Standard notation:  $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

Special case,  $p = 1$ : Let  $\boldsymbol{\Sigma} = \sigma^2$ ; then

$$f(x) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp \left[ -\frac{1}{2} \left( \frac{x - \mu}{\sigma} \right)^2 \right]$$

Special case,  $\boldsymbol{\Sigma}$  diagonal:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \sigma_p^2 \end{bmatrix}$$

$$|\Sigma| = \sigma_1^2 \sigma_2^2 \cdots \sigma_p^2$$

$$\Sigma^{-1} = \begin{bmatrix} \sigma_1^{-2} & 0 & \cdots & 0 \\ 0 & \sigma_2^{-2} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \sigma_p^{-2} \end{bmatrix}$$

$$f(\mathbf{x}) = \prod_{j=1}^p \frac{1}{(2\pi\sigma_j^2)^{1/2}} \exp \left[ -\frac{1}{2} \left( \frac{x_j - \mu_j}{\sigma_j} \right)^2 \right]$$

Conclusion:  $X_1, \dots, X_p$  are mutually independent and normally distributed

---

Recall:  $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \Sigma)$  if its p.d.f. is of the form

$$f(\mathbf{x}) = \text{const.} \times \exp \left[ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})' \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]$$

where

$$\text{const.} = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}}$$

Facts:

$$\boldsymbol{\mu} = E(\mathbf{X}),$$

$$\Sigma = \text{Cov}(\mathbf{X})$$

$$\int_{\mathbb{R}^p} f(\mathbf{x}) d\mathbf{x} = 1$$


---

If  $A$  is a  $k \times p$  matrix then

$$A\mathbf{X} + \mathbf{b} \sim N_k(A\boldsymbol{\mu} + \mathbf{b}, A\Sigma A^T)$$

Proof: Use Fourier transforms

Special cases:

$\mathbf{b} = \mathbf{0}$  and  $A = \mathbf{v}^T$  where  $\mathbf{v} \neq \mathbf{0}$ :

$$\mathbf{v}^T \mathbf{X} \sim N(\mathbf{v}^T \boldsymbol{\mu}, \mathbf{v}^T \Sigma \mathbf{v})$$

Note:  $\mathbf{v}^T \Sigma \mathbf{v} > 0$  since  $\Sigma$  is positive definite

$\mathbf{v} = [1, 0, \dots, 0]^T$ :  $X_1 \sim N(\mu_1, \sigma_{11})$

Similar argument: Each  $X_i \sim N(\mu_i, \sigma_{ii})$

---

Decompose  $\mathbf{X}$  into two subsets,  $\mathbf{X} = \begin{bmatrix} \mathbf{X}_u \\ \mathbf{X}_l \end{bmatrix}$

Similarly, decompose

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_u \\ \boldsymbol{\mu}_l \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{uu} & \boldsymbol{\Sigma}_{ul} \\ \boldsymbol{\Sigma}_{lu} & \boldsymbol{\Sigma}_{ll} \end{bmatrix}$$

Then

$$\begin{aligned} \boldsymbol{\mu}_u &= E(\mathbf{X}_u), & \boldsymbol{\mu}_l &= E(\mathbf{X}_l) \\ \boldsymbol{\Sigma}_{uu} &= \text{Cov}(\mathbf{X}_u), & \boldsymbol{\Sigma}_{ll} &= \text{Cov}(\mathbf{X}_l) \\ \boldsymbol{\Sigma}_{ul} &= \text{Cov}(\mathbf{X}_u, \mathbf{X}_l) \end{aligned}$$

The marginal distribution of  $\mathbf{X}_u$ :

$$\mathbf{X}_u \sim N_u(\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_{uu})$$

The conditional distribution of  $\mathbf{X}_u|\mathbf{X}_l$ :

$$\mathbf{X}_u|\mathbf{X}_l \sim N_u(\dots, \dots)$$

If  $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  then  $\mathbf{v}^T \mathbf{X}$  has a 1-D normal distribution for every vector  $\mathbf{v} \in \mathbb{R}^p$   
Conversely, if  $\mathbf{v}^T \mathbf{X}$  has a 1-D normal distribution for every  $\mathbf{v}$  then  $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

Proof: Fourier transforms again

(The assumption that an  $\mathbf{X}$  is normally distributed is very strong)

Let us use this result to construct an exploratory test of whether some COMBO-17 variables have a multivariate normal distribution

Choose several COMBO-17 variables, e.g.,

Rmag, mumax, Mcz, MCzml, chi2red, UjMAG,  
BjMAG, VjMAG

Use  $R$  to generate a “random” vector  $\mathbf{v} = [v_1, v_2, \dots, v_8]^T$

For each galaxy, calculate

$$v_1 * \text{Rmag} + v_2 * \text{mumax} + \dots + v_8 * \text{VjMAG}$$

This produces 3,462 such numbers ( $\mathbf{v}$ -scores)

Construct a Q-Q plot of all these  $\mathbf{v}$ -scores against the standard normal distribution

Study the plot to see if normality seems plausible

Repeat the exercise with a new random  $\mathbf{v}$

Repeat the exercise  $10^3$  times

Note: We need only those vectors for which  $v_1^2 + \dots + v_8^2 = 1$  (why?)

---

Mardia's test for multivariate normality

If the data contain a substantial number of outliers then it goes against the hypothesis of multivariate normality

If one COMBO-17 variable is not normally distributed then the full set of variables does not have a multivariate normal distribution

In that case, we can try to transform the original variables to produce new variables which are normally distributed

Example: Box-Cox transformations, log transformations (a special case of Box-Cox)

For data sets arising from a multivariate normal distribution, we can perform accurate inference for the mean vector and covariance matrix

---

Variables (random vector):  $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

The parameters  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  are unknown

Data (measurements):  $\mathbf{x}_1, \dots, \mathbf{x}_n$

Problem: Estimate  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$

$\bar{\mathbf{x}}$  is an unbiased and consistent estimator of  $\boldsymbol{\mu}$

$\bar{\mathbf{x}}$  is the MLE of  $\boldsymbol{\mu}$

The MLE of  $\boldsymbol{\Sigma}$  is  $\frac{n-1}{n}S$ ; this is not unbiased

The sample covariance matrix,  $S$ , is an unbiased estimator of  $\boldsymbol{\Sigma}$

Since  $S$  is close to being the MLE of  $\boldsymbol{\Sigma}$ , we estimate  $\boldsymbol{\Sigma}$  using  $S$

---

Naive method: Using only the data on the  $i$ th variable, construct a confidence interval for each  $\mu_i$

Use the collection of confidence intervals as a confidence region for  $\boldsymbol{\mu}$

Good news: This can be done using elementary statistical methods

Bad news: A collection of 95% confidence intervals, one for each  $\mu_i$ , does not result in a 95% confidence region for  $\boldsymbol{\mu}$

Starting with individual intervals with lower confidence levels, we can achieve an overall 95% confidence level for the combined region

Bonferroni inequalities: Some difficult math formulas are needed to accomplish that goal

---

Worse news: The resulting confidence region for  $\boldsymbol{\mu}$  is a rectangle  
This is not consonant with a density function of the form

$$f(\mathbf{x}) = \text{const.} \times \exp \left[ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \right]$$

The contours of the graph of  $f(\mathbf{x})$  are ellipsoids, so we should derive an ellipsoidal confidence region for  $\boldsymbol{\mu}$

Fact: Every positive definite symmetric matrix has a unique positive definite symmetric square root

$\boldsymbol{\Sigma}^{-1/2}$ : The p.d. square-root of  $\boldsymbol{\Sigma}^{-1}$

Recall (see p. 31): If  $A$  is a  $p \times p$  nonsingular matrix and  $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  then

$$A\mathbf{X} + \mathbf{b} \sim N_p(A\boldsymbol{\mu} + \mathbf{b}, A\boldsymbol{\Sigma}A^T)$$

Set  $A = \boldsymbol{\Sigma}^{-1/2}$ ,  $\mathbf{b} = -\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu}$

Then  $A\boldsymbol{\mu} + \mathbf{b} = \mathbf{0}$ ,  $A\boldsymbol{\Sigma}A^T = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1/2} = I_p$

$$\boldsymbol{\Sigma}^{-1/2}(\mathbf{X} - \boldsymbol{\mu}) \sim N_p(\mathbf{0}, I_p)$$

$I_p = \text{diag}(1, 1, \dots, 1)$ , a diagonal matrix

## Methods of Multivariate Analysis

Reduce the number of variables

Structural simplification

Linear functions of variables (Principal Components)

Investigate the dependence between variables      Canonical correlations

Statistical inference

Estimation

Confidence regions

Hypothesis testing

Classify or cluster “similar” objects

Discriminant analysis

Cluster analysis

Predict

Multiple Regression

Multivariate regression

Can we reduce the dimension of the problem?

$\mathbf{X}$ : A  $p$ -dimensional random vector

Covariance matrix:  $\Sigma$

Solve for  $\lambda$ :  $|\Sigma - \lambda I| = 0$

Solutions:  $\lambda_1, \dots, \lambda_p$ , the *eigenvalues* of  $\Sigma$

Assume, for simplicity, that  $\lambda_1 > \dots > \lambda_p$

Solve for  $\mathbf{v}$ :  $\Sigma \mathbf{v} = \lambda_j \mathbf{v}$ ,  $j = 1, \dots, p$

Solution:  $\mathbf{v}_1, \dots, \mathbf{v}_p$ , the *eigenvectors* of  $\Sigma$

Scale each eigenvector to make its length 1

$\mathbf{v}_1, \dots, \mathbf{v}_p$  are orthogonal

The linear combination  $\mathbf{v}^T \mathbf{X}$  such that

(i)  $\text{Var}(\mathbf{v}^T \mathbf{X})$  is maximal, and

(ii)  $\mathbf{v}^T \mathbf{v} = 1$

Maximize  $\text{Var}(\mathbf{v}^T \mathbf{X}) = \mathbf{v}^T \Sigma \mathbf{v}$  subject to  $\mathbf{v}^T \mathbf{v} = 1$

Lagrange multipliers

Solution:  $\mathbf{v} = \mathbf{v}_1$ , the first eigenvector of  $\Sigma$

$\mathbf{v}_1^T \mathbf{X}$  is the first principal component

The second PC: The linear combination  $\mathbf{v}^T \mathbf{X}$  such that

(i)  $\text{Var}(\mathbf{v}^T \mathbf{X})$  is maximal,

(ii)  $\mathbf{v}^T \mathbf{v} = 1$ , and

(iii)  $\mathbf{v}^T \mathbf{X}$  has zero correlation with the first PC

Maximize  $\text{Var}(\mathbf{v}^T \mathbf{X}) = \mathbf{v}^T \Sigma \mathbf{v}$  with  $\mathbf{v}^T \mathbf{v} = 1$  and  $\text{Cov}(\mathbf{v}^T \mathbf{X}, \mathbf{v}_1^T \mathbf{X}) \equiv \mathbf{v}^T \Sigma \mathbf{v}_1 = 0$

Lagrange multipliers

Solution:  $\mathbf{v} = \mathbf{v}_2$ , the second eigenvector of  $\Sigma$

The  $k$ th PC: The linear combination  $\mathbf{v}^T \mathbf{X}$  such that

(i)  $\text{Var}(\mathbf{v}^T \mathbf{X})$  is maximal,

(ii)  $\mathbf{v}^T \mathbf{v} = 1$ , and

(iii)  $\mathbf{v}^T \mathbf{X}$  has zero correlation with all prior PCs

Solution:  $\mathbf{v} = \mathbf{v}_k$ , the  $k$ th eigenvector of  $\Sigma$

The PCs are random variables

Simple matrix algebra:  $\text{Var}(\mathbf{v}_k^T \mathbf{X}) = \lambda_k$

$p$ -dimensional data:  $\mathbf{x}_1, \dots, \mathbf{x}_n$

$S$ : the sample covariance matrix  
 $\tilde{\lambda}_1 > \dots > \tilde{\lambda}_p$ : The eigenvalues of  $S$   
 Remarkable result:

$$\tilde{\lambda}_1 + \dots + \tilde{\lambda}_p = s_{11} + \dots + s_{pp}$$

$\tilde{\mathbf{v}}_1, \dots, \tilde{\mathbf{v}}_p$ : The corresponding eigenvectors  
 $\tilde{\mathbf{v}}_1 \mathbf{X}, \dots, \tilde{\mathbf{v}}_p \mathbf{X}$ : The sample PCs  
 $\tilde{\lambda}_1, \dots, \tilde{\lambda}_p$ : The estimated variances of the PCs  
 Basic idea: Use the sample PCs instead of  $\mathbf{X}$  to analyze the data

Example: (Johnson and Wichern)

$$S = \begin{bmatrix} 4.31 & 1.68 & 1.80 & 2.16 & -.25 \\ 1.68 & 1.77 & .59 & .18 & .17 \\ 1.80 & .59 & .80 & 1.07 & -.16 \\ 2.16 & .18 & 1.07 & 1.97 & -.36 \\ -.25 & .17 & -.16 & -.36 & .50 \end{bmatrix}$$

The sample principal components:

$$Y_1 = .8X_1 + .3X_2 + .3X_3 + .4X_4 - .1X_5$$

$$Y_2 = -.1X_1 - .8X_2 + .1X_3 + .6X_4 - .3X_5$$

etc.

$$\tilde{\lambda}_1 = 6.9, \tilde{\lambda}_2 = 0.8, \dots; \tilde{\lambda}_1 + \dots + \tilde{\lambda}_5 = 8.4$$

$$X_1: \text{Rmag}$$

$$X_2: \text{mumax}$$

etc.

The PCs usually have no physical meaning, but they can provide insight into the data analysis

$\tilde{\lambda}_1 + \dots + \tilde{\lambda}_p$ : A measure of total variability of the data

$\frac{\tilde{\lambda}_k}{\tilde{\lambda}_1 + \dots + \tilde{\lambda}_p}$ : The proportion of total variability of the data “explained” by the  $k$ th PC

How many PC’s should we calculate?

Stop when

$$\frac{\tilde{\lambda}_1 + \dots + \tilde{\lambda}_k}{\tilde{\lambda}_1 + \dots + \tilde{\lambda}_p} \geq 0.9$$

*Scree plot*: Plot the points  $(1, \tilde{\lambda}_1), \dots, (p, \tilde{\lambda}_p)$  and connect them by a straight line. Stop when the graph has flattened.

*Other rule*: Kaiser’s rule; rules based on tests of hypotheses, ...

---

Some feel that PC's should be calculated from correlation matrices, not covariance matrices

Argument for correlation matrices: If the original data are rescaled then the PCs and the  $\tilde{\lambda}_k$  all change

Argument against: If some components have significantly smaller means and variances than others then correlation-based PCs will give all components similarly-sized weights

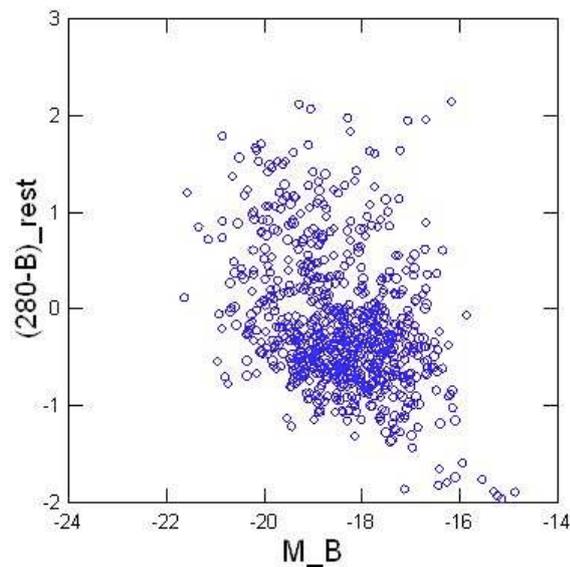
---

### COMBO-17 data:

Two classes of galaxies, redder and bluer, but with overlapping distributions

Dataset: galaxy brightnesses in 17 bands—detailed view of "red" and "blue" for each galaxy

The following figure of  $M_B$  (BjMag) vs (280-B) (S280MAG-BjMag) for restricted range 0.7-0.9 of  $z$  (McZ) shows two cluster ("blue" below and "red" above), similar to the one in the website (also Wolf et al., 2004)




---

We investigate the relationship of these colors to the brightness variables by multivariate analysis.

From combo17 dataset collected the even-numbered columns (30, 32, ..., 54).

Normalized each to (say) the value in column 40 (W640FE) for each galaxy.

These are called “colors”.

Removed variable W640FE from the dataset

We added to this dataset Bmag ( $M_B$ ). Also kept Mcz.

Modified “W” variables have been renamed with an “R” in the beginning.

Table 1 of Wolf et al. (2005, <http://arxiv.org/pdf/astro-ph/0506150v2>)  
mean locations in multidimensional parameter space for “dust-free old” (= “red”) and “blue cloud” (= “blue”) galaxies

red galaxies has a mean value of  $(U - V) = 1.372$

blue galaxies has a mean  $(U - V) = 0.670$ —which are widely separated values

redshift  $z$  is a scientifically (very!) interesting variable denoting age of galaxy

We classify as “red” if  $(U - V) > 0.355$  and as “blue” if  $(U - V) \leq 0.355$ —color variable

This is the dataset. Data for the first few galaxies with the first few “R” readings:

RW420FE	RW462FE	RW485FD	RW518FE	RW571FS	RW604FE	BJMAG	MCZ	U-V	COLOR
-0.018	-0.006	0.000	-0.001	-0.004	-0.002	-17.540	0.832	0.090	1
-0.003	0.002	-0.000	-0.002	0.007	0.006	17.860	0.927	-0.080	1
-0.010	-0.003	0.002	-0.007	-0.000	0.000	-19.910	1.202	0.660	2
0.006	0.010	-0.005	-0.004	-0.005	0.003	-17.390	0.912	-0.160	1
0.002	0.005	0.002	0.010	0.004	0.007	-18.400	0.848	1.680	2
0.004	0.004	0.005	0.002	0.005	0.005	-18.110	0.882	-0.520	1
-0.004	-0.009	-0.008	-0.011	-0.008	-0.011	-16.060	0.896	-0.860	1
-0.002	-0.005	-0.006	-0.000	-0.004	0.002	-16.490	0.930	0.140	1
0.018	0.017	0.008	0.020	0.011	0.015	-17.680	0.774	0.200	1
0.006	0.007	0.001	-0.004	-0.004	-0.000	-11.150	0.062	-0.310	1
-0.009	-0.007	-0.010	-0.009	-0.009	-0.008	-16.990	0.874	-0.030	1
-0.032	-0.021	-0.018	-0.024	-0.019	-0.020	-13.990	0.173	0.650	2
-0.015	-0.009	-0.013	-0.006	-0.013	-0.014	-18.490	1.109	0.190	1
0.002	-0.002	0.002	0.002	0.012	0.002	-10.300	0.143	0.870	2
-0.028	-0.023	-0.020	-0.020	-0.025	-0.017	-18.210	0.626	1.360	2
0.011	0.015	-0.002	-0.003	0.002	0.009	-20.140	1.185	-0.200	1
0.010	0.007	0.012	0.010	0.010	0.015	-16.130	0.921	-1.570	1
0.001	0.004	0.004	0.001	0.002	0.003	-19.420	0.832	-0.170	1
0.005	0.013	-0.002	0.008	0.007	0.007	-18.110	0.793	0.460	2
-0.007	-0.002	-0.009	-0.002	0.000	-0.008	-17.810	0.952	0.290	1
-0.004	-0.004	-0.007	-0.009	-0.007	-0.002	-17.590	0.921	0.770	2
-0.007	-0.008	-0.014	-0.004	-0.003	-0.002	-16.980	0.986	-0.080	1
0.008	-0.004	0.003	-0.001	-0.001	0.007	-18.170	1.044	2.500	2
-0.000	0.002	0.004	0.000	0.004	0.001	-18.680	0.929	1.580	2
0.002	0.003	0.008	-0.003	0.001	0.001	-17.480	0.901	0.030	1
0.020	0.013	0.009	0.009	0.018	0.026	-16.600	0.484	0.190	1

---

0.016	0.008	0.019	0.019	0.014	0.010	-16.390	0.763	-0.730	1
0.001	0.001	0.006	0.004	0.003	0.002	-17.210	0.711	0.760	2
0.003	-0.001	-0.008	0.004	0.002	-0.001	-18.950	1.044	0.270	1
0.007	0.007	0.006	0.008	0.007	0.011	-19.850	0.826	0.670	2
-0.030	-0.013	-0.017	-0.001	0.021	0.025	-17.640	0.340	0.680	2
-0.058	-0.031	-0.037	-0.026	-0.015	-0.012	-17.600	0.365	0.390	2
0.004	0.006	0.008	0.013	0.018	0.021	-20.040	0.898	0.080	1
-0.005	-0.004	-0.006	-0.006	0.001	0.005	-19.540	0.878	0.290	1
-0.009	0.003	-0.009	-0.006	0.001	-0.007	-12.970	0.082	0.510	2

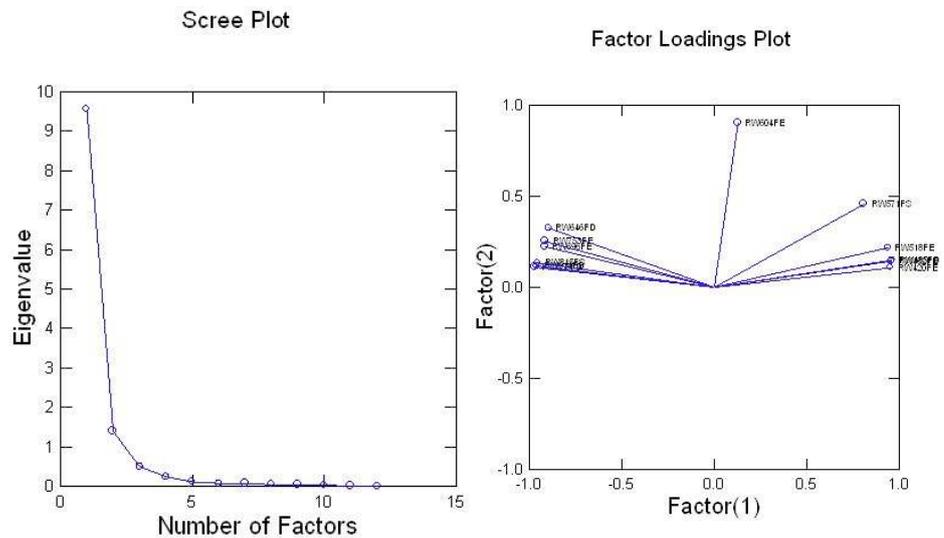
---

**PCA of Combo17 data:**

PCA of the 12 color variables RW420FE RW462FE .... RW856FD RW914FD  
 The scree plot suggests that two components are adequate.

Variable	PC1 weight	PC2 weight
RW420FE	0.954	0.107
RW462FE	0.957	0.144
RW485FD	0.960	0.149
RW518FE	0.938	0.218
RW571FS	0.810	0.456
RW604FE	0.128	0.902
RW646FD	-0.897	0.326
RW696FE	-0.914	0.223
RW753FE	-0.913	0.252
RW815FS	-0.953	0.134
RW856FD	-0.970	0.110
RW914FD	-0.961	0.117
Variance explained	9.547	1.386
% Variance explained	79.555	11.553

---



Two components explain most of the variation (about 91%)

#### Interpretation:

##### Principal Component 1:

Weights are nearly the same in magnitude (except for RW604FE—insignificant)  
RW4... and RW5... vs RW6... RW7.. RW8.. RW9..

##### Principal Component 2:

RW604E the main component

Rest are nearly equal and small

Two components complement each other

Plot of PC scores of galaxies can be used for classification

Will see this in the Cluster Analysis chapter

#### Classification Methods:

Two distinct types of classification problems—unsupervised and supervised

Unsupervised classification: Cluster Analysis:

to find groups in the data objects

objects within a group are similar

Example: what kinds of celestial objects are there— stars, planets, moons, asteroids, galaxies, comets, etc.

Multivariate (qualitative and quantitative) data on objects used

Characterize each type of object by these variables

Example: C. Wolf, M. E. Gray, and K. Meisenheimer (2008): Red-sequence galaxies with young stars and dust: The cluster Abell 901/902 seen with COMBO-17. *Astronomy & Astrophysics* classify galaxies into three classes with properties in the following table

by cluster analysis

Property	Dust-free old	Dusty red-seq	Blue cloud
$N_{\text{galaxy}}$	294	168	333
$N_{\text{fieldcontamination}}$	6	7	49
$N_{\text{spectra}}$	144	69	36
$z_{\text{spec}}$	0.1646	0.1646	0.1658
$\sigma_{cz}/(1+z)/(\text{km/s})$	939	1181	926
$z_{\text{spec,N}}$	0.1625	0.1615	N/A
$z_{\text{spec,S}}$	0.1679	0.1686	N/A
$\sigma_{cz,N}/(1+z)/(\text{km/s})$	589	597	N/A
$\sigma_{cz,S}/(1+z)/(\text{km/s})$	522	546	N/A
$\log(\Sigma_{10}(\text{Mpc/h})^2)$	2.188	1.991	1.999
$EW_e(OII)/\overset{\circ}{\text{A}}$	N/A	$4.2 \pm 0.4$	$17.5 \pm 1.5$
$EW_a(H\delta)/\overset{\circ}{\text{A}}$	$2.3 \pm 0.5$	$2.6 \pm 0.5$	$4.5 \pm 1.0$
age/Gyr	6.2	3.5	1.2
$E_{B-V}$	0.044	0.212	0.193
$(U - V)_{\text{rest}}$	1.372	1.293	0.670
$M_{V,\text{rest}}$	-19.31	-19.18	-18.47
B - R	1.918	1.847	1.303
V - I	1.701	1.780	1.290
R - I	0.870	0.920	0.680
U - 420	0.033	-0.079	-0.377
420 - 464	0.537	0.602	0.560
464 - 518	0.954	0.827	0.490
604 - 646	0.356	0.339	0.238
753 - 815	0.261	0.274	0.224

Mean properties of the three galaxy SED class samples.

### Supervised Learning or Discriminant Analysis

Know that there are these three types of galaxies

Have **Training Samples** where an expert (supervisor) classifies units in the sample

Multivariate observations on the sample units available

A new object is seen on which multivariate observations made

Problem: Classify it in one or other of the groups

In discriminant Analysis we develop a formula for such classification

Formula arrived at by performing discriminant analysis of training data

Some assumptions are often made

Multivariate normality in each group with a common covariance matrix

Find a classification rule that minimizes misclassification

This leads to **Linear Discriminant Function**, a linear combination of observed variables

### Discriminant Analysis Example

Use “R” data to develop a formula for classification into color 1 or 2

The linear discriminant function is

$$0.345 + RW420FE*14.277 - RW462FE*0.844 \\ - RW485FD*36.890 + RW518FE*6.541 \\ + RW571FS*2.249 + RW604FE*25.670 \\ + RW646FD*18.331 + RW696FE*15.123 - RW753FE*29.072 \\ - RW815FS*16.970 - RW856FD*16.467 + RW914FD*2.024$$

If this value is  $> 0$  we classify a galaxy as 1 (red); else 2 (blue)

Using the formula on the training sample, we get an idea of the performance of the classification rule as follows:

Actual Group	Classified Group		%correct
	1	2	
1	2,111	45	98
2	1,020	286	22
Total	3,131	331	69

This is not a very good classification rule—the chosen variables do not provide adequate separation between blue and red

### Multiple Regression

If a supervisor had used the value of  $U - V$  to classify the galaxies into red and blue, and if values of  $U - V$  are indeed available, then why not use them rather than the red-blue classification?

$U - V$  data rather than color data in training sample

Leads to Multiple Regression Analysis



## Chapter 20

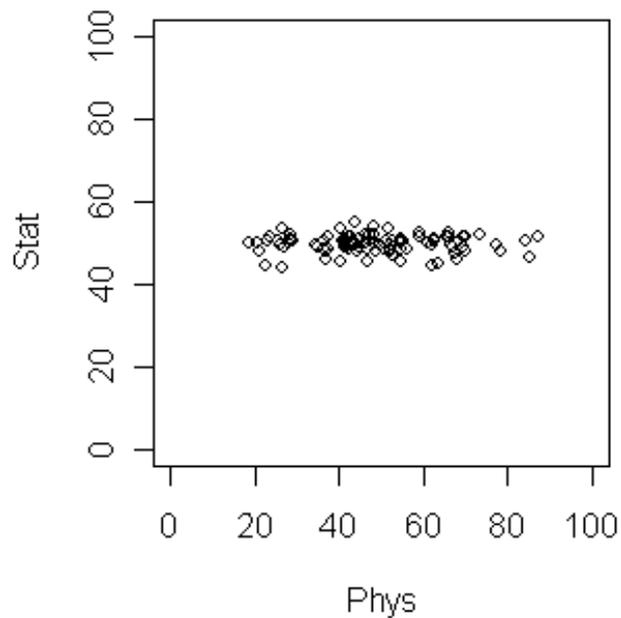
# PRINCIPAL COMPONENT ANALYSIS IN R

*Notes by Arnab Chakraborty*

# Principal Component Analysis

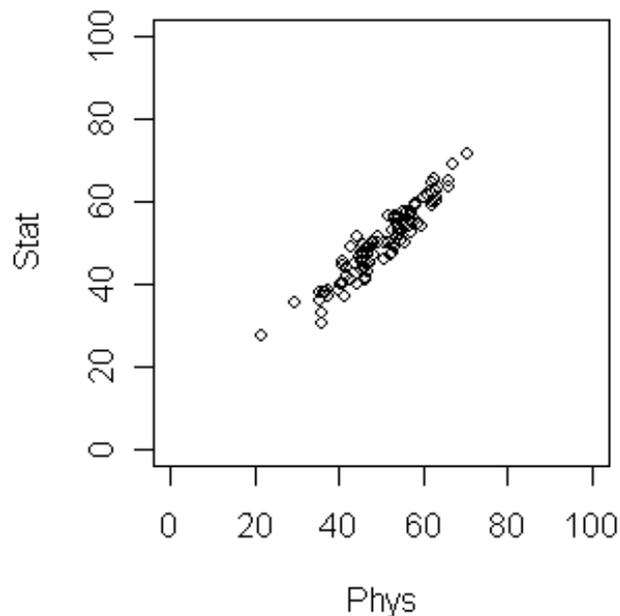
## A simple example

Consider 100 students with Physics and Statistics grades shown in the diagram below. The data set is in [marks.dat](#).



If we want to compare among the students which grade should be a better discriminating factor? Physics or Statistics? Surely Physics, since the variation is larger there. This is a common situation in data analysis where the direction along which the data *varies the most* is of special importance.

Now suppose that the plot looks like the following. What is the best way to compare the students now?

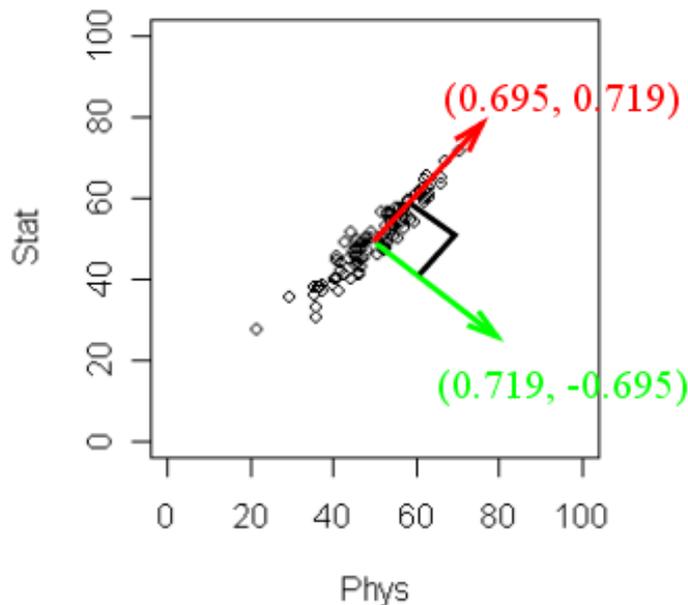


Here the direction of maximum variation is like a slanted straight line. This means we should take linear combination of the two grades to get the best result. In this simple data set the direction of maximum variation is more or less clear. But for many data sets (especially high dimensional ones) such visual inspection is not adequate or even possible! So we need an objective method to find such a direction. **Principal Component Analysis (PCA)** is one way to do this.

```
dat = read.table("marks.dat", head=T)
dim(dat)
names(dat)
pc = princomp(~Stat+Phys, dat)
pc$loading
```

Notice the somewhat non-intuitive syntax of the `princomp` function. The first argument is a so-called formula object in R (we have encountered this beast in the regression tutorial). In `princomp` the first argument must start with a `~` followed by a list of the variables (separated by plus signs).

The output may not be readily obvious. The next diagram will help.



R has returned two **principal components**. (Two because we have two variables). These are a unit vector at right angles to each other. You may think of PCA as choosing a new coordinate system for the data, the principal components being the unit vectors along the axes. The first principal component gives the direction of the maximum spread of the data. The second gives the direction of maximum spread perpendicular to the first direction. These two directions are packed inside the matrix `pc$loadings`. Each column gives a direction. The direction of maximum spread (the first principal component) is in the first column, the next principal component in the second and so on.

There is yet more information. Type

```
| pc
```

to learn the amount of spread of the data along the chosen directions. Here the spread along the first direction is 12.40, while that along the second is much smaller 1.98. These numbers are often not of main importance, it is their relative magnitude that matters.

To see all the stuff that is neatly tucked inside `pc` we can type

```
| names (pc)
```

We shall not go into all these here. But one thing deserves mention: `scores`. These are the projections of the data points

along the principal components. We have already mentioned that the principal components may be viewed as a new reference frame. Then the scores are the coordinates of the points w.r.t. this frame.

`pc$scores`

## Higher dimensions

Most statisticians consider PCA a tool for reducing dimension of data. To see this consider the interactive 3D scatterplot below. It is possible to rotate this plot with the mouse. By rotating suitably we can see that the cloud of points is basically confined in a 2D plane. In other words, the data set is essentially 2D.

### Drag the picture with the mouse

The same conclusion may be obtained by PCA. Here the first two components will be along the plane, while the third will be perpendicular to the plane. These are shown as the three lines.

## Putting it to action

Now that we have seen how PCA can identify if the data cloud

resides in a lower dimensional space, we are ready to apply our knowledge to astronomy. We shall work with the SDSS Quasar data set stored in the file [SDSS\\_quasar.dat](#). First we prepare the data set for analysis.

```
quas = read.table("SDSS_quasar.dat", head=T)
dim(quas)
names(quas)
quas = na.omit(quas)
dim(quas)
```

Now we shall apply PCA.

```
pc = princomp(quas[, -1], scores=T)
```

The `scores=T` option will automatically compute the projections of the data along the principal component directions.

Before looking inside `pc` let us make a mental note of what information we should be looking for. We should look for the loadings (which are 22 mutually perpendicular directions in a 22-dimensional space). Each direction is represented by a unit vector with 22 components. So we have 22 times 22 = 484 numbers! Whew! We should also know the spread of the data cloud along each of the 22 directions. The spread along each direction is given by just a single number. So we have to just look at 22 numbers for this (lot less than 484). So we shall start by looking for these 22 numbers first. Type

```
pc
```

to see them. Well, some of these are much larger than the rest. To get an idea of the relative magnitudes, let us plot them.

```
plot(pc)
```

Incidentally, this plot is often called a **screeplot**, and R has a function with that name (it produces the same output as the `plot` command).

```
screepplot(pc)
```

By default, the spreads are shown as bars. Most textbooks, however, prefer to depict the same information as a line diagram:

```
screepplot(pc, type="lines")
```

The term ``scree" refers to pebbles lying around the base of a cliff, and a screeplot drawn with lines makes the analogy clear. But one should not forget that the lines do not have any significance. The points are the only important things.

We can see from the screeplot that only the first 2 components account for the bulk. In other words, the 22-dimensional data cloud essentially resides in just a 2D plane! Which is that plane? The answer lies in the first two columns of `pc$loadings`.

```
|pc$loading[,1:2]
```

These give two mutually perpendicular unit vectors defining the plane. To make sure that this is indeed the case you may check as

```
|M = pc$loading[,1:2]
|t(M) %% M #should ideally produce the 2 by 2 identity matrix
```

You might like to project the entire data set onto this plane.

```
|plot(pc$scores[,1],pc$scores[,2],pch=".")
```

This is how the data cloud looks like in that magic plane in 22-dimensional space. And with my limited astronomy knowledge I have no idea why these look like this!!! (An astronomy student had once told me that this pattern has to do something with the way the survey was conducted in outer space.)

## Peeping behind the scree...

It may be of some interest to know how PCA works. We shall not go into all the nitty gritty details, but the basic idea is not too hard to grasp, if you know eigen values and eigen vectors.

What PCA does is, roughly speaking, computing the eigen values and eigen vectors of the covariance matrix of the data. A detailed exposition of why that is done is beyond the scope of this tutorial. But we may use R's eigen analysis tools to hack a rough imitation of `princomp` ourselves. We shall take the students' grades data as our running example.

First compute the covariance matrix.

```
|dat = read.table("marks.dat",head=T)
|covar = cov(dat)
```

Next compute the eigen values and vectors:

```
|eig = eigen(covar)
```

Now compare:

```
|val = eig$values
|sqrt(val)
|pc = princomp(~Stat+Phys, dat)
|pc
```

The values will not match *exactly* as there are more bells and whistles inside `princomp` than I have cared to divulge here. But still the results are comparable.

### A better way

The `princomp` function is numerically less stable than `prcomp` which is a better alternative. However, its output structure differs somewhat from that for `princomp`.

**Exercise:** Read the help on `prcomp` and redo the above computation on the SDSS quasar data using this function.

[\[Solution\]](#)

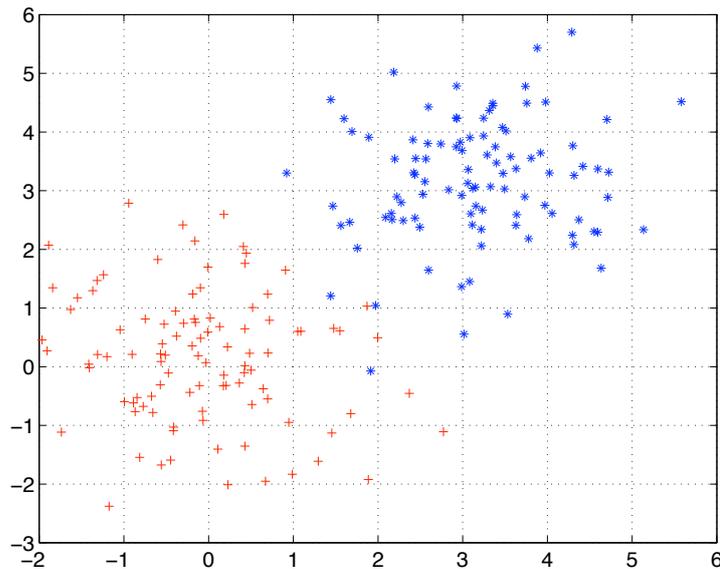
# Chapter 21

## CLUSTER ANALYSIS

*Notes by Thriyambakam Krishnan & Jia Li*

## Clustering

- A basic tool in data mining/pattern recognition:
  - Divide a set of data into groups.
  - Samples in one cluster are close and clusters are far apart.



- Motivations:
  - Discover classes of data in an unsupervised way (unsupervised learning).
  - Efficient representation of data: fast retrieval, data complexity reduction.
  - Various engineering purposes: tightly linked with pattern recognition.

---

## Approaches to Clustering

- Represent samples by feature vectors.
- Define a distance measure to assess the closeness between data.
- “Closeness” can be measured in many ways.
  - Define distance based on various norms.

- 
- For stars with measured parallax, the multivariate “distance” between stars is the spatial Euclidean distance. For a galaxy redshift survey, however, the multivariate “distance” depends on the Hubble constant which scales velocity to spatial distance. For many astronomical datasets, the variables have incompatible units and no prior known relationship. The result of clustering will depend on the arbitrary choice of variable scaling.
- 

## Approaches to Clustering

- Clustering: grouping of similar objects (unsupervised learning)
  - Approaches
    - Prototype methods:
      - \* K-means (for vectors)
      - \* K-center (for vectors)
      - \* D2-clustering (for bags of weighted vectors)
    - Statistical modeling
      - \* Mixture modeling by the EM algorithm
      - \* Modal clustering
    - Pairwise distance based partition:
      - \* Spectral graph partitioning
      - \* Dendrogram clustering (agglomerative): single linkage (friends of friends algorithm), complete linkage, etc.
- 

## Agglomerative Clustering

- Generate clusters in a hierarchical way.
- Let the data set be  $A = \{x_1, \dots, x_n\}$ .
- Start with  $n$  clusters, each containing one data point.
- Merge the two clusters with minimum pairwise distance.
- Update between-cluster distance.
- Iterate the merging procedure.

- The clustering procedure can be visualized by a tree structure called *dendrogram*.
- Definition for between-cluster distance?
  - For clusters containing only one data point, the between-cluster distance is the between-object distance.
  - For clusters containing multiple data points, the between-cluster distance is an agglomerative version of the between-object distances.
    - \* Examples: minimum or maximum between-objects distances for objects in the two clusters.
  - The agglomerative between-cluster distance can often be computed recursively.

## Principal Components Clustering

If the several dimensions can be satisfactorily reduced to two, by say Principal Components, then plotting the two component scores for each object will result in a picture which will help find clusters.

Combo17 Example:

This figure suggests one strong dense cluster and the remaining perhaps forming another dissipated one.

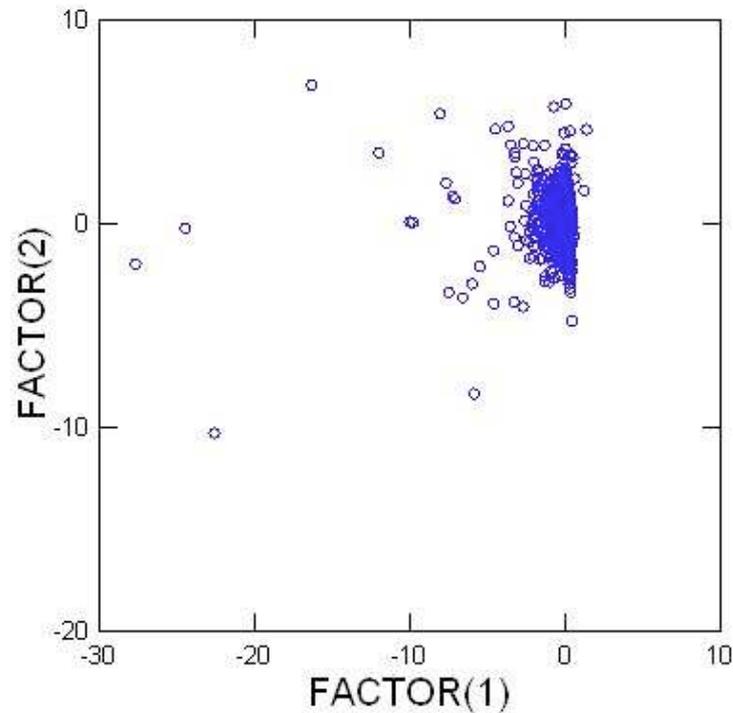
## K-means

- Assume there are  $M$  prototypes denoted by

$$\mathcal{Z} = \{z_1, z_2, \dots, z_M\}.$$

- Each training sample is assigned to one of the prototype. Denote the assignment function by  $A(\cdot)$ . Then  $A(x_i) = j$  means the  $i$ th training sample is assigned to the  $j$ th prototype.
- Goal: minimize the total mean squared error between the training samples and their representative prototypes, that is, the trace of the pooled within cluster covariance matrix.

$$\arg \min_{\mathcal{Z}, A} \sum_{i=1}^N \|x_i - z_{A(x_i)}\|^2$$



- Denote the objective function by

$$L(\mathcal{Z}, A) = \sum_{i=1}^N \|x_i - z_{A(x_i)}\|^2 .$$

- Intuition: training samples are tightly clustered around the prototypes. Hence, the prototypes serve as a compact representation for the training data.

## Necessary Conditions

- If  $\mathcal{Z}$  is fixed, the optimal assignment function  $A(\cdot)$  should follow the nearest neighbor rule, that is,

$$A(x_i) = \arg \min_{j \in \{1, 2, \dots, M\}} \|x_i - z_j\| .$$

- If  $A(\cdot)$  is fixed, the prototype  $z_j$  should be the average (centroid) of all the samples assigned to the  $j$ th prototype:

$$z_j = \frac{\sum_{i:A(x_i)=j} x_i}{N_j} ,$$

where  $N_j$  is the number of samples assigned to prototype  $j$ .

---

## The Algorithm

- Based on the necessary conditions, the k-means algorithm alternates the two steps:
    - For a fixed set of centroids (prototypes), optimize  $A(\cdot)$  by assigning each sample to its closest centroid using Euclidean distance.
    - Update the centroids by computing the average of all the samples assigned to it.
  - The algorithm converges since after each iteration, the objective function decreases (non-increasing).
  - Usually converges fast.
  - Stopping criterion: the ratio between the decrease and the objective function is below a threshold.
- 

## Example

- Training set:  $\{1.2, 5.6, 3.7, 0.6, 0.1, 2.6\}$ .
- Apply k-means algorithm with 2 centroids,  $\{z_1, z_2\}$ .
- Initialization: randomly pick  $z_1 = 2, z_2 = 5$ .

fixed	update
2	$\{1.2, 0.6, 0.1, 2.6\}$
5	$\{5.6, 3.7\}$
$\{1.2, 0.6, 0.1, 2.6\}$	1.125
$\{5.6, 3.7\}$	4.65
1.125	$\{1.2, 0.6, 0.1, 2.6\}$
4.65	$\{5.6, 3.7\}$

The two prototypes are:  $z_1 = 1.125, z_2 = 4.65$ . The objective function is  $L(\mathcal{Z}, A) = 5.3125$ .

---

- Initialization: randomly pick  $z_1 = 0.8$ ,  $z_2 = 3.8$ .

fixed	update
0.8	{1.2, 0.6, 0.1}
3.8	{5.6, 3.7, 2.6}
{1.2, 0.6, 0.1 }	0.633
{5.6, 3.7, 2.6 }	3.967
0.633	{1.2, 0.6, 0.1}
3.967	{5.6, 3.7, 2.6}

The two prototypes are:  $z_1 = 0.633$ ,  $z_2 = 3.967$ . The objective function is  $L(\mathcal{Z}, A) = 5.2133$ .

- Starting from different initial values, the k-means algorithm converges to different local optimum.
- It can be shown that  $\{z_1 = 0.633, z_2 = 3.967\}$  is the global optimal solution.

## Initialization

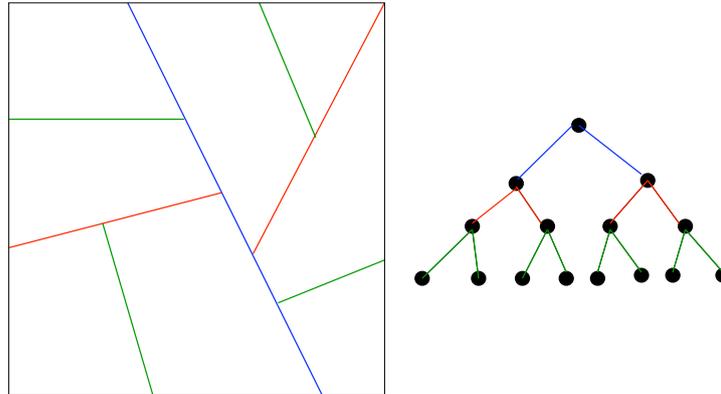
- Randomly pick up the prototypes to start the k-means iteration.
- Different initial prototypes may lead to different local optimal solutions given by k-means.
- Try different sets of initial prototypes, compare the objective function at the end to choose the best solution.
- When randomly select initial prototypes, better make sure no prototype is out of the range of the entire data set.
- Initialization in the above simulation:
  - Generated  $M$  random vectors with independent dimensions. For each dimension, the feature is uniformly distributed in  $[-1, 1]$ .
  - Linearly transform the  $j$ th feature,  $Z_j$ ,  $j = 1, 2, \dots, p$  in each prototype (a vector) by:  $Z_j s_j + m_j$ , where  $s_j$  is the sample standard deviation of dimension  $j$  and  $m_j$  is the sample mean of dimension  $j$ , both computed using the training data.

## Linde-Buzo-Gray (LBG) Algorithm

- An algorithm developed in vector quantization for the purpose of data compression.
- Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Communication*, Vol. COM-28, pp. 84-95, Jan. 1980.
- The algorithm
  1. Find the centroid  $z_1^{(1)}$  of the entire data set.
  2. Set  $k = 1, l = 1$ .
  3. If  $k < M$ , split the current centroids by adding small offsets.
    - If  $M - k \geq k$ , split all the centroids; otherwise, split only  $M - k$  of them.
    - Denote the number of centroids split by  $\tilde{k} = \min(k, M - k)$ .
    - For example, to split  $z_1^{(1)}$  into two centroids, let  $z_1^{(2)} = z_1^{(1)}, z_2^{(2)} = z_1^{(1)} + \epsilon$ , where  $\epsilon$  has a small norm and a random direction.
  4.  $k \leftarrow k + \tilde{k}; \quad l \leftarrow l + 1$ .
  5. Use  $\{z_1^{(l)}, z_2^{(l)}, \dots, z_k^{(l)}\}$  as initial prototypes. Apply k-means iteration to update these prototypes.
  6. If  $k < M$ , go back to step 3; otherwise, stop.

## Tree-structured Clustering

- Studied extensively in vector quantization from the perspective of data compression.
- Referred to as tree-structured vector quantization (TSVQ).
- The algorithm
  1. Apply 2 centroids k-means to the entire data set.
  2. The data are assigned to the 2 centroids.
  3. For the data assigned to each centroid, apply 2 centroids k-means to them separately.
  4. Repeat the above step.



- Compare with LBG:
  - For LBG, after the initial prototypes are formed by splitting, k-means is applied to the overall data set. The final result is  $M$  prototypes.
  - For TSVQ, data partitioned into different centroids at the same level will never affect each other in the future growth of the tree. The final result is a tree structure.
- Fast searching
  - For k-means, to decide which cell a query  $x$  goes to,  $M$  (the number of prototypes) distances need to be computed.
  - For the tree-structured clustering, to decide which cell a query  $x$  goes to, only  $2\log_2(M)$  distances need to be computed.
- Comments on tree-structured clustering:
  - It is structurally more constrained. But on the other hand, it provides more insight into the patterns in the data.
  - It is greedy in the sense of optimizing at each step sequentially. An early bad decision will propagate its effect.
  - It provides more algorithmic flexibility.

---

### Example Distances

- Suppose cluster  $r$  and  $s$  are two clusters merged into a new cluster  $t$ . Let  $k$  be any other cluster.
- Denote between-cluster distance by  $D(\cdot, \cdot)$ .

- How to get  $D(t, k)$  from  $D(r, k)$  and  $D(s, k)$ ?

- *Single-link clustering:*

$$D(t, k) = \min(D(r, k), D(s, k))$$

$D(t, k)$  is the *minimum* distance between two objects in cluster  $t$  and  $k$  respectively.

- *Complete-link clustering:*

$$D(t, k) = \max(D(r, k), D(s, k))$$

$D(t, k)$  is the *maximum* distance between two objects in cluster  $t$  and  $k$  respectively.

- *Average linkage clustering:*

Unweighted case:

$$D(t, k) = \frac{n_r}{n_r + n_s} D(r, k) + \frac{n_s}{n_r + n_s} D(s, k)$$

Weighted case:

$$D(t, k) = \frac{1}{2} D(r, k) + \frac{1}{2} D(s, k)$$

$D(t, k)$  is the average distance between two objects in cluster  $t$  and  $k$  respectively.

For the unweighted case, the number of elements in each cluster is taken into consideration, while in the weighted case each cluster is weighted equally. So objects in smaller cluster are weighted more heavily than those in larger clusters.

- *Centroid clustering:*

Unweighted case:

$$D(t, k) = \frac{n_r}{n_r + n_s} D(r, k) + \frac{n_s}{n_r + n_s} D(s, k) - \frac{n_r n_s}{n_r + n_s} D(r, s)$$

Weighted case:

$$D(t, k) = \frac{1}{2} D(r, k) + \frac{1}{2} D(s, k) - \frac{1}{4} D(r, s)$$

A centroid is computed for each cluster and the distance between clusters is given by the distance between their respective centroids.

- *Ward's clustering:*

$$D(t, k) = \frac{n_r + n_k}{n_r + n_s + n_k} D(r, k) + \frac{n_s + n_k}{n_r + n_s + n_k} D(s, k) - \frac{n_k}{n_r + n_s + n_k} D(r, s)$$

Merge the two clusters for which the change in the variance of the clustering is minimized. The variance of a cluster is defined as the sum of squared-error between each object in the cluster and the centroid of the cluster.

- The dendrogram generated by single-link clustering tends to look like a chain. Clusters generated by complete-link may not be well separated. Other methods are intermediates between the two.

---

### Pseudo Code

1. Begin with  $n$  clusters, each containing one object. Number the clusters 1 through  $n$ .
2. Compute the between-cluster distance  $D(r, s)$  as the between-object distance of the two objects in  $r$  and  $s$  respectively,  $r, s = 1, 2, \dots, n$ . Let square matrix  $D = (D(r, s))$ .
3. Find the most similar pair of clusters  $r, s$ , that is,  $D(r, s)$  is minimum among all the pairwise distances.
4. Merge  $r$  and  $s$  to a new cluster  $t$ . Compute the between-cluster distance  $D(t, k)$  for all  $k \neq r, s$ . Delete the rows and columns corresponding to  $r$  and  $s$  in  $D$ . Add a new row and column in  $D$  corresponding to cluster  $t$ .
5. Repeat Step 3 a total of  $n - 1$  times until there is only one cluster left.

---

### Hipparcos Data

- Clustering based on  $\log L$  and  $BV$ .

---

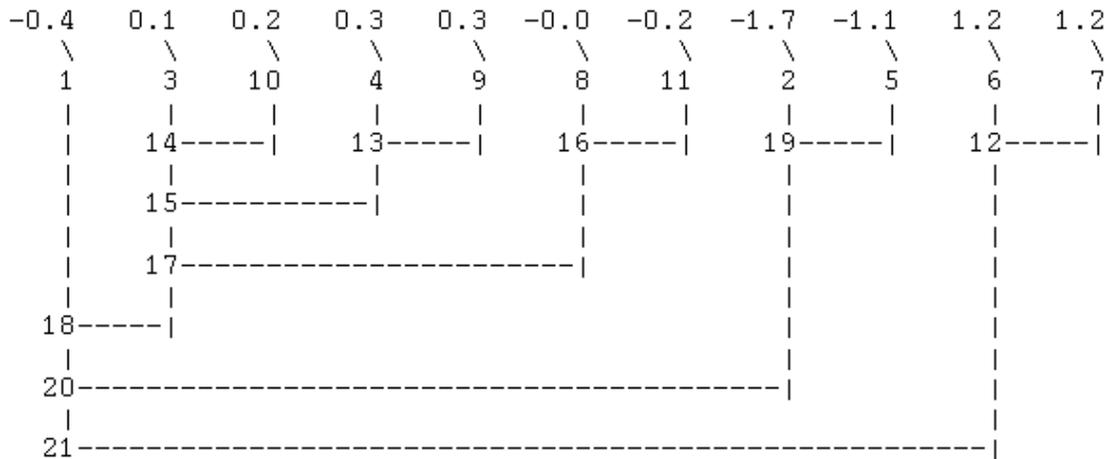
### Mixture Model-based Clustering

- Each cluster is mathematically represented by a parametric distribution. Examples: Gaussian (continuous), Poisson (discrete).
- The entire data set is modeled by a mixture of these distributions.

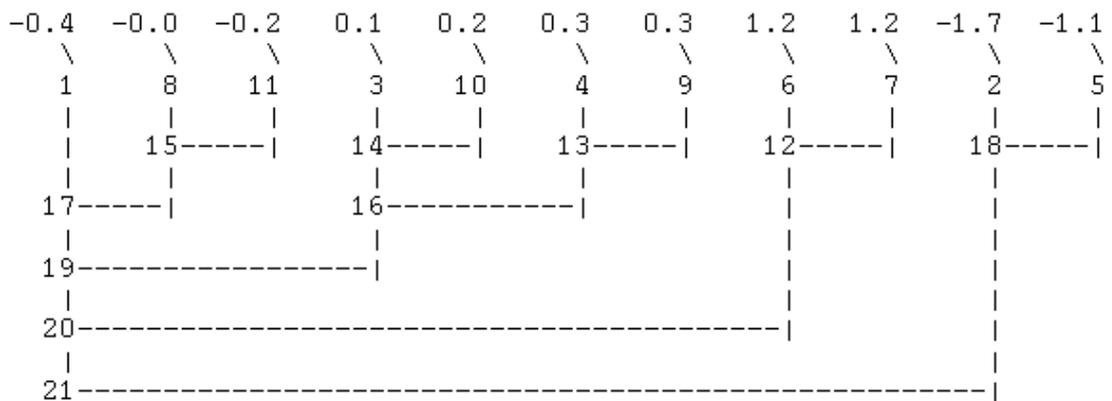
Data set (11 points):

-0.432565, -0.037633, -0.186709, 0.125332, 0.174639  
 0.287676, 0.327292, 1.190915, 1.189164, -1.665584, -1.146471

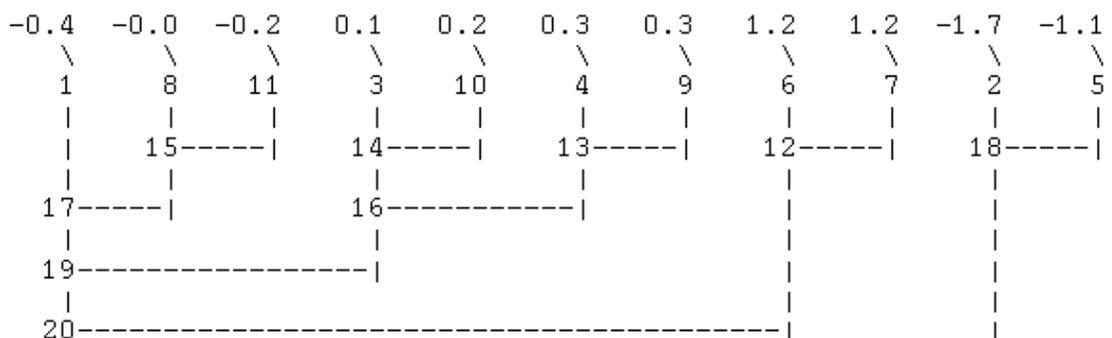
Single-link clustering:



Complete-link clustering:



Ward's clustering:



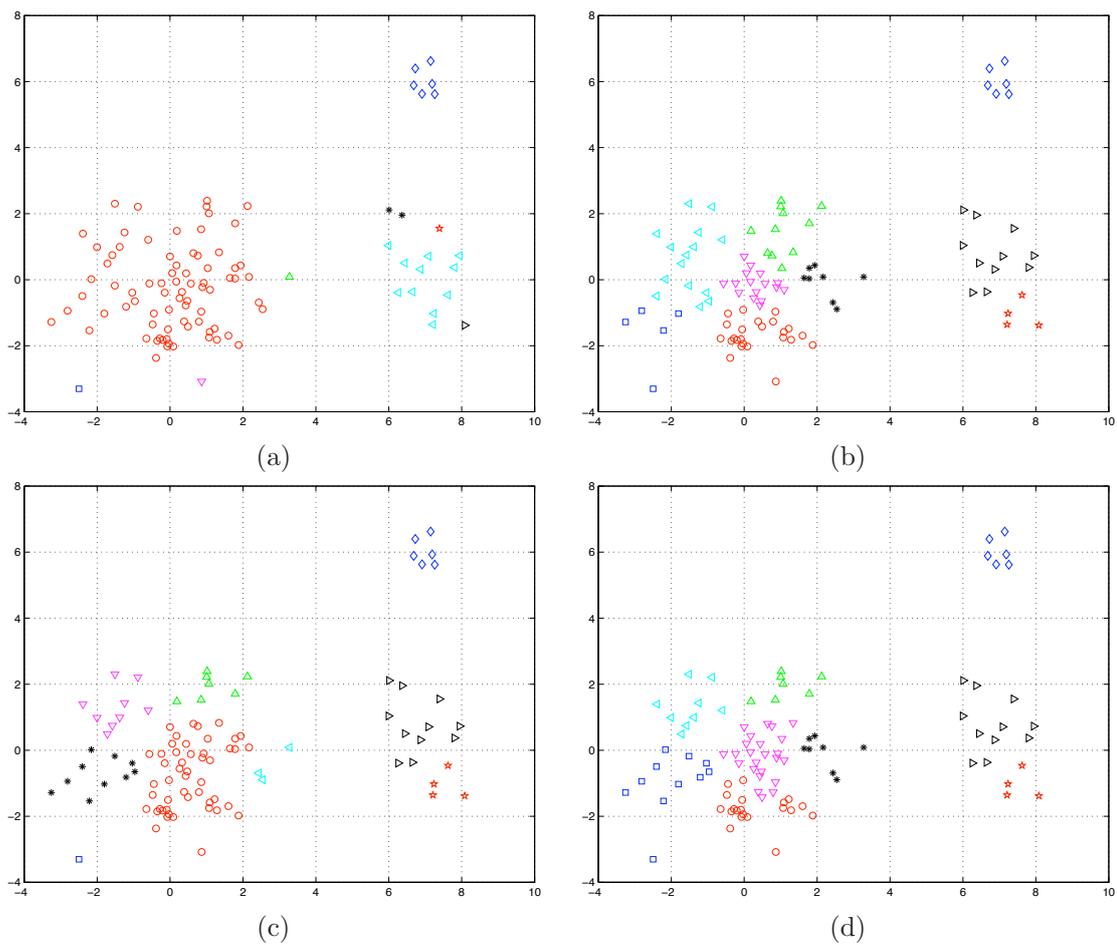


Figure 1: Agglomerate clustering of a data set (100 points) into 9 clusters. (a): Single-link, (b): Complete-link, (c): Average linkage, (d) Wards clustering

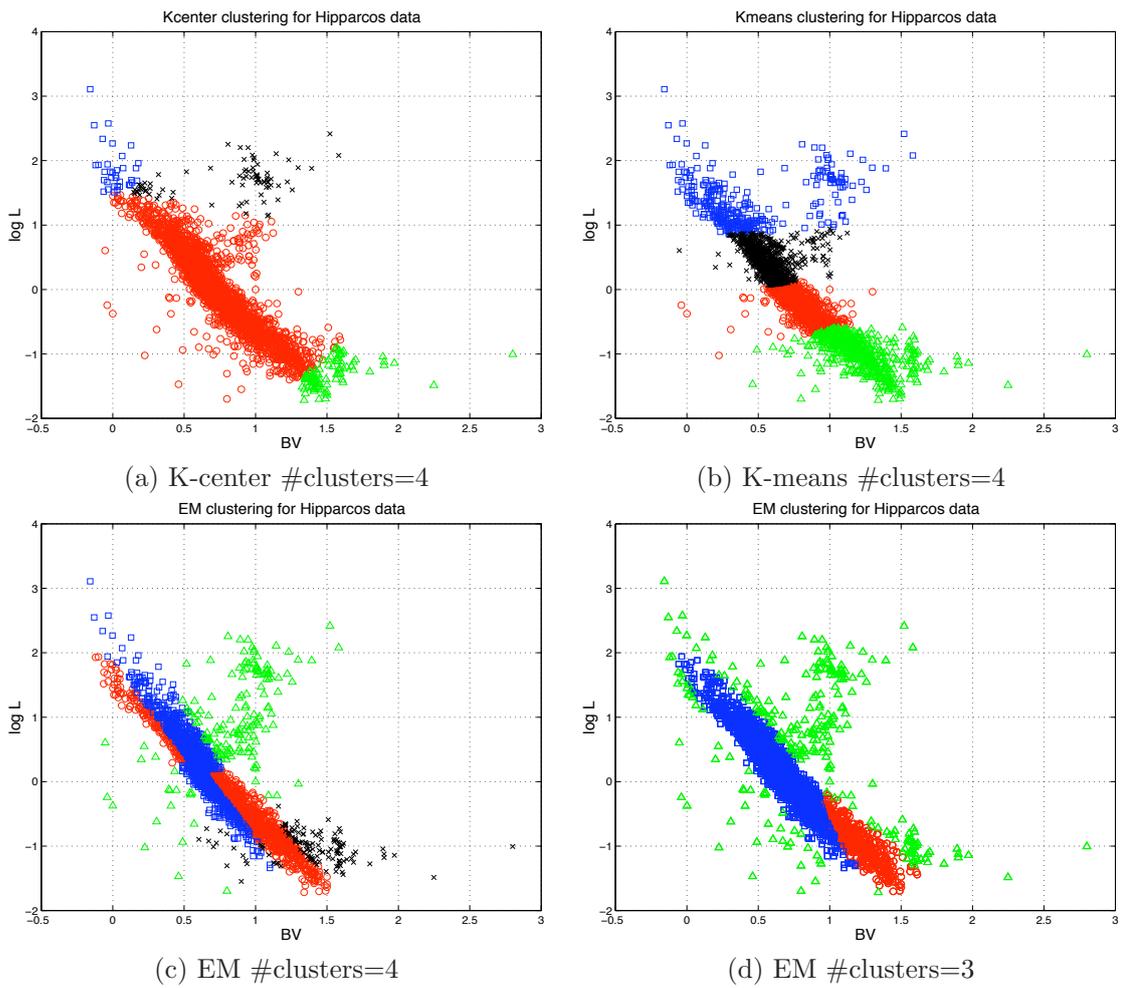


Figure 2: Clustering of the Hipparcos data

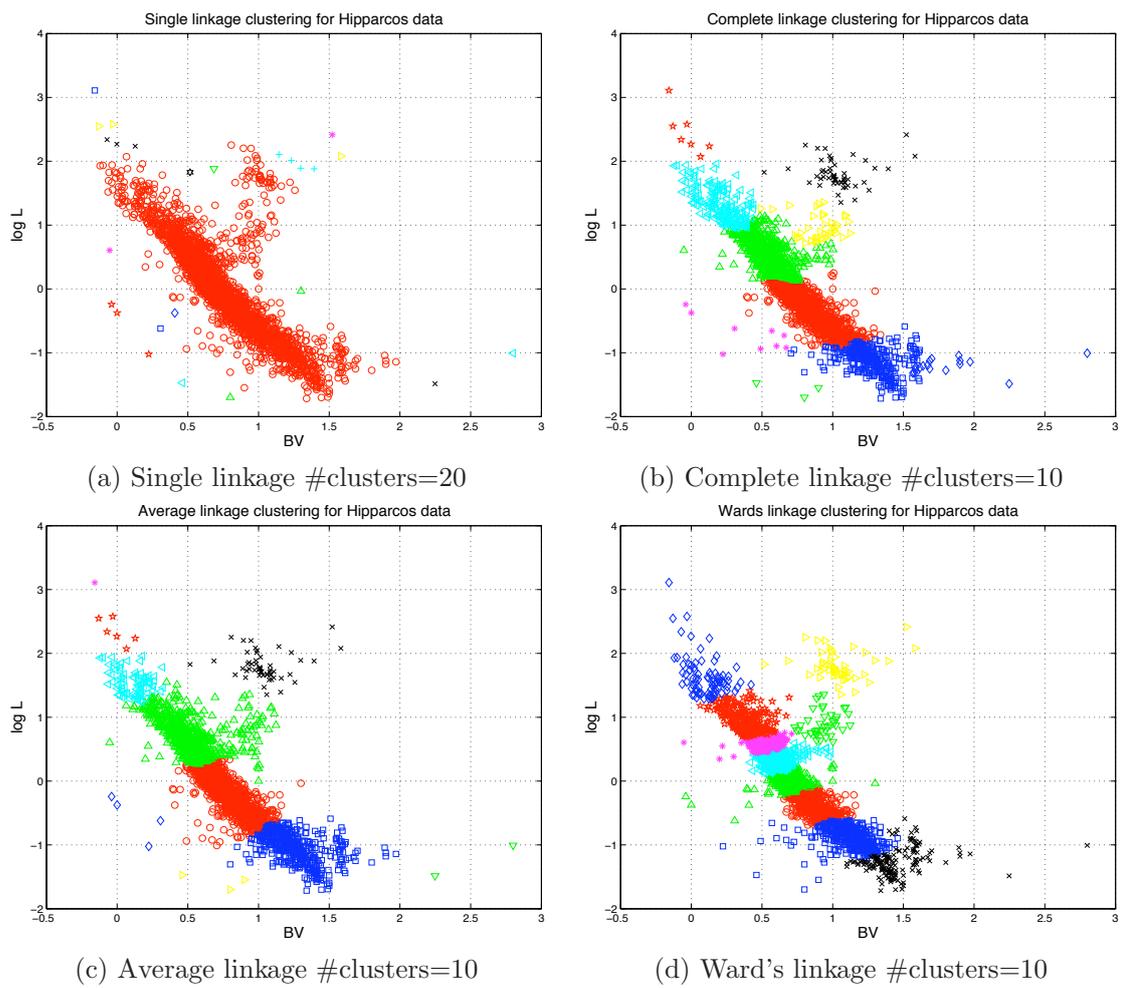


Figure 3: Clustering of the Hipparcos data

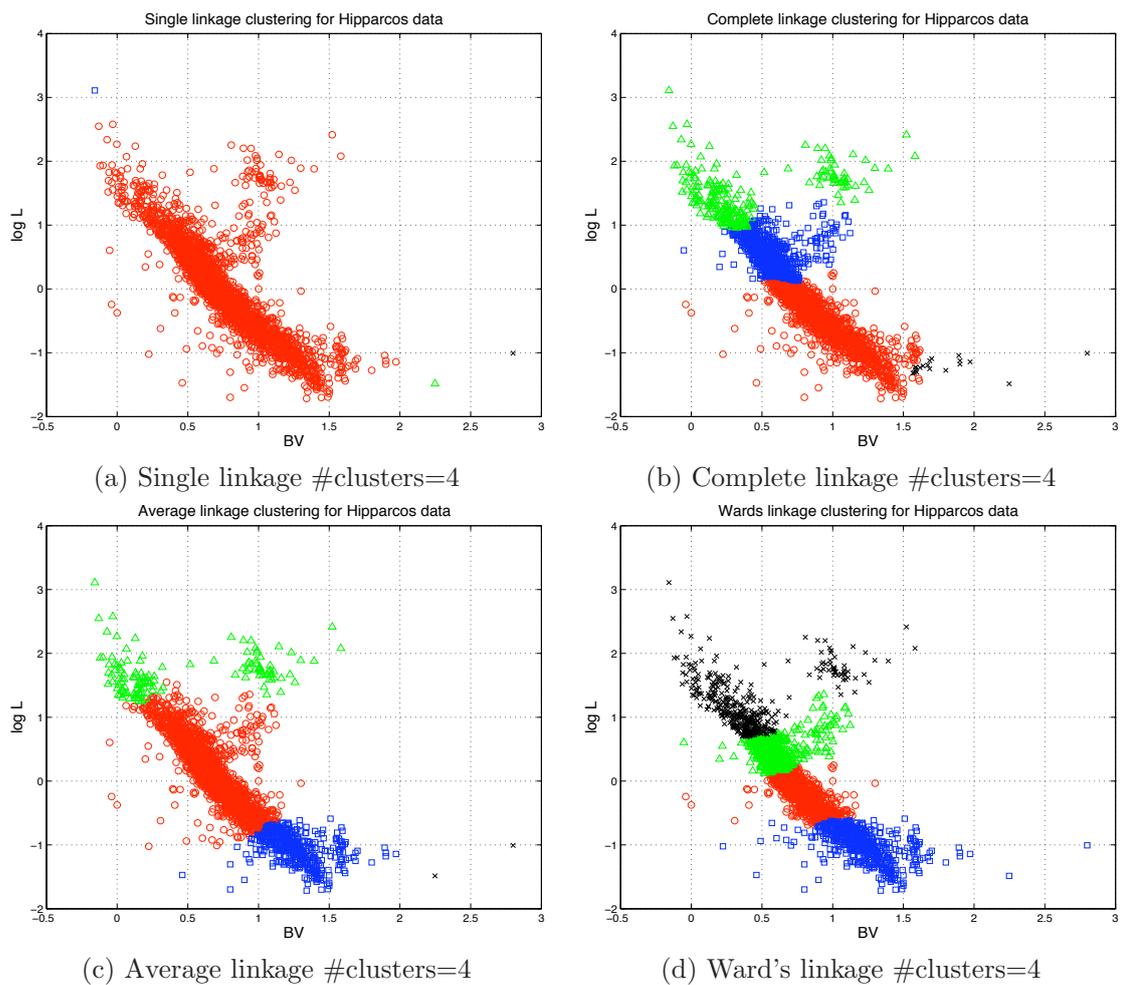


Figure 4: Clustering of the Hipparcos data

- An individual distribution used to model a specific cluster is often referred to as a component distribution.
- Suppose there are  $K$  components (clusters). Each component is a Gaussian distribution parameterized by  $\mu_k, \Sigma_k$ . Denote the data by  $X, X \in \mathcal{R}^d$ . The density of component  $k$  is

$$\begin{aligned} f_k(x) &= \phi(x \mid \mu_k, \Sigma_k) \\ &= \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} \exp\left(-\frac{(x - \mu_k)^t \Sigma_k^{-1} (x - \mu_k)}{2}\right). \end{aligned}$$

- The prior probability (weight) of component  $k$  is  $a_k$ . The mixture density is:

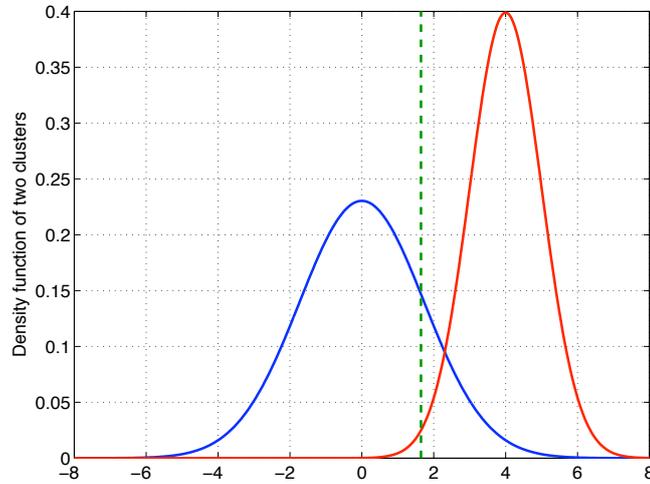
$$f(x) = \sum_{k=1}^K a_k f_k(x) = \sum_{k=1}^K a_k \phi(x \mid \mu_k, \Sigma_k).$$

## Advantages

- A mixture model with high likelihood tends to have the following traits:
  - Component distributions have high “peaks” (data in one cluster are tight)
  - The mixture model “covers” the data well (dominant patterns in the data are captured by component distributions).
- **Advantages**
  - Well-studied statistical inference techniques available.
  - Flexibility in choosing the component distributions.
  - Obtain a density estimation for each cluster.
  - A “soft” classification is available.

## EM Algorithm

- The parameters are estimated by the maximum likelihood (ML) criterion using the EM algorithm.
- The EM algorithm provides an iterative computation of maximum likelihood estimation when the observed data are incomplete.



- Incompleteness can be conceptual.
  - We need to estimate the distribution of  $X$ , in sample space  $\mathcal{X}$ , but we can only observe  $X$  indirectly through  $Y$ , in sample space  $\mathcal{Y}$ .
  - In many cases, there is a mapping  $x \rightarrow y(x)$  from  $\mathcal{X}$  to  $\mathcal{Y}$ , and  $x$  is only known to lie in a subset of  $\mathcal{X}$ , denoted by  $\mathcal{X}(y)$ , which is determined by the equation  $y = y(x)$ .
  - The distribution of  $X$  is parameterized by a family of distributions  $f(x | \theta)$ , with parameters  $\theta \in \Omega$ , on  $x$ . The distribution of  $y$ ,  $g(y | \theta)$  is

$$g(y | \theta) = \int_{\mathcal{X}(y)} f(x | \theta) dx .$$

- The EM algorithm aims at finding a  $\theta$  that maximizes  $g(y | \theta)$  given an observed  $y$ .
- Introduce the function

$$Q(\theta' | \theta) = E(\log f(x | \theta') | y, \theta) ,$$

that is, the expected value of  $\log f(x | \theta')$  according to the conditional distribution of  $x$  given  $y$  and parameter  $\theta$ . The expectation is assumed to exist for all pairs  $(\theta', \theta)$ . In particular, it is assumed that  $f(x | \theta) > 0$  for  $\theta \in \Omega$ .

- **EM Iteration:**
  - E-step: Compute  $Q(\theta | \theta^{(p)})$ .
  - M-step: Choose  $\theta^{(p+1)}$  to be a value of  $\theta \in \Omega$  that maximizes  $Q(\theta | \theta^{(p)})$ .

## EM for the Mixture of Normals

- Observed data (incomplete):  $\{x_1, x_2, \dots, x_n\}$ , where  $n$  is the sample size. Denote all the samples collectively by  $\mathbf{x}$ .
- Complete data:  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ , where  $y_i$  is the cluster (component) identity of sample  $x_i$ .
- The collection of parameters,  $\theta$ , includes:  $a_k, \mu_k, \Sigma_k, k = 1, 2, \dots, K$ .
- The likelihood function is:

$$L(\mathbf{x}|\theta) = \sum_{i=1}^n \log \left( \sum_{k=1}^K a_k \phi(x_i | \mu_k, \Sigma_k) \right) .$$

- $L(\mathbf{x}|\theta)$  is the objective function of the EM algorithm (maximize). Numerical difficulty comes from the sum inside the log.

- The  $Q$  function is:

$$\begin{aligned} Q(\theta'|\theta) &= E \left[ \log \prod_{i=1}^n a'_{y_i} \phi(x_i | \mu'_{y_i}, \Sigma'_{y_i}) | \mathbf{x}, \theta \right] \\ &= E \left[ \sum_{i=1}^n (\log(a'_{y_i}) + \log \phi(x_i | \mu'_{y_i}, \Sigma'_{y_i}) | \mathbf{x}, \theta) \right] \\ &= \sum_{i=1}^n E [\log(a'_{y_i}) + \log \phi(x_i | \mu'_{y_i}, \Sigma'_{y_i}) | x_i, \theta] . \end{aligned}$$

The last equality comes from the fact the samples are independent.

- Note that when  $x_i$  is given, only  $y_i$  is random in the complete data  $(x_i, y_i)$ . Also  $y_i$  only takes a finite number of values, i.e, cluster identities 1 to  $K$ . The distribution of  $Y$  given  $X = x_i$  is the posterior probability of  $Y$  given  $X$ .
- Denote the posterior probabilities of  $Y = k, k = 1, \dots, K$  given  $x_i$  by  $p_{i,k}$ . By the Bayes formula, the posterior probabilities are:

$$p_{i,k} \propto a_k \phi(x_i | \mu_k, \Sigma_k), \quad \sum_{k=1}^K p_{i,k} = 1 .$$

- Then each summand in  $Q(\theta'|\theta)$  is

$$\begin{aligned} &E [\log(a'_{y_i}) + \log \phi(x_i | \mu'_{y_i}, \Sigma'_{y_i}) | x_i, \theta] \\ &= \sum_{k=1}^K p_{i,k} \log a'_k + \sum_{k=1}^K p_{i,k} \log \phi(x_i | \mu'_k, \Sigma'_k) . \end{aligned}$$

- Note that we cannot see the direct effect of  $\theta$  in the above equation, but  $p_{i,k}$  are computed using  $\theta$ , i.e, the current parameters.  $\theta'$  includes the updated parameters.
- We then have:

$$\begin{aligned} Q(\theta'|\theta) &= \sum_{i=1}^n \sum_{k=1}^K p_{i,k} \log a'_k + \\ &\quad \sum_{i=1}^n \sum_{k=1}^K p_{i,k} \log \phi(x_i | \mu'_k, \Sigma'_k) \end{aligned}$$

- Note that the prior probabilities  $a'_k$  and the parameters of the Gaussian components  $\mu'_k, \Sigma'_k$  can be optimized separately.

- The  $a'_k$ 's subject to  $\sum_{k=1}^K a'_k = 1$ . Basic optimization theories show that  $a'_k$  are optimized by

$$a'_k = \frac{\sum_{i=1}^n p_{i,k}}{n}.$$

- The optimization of  $\mu_k$  and  $\Sigma_k$  is simply a maximum likelihood estimation of the parameters using samples  $x_i$  with weights  $p_{i,k}$ . Basic optimization techniques also lead to

$$\mu'_k = \frac{\sum_{i=1}^n p_{i,k} x_i}{\sum_{i=1}^n p_{i,k}}$$

$$\Sigma'_k = \frac{\sum_{i=1}^n p_{i,k} (x_i - \mu'_k)(x_i - \mu'_k)^t}{\sum_{i=1}^n p_{i,k}}$$

- After every iteration, the likelihood function  $L$  is guaranteed to increase (may not strictly).
- We have derived the EM algorithm for a mixture of Gaussians.

## EM Algorithm for the Mixture of Gaussians

Parameters estimated at the  $p$ th iteration are marked by a superscript  $(p)$ .

1. Initialize parameters
2. E-step: Compute the posterior probabilities for all  $i = 1, \dots, n$ ,  $k = 1, \dots, K$ .

$$p_{i,k} = \frac{a_k^{(p)} \phi(x_i | \mu_k^{(p)}, \Sigma_k^{(p)})}{\sum_{k=1}^K a_k^{(p)} \phi(x_i | \mu_k^{(p)}, \Sigma_k^{(p)})}.$$

3. M-step:

$$a_k^{(p+1)} = \frac{\sum_{i=1}^n p_{i,k}}{n}$$

$$\mu_k^{(p+1)} = \frac{\sum_{i=1}^n p_{i,k} x_i}{\sum_{i=1}^n p_{i,k}}$$

$$\Sigma_k^{(p+1)} = \frac{\sum_{i=1}^n p_{i,k} (x_i - \mu_k^{(p+1)})(x_i - \mu_k^{(p+1)})^t}{\sum_{i=1}^n p_{i,k}}$$

4. Repeat step 2 and 3 until converge.



## Chapter 22

# CLUSTER ANALYSIS IN R

*Notes by Arnab Chakraborty*

# Cluster Analysis

## Up above us all so high...

In the first part of this tutorial we shall imagine ourselves in a satellite taking photographs of the earth. In the process we shall learn some image processing as well as some clustering techniques.



A satellite image

This shows part of a blue ocean and two land masses with some green vegetation. We can recognize these by the color. A human eye is extremely adept at detecting large regions of similar colors in an image. A camera, however, has no such ability. It merely scans a scene in a mechanical way, faithfully reporting every dot in it, but has no idea if the dots constitute any pattern or not. The above image, for example, is made of 3000 dots, arranged in a rectangle that is 75 dots wide and 40 dots high. Each dot, by the way, is called a **pixel** (a short form for "picture element"). The camera uses three filters: red, green and blue. The camera reports the red-ness, green-ness and blue-ness of each pixel. The numbers are from 0 to 255. Thus, a perfectly red dot will have red value 255, green value 0 and blue value 0. If you increase the green component the dot will appear yellowish (since red plus green light make yellow light).

The 3000 dots in our pictures, therefore, are just 9000 numbers to the satellite. It is better to think of them as 3000 points in 3D. Our eye could quickly group (or cluster) these 3000 points into two clusters: ocean and land. The ability to group a large number of points in **d**-dimensions into a relatively smaller number of classes is the aim of **cluster analysis**. The file [sat.dat](#) stores the data set for this image. Before going into the statistics let us learn how to turn these 3000 points into an image using R. First read the data.

```
dat = read.table("sat.dat")
```

Notice that this file has no header line, so we have omitted the usual `head=T` option. We are told that the data file has 3 columns and 3000 rows of numbers. Each row is for a pixel, and the 3 columns are for red, green and blue, respectively. Let us assign these names to the columns.

```
names(dat) = c("red", "green", "blue")
attach(dat)
```

Next make a vector of colors, one color for each point.

```
mycol = rgb(red, green, blue, max=255)
```

The function **rgb** makes colors by combining **red**, **green** and **blue** values. The `max` specification means 255 is to be considered the highest value for each color.

Next we need to specify the size of the image. It is 75 pixels in width and 40 pixels in height.

```
rows = 1:75
columns = 1:40
```

So the 3000 points are actually arranged as a 40 by 75 matrix. (The height, 40, is the number of rows.)

```
z = matrix(1:3000, nrow=75)
```

Now we are ready to make the image.

```
image(rows, columns, z, col=mycol)
```

If you do not like to see the axes and labels, then you may remove them using

```
image(rows, columns, z, col=mycol, bty="n", yaxt="n",
      xaxt="n", xlab="", ylab="")
```

Based on the 3000 points the statistical brain of the satellite has to decide that is seeing two land masses peeping out from a vast blue ocean. And one important tool to achieve this is called **clustering**.

But before we go into it let us make a 3D scatterplot out of the 3 variables.

```
library(scatterplot3d) #Warning! It won't work unless
                      #scatterplot3d is installed.
                      #I mention this to explain how
                      #you may do it if you can download
                      #this (pretty large) package.
                      #Also, this package is readily available for
                      #only Windows. To use it in Linux, you have
                      #to download the source code and compile!
                      #But we shall see a work around for this soon.
```



`scatterplot3d` is a R package. But it may not be already available on your computer. Then you can download it from the internet using

```
install.packages("scatterplot3d")
```

This will first let you choose from a list of web repositories to download from. (Preferably choose a US-based repository as these contain more packages). The download and installation starts automatically (may be slow depending on your internet connection).



Alternatively, you may download the zip file for your package (after finding it out using Google, say) from the internet directly (using your favorite downloader) and then install it locally by the command

```
install.packages(choose.file()) #for Windows
install.packages(file.choose()) #for Linux
```



Then you can use:

```
scatterplot3d(red, green, blue, color=mycol)
```

One problem with `scatterplot3d` is that there is no simple way to specify the viewing angle. This is rectified in the small script [scat3d.r](#), which you can download. Now use

```
source("scat3d.r")
scat3d(red, green, blue, col=mycol)
```

You can also turn the plot around specifying the angles `theta` and `phi` as follows.

```
scat3d(red, green, blue, col=mycol, theta = 20, phi=0)
```

Just a jumble of points, right? We shall apply a technique called **k-means** clustering to group the 3000 points into 2 clusters. Then later we shall learn how **k-means** clustering

works.

## k-means clustering

Apply the command

```
cl = kmeans(dat,2)
```

and see what **kmeans** has returned.

```
names(cl)
cl$clus
```

Each pixel is put into one of two clusters (called 1 and 2), and `cl$clus` tells us the cluster of each pixel. The centers of the two clusters are given by

```
cl$center
```

Now we shall make an image of the clustering result. We are going to have an image consisting of only two colors, one for each cluster. It is natural to choose the center as the representative color for a cluster.

```
c2 = rgb(cl$cen[,1],cl$cen[,2],cl$cen[,3],max=255)
c2
```

Next we have to wrap the long vector `cl$clus` in to a 75 by 40 matrix in order to make the image.

```
image(rows,columns, matrix(cl$clus,75,40),col=c2)
```

Do you think that the ocean is well-separated from the land?

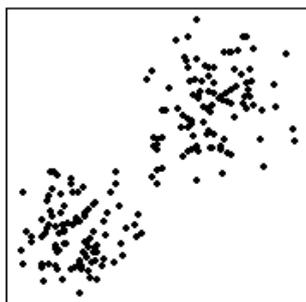
What about adding some more details? For this we shall increase **k** to 3, say.

```
cl = kmeans(dat,3)
c3 = rgb(cl$cen[,1],cl$cen[,2],cl$cen[,3],max=255)
image(rows,columns, matrix(cl$clus,75,40),col=c3)
```

As you increase the number of clusters more and more details are added to the image. Notice that in our example the details get added to the land masses instead of the ocean which is just a flat uninteresting sheet of water.

### How the k-means algorithm works

Consider the data set shown below. We can see that it has two clusters of points.



### Want to find the two clusters

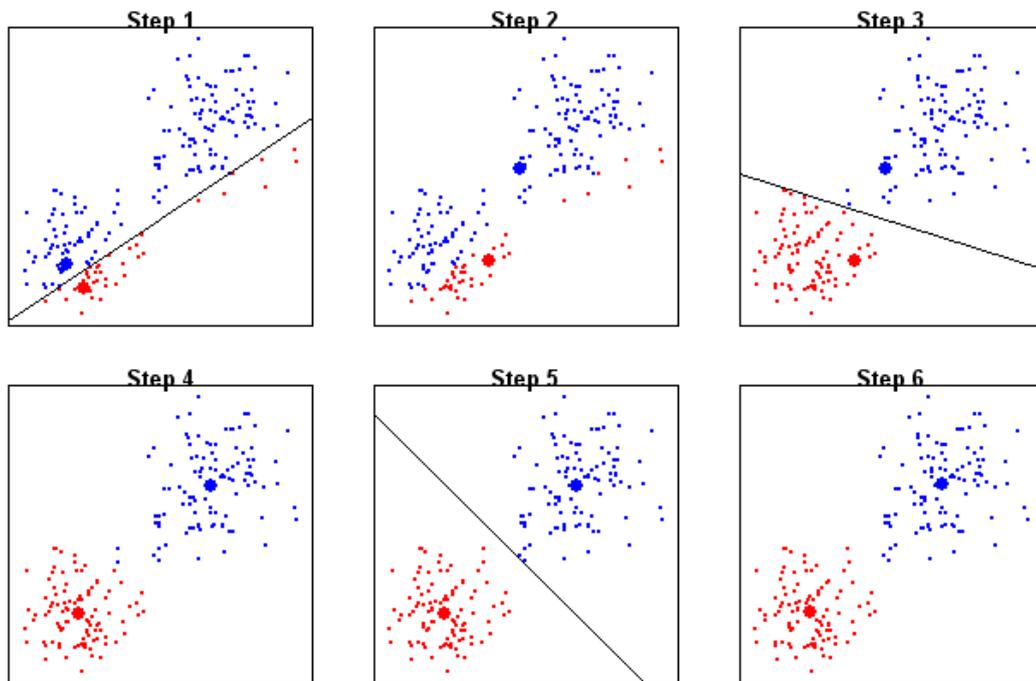
In order to find these two clusters we shall employ the k-means algorithm with **k=2**.

The algorithm proceeds by alternating two steps that we shall call Monarchy and Democracy (see the figure below). In step 1, we apply Monarchy by choosing two random points as the kings of the two clusters. These two kings create their empires by enlisting the data points nearest to them as their subjects. So we get two kingdoms separated by the perpendicular bisector between the kings.

In step 2 we have Democracy, where the king is abolished and the data points in each kingdom choose their respective leaders as the average of themselves.

This election over, Monarchy kicks in once again as the elected leaders behave as kings, enlisting the nearest data points as subjects. This redefines the boundary of the kingdoms.

Then Democracy starts again, after which comes Monarchy. This process continues until the kingdoms do not change any more. These two kingdoms finally give the two clusters. The kings (or elected leaders) are the centers of the clusters.

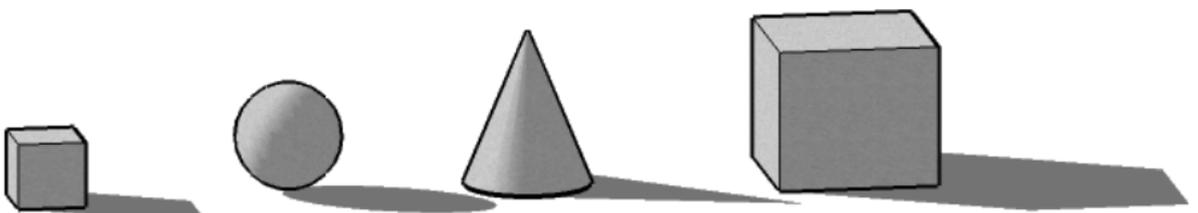


**Steps in the 2-means algorithm**

## Hierarchical clustering

Since the amount of details increases with a larger number of clusters, we come to the question "How to choose the best number of clusters?" One way to resolve the issue is to first take a look at the results for *all* possible numbers of clusters. Clearly, if there are  $n$  cases in the data, then the number of clusters can go from 1 to  $n$ . This is the idea behind hierarchical clustering. It is like zooming down gradually on the details.

Consider the four shapes below.



**Group these**

Suppose that you are to group the similar shapes together making just *two* clusters. You'd most possibly put the two cubes in one cluster, while the sphere and the cone (both having round surfaces) will make the second cluster. So the two clusters are

(small cube, large cube) , (sphere, cone).

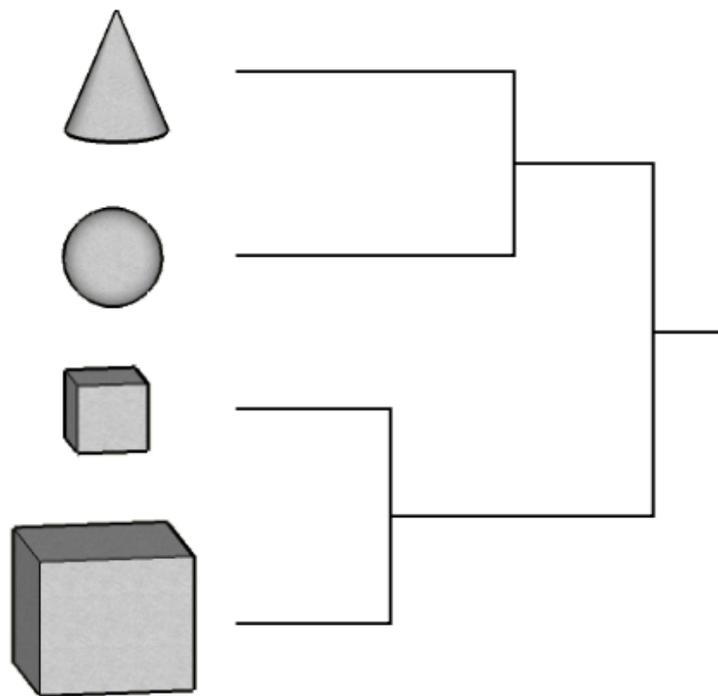
Next, suppose that you are to further split any one of the clusters (making *three* clusters in all). One natural way is to split the (sphere, cone) cluster into two separate clusters (sphere) and (cone) resulting in three clusters:

(small cube, large cube) , (sphere), (cone).

If we are to further increase the number of clusters, we have to split the first cluster:

(small cube), (large cube) , (sphere), (cone).

And we have reached the maximum number of clusters. This step-by-step clustering process may be expressed using the following diagram which is sometimes called a **clustering tree** and sometimes called a **dendrogram**.



**Cluster tree**

Note that this *single* tree contains information about *all* possible numbers of clusters.

**An application**

Here we shall illustrate an application of hierarchical clustering using the data set [shapley.dat](#).

```
shap = read.table("shapley.dat", head=T)
dim(shap)
names(shap)
```

In this file we have some missing values for the variable `Mag` in column 3 that are coded as 0 (instead of **NA**). Let us convert these to **NA**

```
shap[shap[,3]==0,3] = NA
plot(shap,pch=".")
```

Now we shall perform hierarchical clustering, but we shall use only a subset of the data. We shall take only those cases where `v` is between 12000 and 16000. Also we shall consider only the three variables `RA`, `DE` and `v`.

```
attach(shap)
shap = shap[V>12000 & V<16000,c("RA","DE","V")]
```

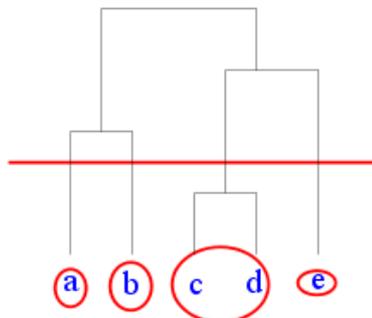
Next we shall centre and scale each variable (*i.e.*, we shall subtract the mean and divide by the standard deviation). This is conveniently done by the `scale` function.

```
shap = scale(shap)
```

In order to perform hierarchical clustering using the `hclust` function we have to find the distance matrix of the data. The  $(i,j)$ -th entry of this matrix gives the Euclidean distance between the  $i$ -th and the  $j$ -th point in the data set.

```
d = dist(shap)
mytree = hclust(d) #this may take some time
plot(mytree)
```

As we have mentioned earlier this hairy tree-like object (called a **dendrogram**) represents clustering allowing *all possible* numbers of clusters. In order to use it we have to "cut" it at a particular height as shown in the diagram below.



**Cutting a dendrogram**

Suppose that we want 3 clusters for the Shapley data set. The dendrogram shows that we need to cut near height 6.

```
classes = cutree(mytree,h=6)
table(classes) #How many points in each class
```

Since we are working with 3D data (we have only 3 columns) so we can make a 3D scatterplot.

```
scatterplot3d(shap,color=classes)
```

There are many clustering algorithms. In fact there is an entire R package called `cluster` devoted to clustering methods.



Different clustering methods sometimes produce widely differing clusterings! One needs discretion and domain knowledge to choose one over the other.

Let us take a look at a method called AGglomerative NESTing (AGNES).

```
library('cluster')
```

The R function that we shall employ is called **agnes**. We shall apply it on the Shapley data.

```
agn = agnes(shap) #this will take some time
plot(agn, which=2)
```

R can make two possible plots of the output from **agnes**. Here the argument `which=2` requests the second of these. The returned object `agn` is a clustering tree somewhat like that returned by `hclust`. But the internal structures differ. So you cannot directly apply the **cutree** function to it. We first need to convert it to the structure compatible with **hclust**.

```
agnh = as.hclust(agn)
tmp = cutree(agnh, h=2.2)
table(tmp)
```

Here we have performed the cut at a specified height. We could also have specified the number of required clusters.

```
tmp2 = cutree(agnh, k=10)
table(tmp2)
```

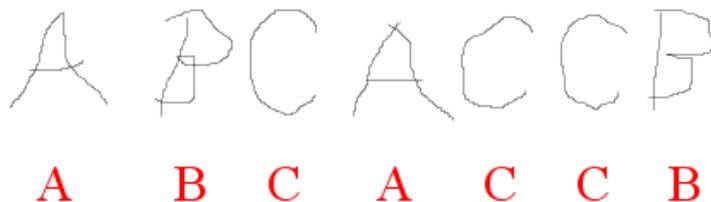
You may recall the terms "single linkage", "complete linkage" and "average linkage" from the theory class. What **agnes** uses by default is "average linkage". Let us see how "single linkage" performs for our data.

```
agn.sing = agnes(shap, method="single")
plot(agn.sing, which=2)
```

**Exercise:** Write an R script to split the Shapley data into 5 clusters using the "complete linkage" method of AGNES, and then make a 3D scatterplot of the data set using the colors red, green, blue, black and pink for the 5 clusters. [\[Solution\]](#)

### **k-nearest neighbors classification**

This method is motivated by how a child is taught to read. The teacher first shows her some letters and tells her the names of the letters, so that she can associate the names with the shapes.



**The teacher's voice is shown in red**

Then she is presented with some letters to see if she can identify them.



**The child is to identify these**

When she sees a new letter the child quickly matches the new shape with the ones she has learned, and says the name of the letter that comes closest to the new letter.

Let us take a note of the different parts of this process. First, we have a **training data set**: the letter shapes and their names (shown in the first picture). Then we have a **test data set** which is very much like the training set, except that the names are not given.

We shall employ the same method to recognize the Hyades stars! We shall start with a training data set consisting of 120 stars of which the first 60 will be Hyades and the others not Hyades. We are supposed to learn to recognize a Hyades star based on these. Then we shall be given a test data set consisting of 56 new stars, some of which are Hyades (we are not told which). Our job will be to identify the Hyades stars in the test data sets. First download (right-click and save to your computer) the two data sets: [train.dat](#) and [tst.dat](#). Next, load them into R.

```
trn = read.table("train.dat",head=T)
tst = read.table("tst.dat",head=T)
dim(trn)
dim(tst)
```

We are told that the first 60 stars in the training data set are Hyades, while the remaining 60 are not. So accordingly we shall make a vector of size 120:

```
trueClass = c(rep("Hyad", 60), rep("NonHyad", 60))
trueClass
```

Now it is time to apply the **k**-nearest neighbors algorithm.

```
library("class")
foundClass = knn(trn,tst,trueClass,k=3)
```

Remember that `tst` consists of 56 stars, so `foundClass` is a vector of that length.

```
foundClass
```

Here it so happens that there is not a single mistake!



# Chapter 23

## MCMC

*Notes by Arnab Chakraborty*

# A gentle introduction to MCMC

*Arnab Chakraborty*

## 1 Bit of a history

We see the effects of chance everywhere around us—from the roll of a die to the inevitable measurement errors. Many chance phenomena are repetitive in nature. And this provides a way to deal with them. The earliest technique used for this purpose is to wait and see. For example, by repeating a measurement many times we can get an idea about the measurement error. But this may prove too expensive. So the next technique was developed—probability theory, or the mathematics of chance. Long term behaviour could be approximated mathematically using this technique. But unfortunately the math is often too difficult to compute. The advent of modern computers able to generate random numbers revived interest in the old “wait and see” approach. This new version is called the Monte Carlo approach. It is often used even for problems that have no randomness in it.

```
n= 10000
x = runif(n, min=-1, max=1)
y = runif(n, min=-1, max=1)
inside= x*x+y*y<1
4*mean(inside)
plot(x,y,col=ifelse(inside,'red','black'),pch=20)
```

This achievement, however, was soon overtaken by ambition. Probability distributions were encountered that were not easy even to simulate from. These distributions came most often from Bayesian computations.

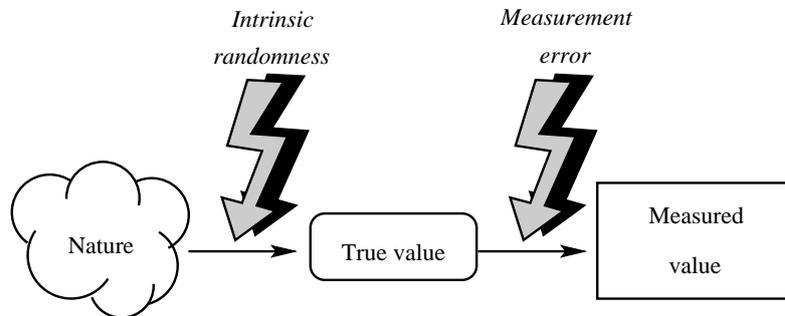
## 2 A Bayesian excursion

**Example:** This example is taken from Brandon Kelly’s paper [The Astrophysical Journal, 665:1489-1506, 2007] on linear regression with measurement error. Kelly’s solution to the problem (which we shall discuss later) is part of the Astrolib library maintained by NASA.

Here we start with the usual linear regression equation

$$Y = a + bX.$$

In a typical astronomical set up, both  $X$  and  $Y$  are measured quantities. Usually a measurement is subject to two types of randomness—intrinsic randomness affecting the quantity to be measured plus the measurement error. Accordingly, it helps to visualise the entire process as a two-step experiment as shown below.



So behind every measured quantity there is an unobserved true value, which is again random. Let the true values underlying  $X$  and  $Y$  be  $\xi$  and  $\eta$ . Kelly assumes that the intrinsic randomness in  $\xi$  can be captured via a Gaussian mixture.

[Eqn 15]

Then  $\eta$  is generated from  $\xi$  plus more intrinsic randomness:

[Eqn 13]

Finally he models the measurement errors as

[Eqn 14]

Next we shall specify the priors.

We assume uniform prior for  $\pi$ 's.

We assume improper, uniform prior for  $a, b$  and  $\sigma^2$ .

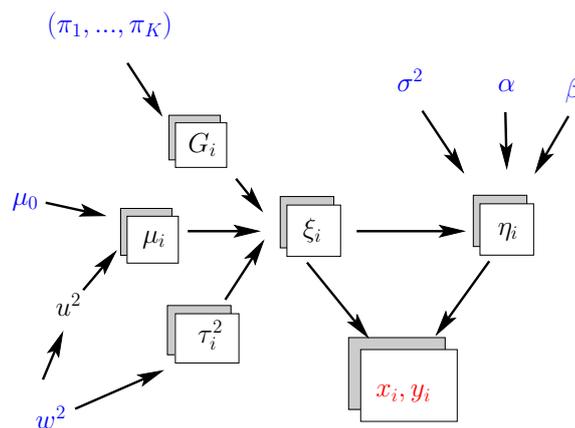
The  $\mu$ 's are given  $N(\mu_0, u^2)$  prior and  $u^2, \tau^2$ 's get inverse  $w^2\chi^2_{(1)}$ -priors.

Finally  $\mu_0$  and  $w^2$  are given improper, uniform prior.

[Eqns 43-49]

So the posterior is quite complicated.

The entire set up can be depicted as a *graphical model*:



### 3 MCMC

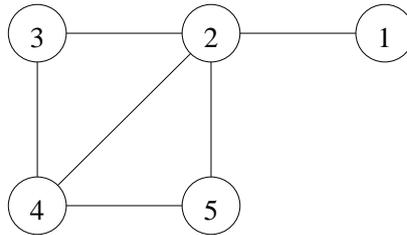
#### 3.1 Basic idea

A novel approach was suggested: MCMC. Consider shuffling a deck of 52 cards. The aim is to select a random permutation out of the  $52!$  permutations. In principle we could do this by–

- drawing one chit at random from  $52!$  chits mixed up in an urn. [Practically impossible]
- picking one cards successively at random, and put them down on the table. [Simpler, but still too cumbersome]

Instead, we simply shuffle the deck –stepwise randomisation. Each shuffle mixes the cards to some extent. Repeated shuffles eventually achieve sufficient randomness. A natural question is: when can a randomisation be split up into steps like this? You can think of this as iterative randomiser, just as the Newton-Raphson method is an iterative equation solver.

Here is another example.



A man starts from vertex 1. At each step he chooses any one of the adjacent vertices randomly, and moves there. Thus his position becomes more and more random. What is the limiting distribution? The answer may be guessed: he would spend more time in vertices with higher number of neighbours. Thus, if  $N_i$  denotes the the number of neighbours of vertex  $i$ , then the limiting distribution is proportional to

$$(N_1, \dots, N_5) = (1, 4, 2, 3, 2).$$

The following simulation confirms this.

```
dest = list(2,c(1,3,4,5),c(2,4),c(2,3,5),c(2,4))

doJump = function(wherNow) {
  if(length(dest[[wherNow]]) == 1)
    dest[[wherNow]]
  else
    sample(dest[[wherNow]],1)
}
```

```

propChain = function(n) {
  x = numeric(n)
  x[1] = 1
  for(i in 2:n)
    x[i] = doJump(x[i-1])

  return(x)
}

```

### 3.2 Gibbs' sampler

Given a distribution how to come up with a mixing scheme? How to determine when sufficient mixing has been achieved?

A popular method is to use Gibbs' sampler. Consider a man trying to move through a distance that is too long to manage in a single jump. Then he starts walking, which means moving one leg at a time keeping the other foot firmly fixed on ground. Well, that's what a Gibbs' sampler does. To simulate from the joint distribution of  $(X_1, \dots, X_n)$  it proceeds as follows.

1. First it keeps  $X_2, \dots, X_n$  fixed, and generates  $X_1$  from the conditional distribution of  $X_1$  given  $(X_2, \dots, X_n)$ .
2. Then it keeps  $X_1$  fixed at this new value, and  $X_3, \dots, X_n$  at the old values, and generates  $X_2$  from the conditional distribution of  $X_2$  given  $(X_1, X_3, \dots, X_n)$ .
3. The process is continued until we have new values for all the  $X_i$ 's.
4. The entire process is repeated until convergence.

Why should the full conditionals be easier to simulate from than the joint distribution? Two reasons:

1. One dimensional distributions are easier to simulate from.
2. In a Bayesian set up we can take advantage of conjugate priors.

[Gibb's sampler in Kelly's model: eqns 53-58, 66-68, 73-89, 93-97, 101-103]

What if all the full conditionals are not simple? Use Metropolis-Hastings. We shall discuss it later.

## 4 Convergence issues

How to determine convergence? No satisfactory answer known. Remember that here we are talking about convergence of distributions, and not the generated numbers. So ideally we should estimate the distribution at different points in the chain, and compare. There are a number of approaches possible:

- Run parallel chains, draw histograms, compare.
- Run one chain, consider well-separated batches, compare using moments like mean or variance.
- Wait for autocorrelation to die down.

#### 4.1 Bottleneck problem

```

condSim = function(given) {
  if(given < 0.9) {
    runif(1,min=0,max=1)
  }
  else if( given < 1) {
    runif(1,min=0,max=given+0.1)
  }
  else if( given < 3) {
    runif(1,min=given-0.1,max=given+0.1)
  }
  else if( given < 3.1) {
    runif(1,min=given-0.1,max=4)
  }
  else {
    runif(1,min=3,max=4)
  }
}

rawSim = function(n = 5000) {
  x = numeric(n)
  y = numeric(n)
  x[1] = 0.5
  y[1] = 0.5
  for(i in 2:n) {
    x[i] = condSim(y[i-1])
    y[i] = condSim(x[i])
  }

  par(mfrow=c(2,2))
  plot(x,xlim=c(0,4),ylim=c(0,4),ty='n',ylab='y',xlab='x')
  polygon(c(0,1,1,3.1,4,4,3,3,0.9,0,0),
          c(0,0,0.9,3,3,4,4,3.1,1,1,0),
          col="pink")
  points(x,y,pch='.')
  plot(y,ty='l',ylim=c(0,4),xlab='step')
}

```

```
plot(x,ty='l',ylim=c(0,4),xlab='step')
par(mfrow=c(1,1))
}
```

```
s2 = sqrt(2)
x1 = 1/s2
delX = 0.1/s2
x2 = s2 - delX
ts2 = 3*s2
x3 = ts2 + delX
x4 = ts2 + x1
```

```
fs2 = 4*s2
```

```
yFromX = function(x) {
  if(x<x1) {
    runif(1,min=-x,max=x)
  }
  else if(x<x2) {
    runif(1,min=x-s2,max=s2-x)
  }
  else if(x<x3) {
    runif(1,min=-delX,max=delX)
  }
  else if(x<x4) {
    runif(1,min=ts2-x,max=x-ts2)
  }
  else {
    runif(1,min=x-fs2,max=fs2-x)
  }
}
```

```
y1 = delX
y2 = 1/s2
```

```
xFromY = function(y) {
  y= abs(y)

  if(y < y1) {
    runif(1,min=y,max=fs2-y)
  }
  else {
    ifelse(runif(1) < 0.5,
           runif(1,min=y,max=s2-y),
           runif(1,min=ts2+y,max=fs2-y))
  }
}
```

```

    }
  }

rotSim = function(n = 5000) {
  x = numeric(n)
  y = numeric(n)
  x[1] = 0.5
  y[1] = 0.5
  for(i in 2:n) {
    x[i] = xFromY(y[i-1])
    if(x[i] >= fs2) stop(paste("Impossible x from y = ",y[i-1]))
    y[i] = yFromX(x[i])
    if(abs(y[i]) >= x1) stop(paste("Impossible y from x = ",x[i]))
  }

  par(mfrow=c(2,2))
  plot(x,xlim=c(0,fs2),ylim=c(-x1,x1),ty='n',xlab='x',ylab='y')
  polygon(c(0,x1,x2,x3,x4,fs2,x4,x3,x2,x1,0),
          c(0,-x1,-y1,-y1,-x1,0,x1,y1,y1,x1,0),
          col="pink")
  points(x,y,pch='.')
  plot(y,ty='l',ylim=c(-x1,x1),xlab='step')
  plot(x,ty='l',ylim=c(0,fs2),xlab='step')
  par(mfrow=c(1,1))
}

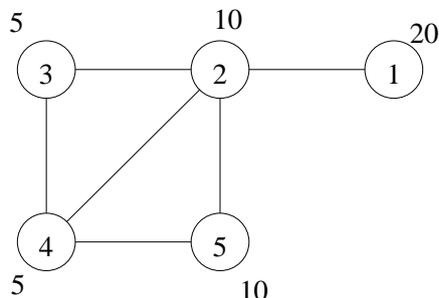
```

## 5 Metropolis-Hastings' method

If Gibbs' sampler is compared to walking, then Metropolis-Hastings' method is a generalisation that also allows jumps with both feet off the ground simultaneously. You do not *have* to jump, walking is still allowed. So Gibbs' sampler is a special case of Metropolis-Hastings' method.

Metropolis-Hastings' method has a surprising amount of generality. It starts with any Markov Chain (ie, any set of jump rules), and tweaks it so that it converges to any target distribution of our choice! Well, that's bit of an overstatement. We do need the initial chain to satisfy *some* conditions, but they are really weak conditions.

Consider the graph walking example again. We started with the Markov Chain (jump rule): jump to any adjacent vertex with equal probability. This causes us to visit vertices with more neighbours to be visited more often. But suppose that we want to visit them randomly according to the probabilities proportional to the following labels:



Then Metropolis-Hastings' method modifies the original Markov Chain as follows. Suppose we are at 2, and we plan to jump to 3. But we do not immediately carry out our plan. Instead, we check the two labels,  $\pi_2$  and  $\pi_3$ . We jump with probability

$$\min \left\{ 1, \frac{\pi(3)p(2|3)}{\pi(2)p(3|2)} \right\}.$$

Considering the case  $p(2|3) = p(3|2)$ , we can say that we are more willing to jump to higher labels than to lower ones.

```
MHchain = function(prob=rep(1,5),n=5000) {
  x = numeric(n)
  x[1] = 1
  for(i in 2:n) {
    proposal = doJump(x[i-1])
    p = (prob[proposal]*length(dest[[x[i-1]]]))/(prob[x[i-1]]*length(dest[[proposal]]))
    if(p >= 1) {
      x[i] = proposal
    }
    else {
      if(runif(1)<p)
        x[i] = proposal
      else
        x[i] = x[i-1]
    }
  }
  return(x)
}
```

```
display = function(x) {
  n = length(x)
  par(mfrow=c(1,1))
  plot(0,xlim=c(0,4),ylim=c(0,3),ty='n',xlab='',ylab='',bty='n',xaxt='n',yaxt='n')
  px = c(3,2,1,1,2)
  py = c(2,2,2,1,1)
  lines(px,py)
  lines(px[c(5,2,4)],py[c(5,2,4)])
}
```

```
  points(px,py,cex=10*table(x)/n,col="red",lwd=3)  
}
```

# Chapter 24

## MCMC

*Notes by Murali Haran*

## A Markov chain Monte Carlo example

Summer School in Astrostatistics, Center for Astrostatistics, Penn State University  
Murali Haran, Dept. of Statistics, Penn State University

This module works through an example of the use of Markov chain Monte Carlo for drawing samples from a multidimensional distribution and estimating expectations with respect to this distribution. The algorithms used to draw the samples is generally referred to as the Metropolis-Hastings algorithm of which the Gibbs sampler is a special case. We describe a model that is easy to specify but requires samples from a relatively complicated distribution for which classical Monte Carlo sampling methods are impractical. We describe how to implement a Markov chain Monte Carlo (MCMC) algorithm for this example.

The purpose of this is twofold: First to illustrate how MCMC algorithms are easy to implement (at least in principle) in situations where classical Monte Carlo methods do not work and second to provide a glimpse of practical MCMC implementation issues. It is difficult to work through a truly complex example of a Metropolis-Hastings algorithm in a short tutorial. Our example is therefore necessarily simple but working through it should provide a beginning MCMC user a taste for how to implement an MCMC procedure for a problem where classical Monte Carlo methods are unusable.

### Datasets and other files used in this tutorial:

- [COUP551\\_rates.dat](#)
- [MCMCchpt.R](#)
- [batchmeans.R](#)

### pdf files referred to in this tutorial that give technical details:

- [chptmodel.pdf](#)
- [fullcond.pdf](#)
- [chptmodel2.pdf](#)
- [fullcond2.pdf](#)

## Introduction

**Monte Carlo** methods are a collection of techniques that use pseudo-random (computer simulated) values to estimate solutions to mathematical problems. In this tutorial, we will focus on using Monte Carlo for Bayesian inference. In particular, we will use it for the evaluation of expectations with respect to a probability distribution. Monte Carlo methods can also be used for a variety of other purposes, including estimating maxima or minima of functions (as in likelihood-based inference) but we will not discuss these here.

Monte Carlo works as follows: Suppose we want to estimate an expectation of a function  $g(x)$  with respect to the probability distribution  $f$ . We denote this desired quantity  $m = E_f g(x)$ . Often,  $m$  is analytically intractable (the integration or summation required is too complicated). A Monte Carlo estimate of  $m$  is obtained by simulating  $N$  pseudo-random values from the distribution  $f$ , say  $X_1, X_2, \dots, X_N$  and simply taking the average of  $g(X_1), g(X_2), \dots, g(X_N)$  to estimate  $m$ . As  $N$  (number of samples) gets large, the estimate converges to the true expectation  $m$ .

A toy example to calculate the  $P(-1 < X < 0)$  when  $X$  is a  $N(0,1)$  random variable:

```
xs = rnorm(10000) # simulate 10,000 draws from N(0,1)
```

```
xcount = sum((xs>-1) & (xs<0)) # count number of draws between -1 and 0
xcount/10000 # Monte Carlo estimate of probability
pnorm(0)-pnorm(-1) # Compare it to R's answer (cdf at 0) - (cdf at -1)
```

**Importance sampling:** Another powerful technique for estimating expectations is importance sampling where we produce draws from a different distribution, say  $q$ , and compute a specific weighted average of these draws to obtain estimates of expectations with respect to  $f$ . In this case, A Monte Carlo estimate of  $m$  is obtained by simulating  $N$  pseudo-random values from the distribution  $q$ , say  $Y_1, Y_2, \dots, Y_N$  and simply taking the average of  $g(Y_1)w(Y_1), g(Y_2)w(Y_2), \dots, g(Y_N)w(Y_N)$  to estimate  $m$ , where  $W_1, W_2, \dots, W_N$  are weights obtained as follows:  $W_i = f(Y_i)/q(Y_i)$ . As  $N$  (number of samples) gets large, the estimate converges to the true expectation  $m$ . Often, when normalizing constants for  $f$  or  $q$  are unknown, and for numerical stability, the weights are 'normalized' by dividing the above weights by the sum of all weights (sum over  $W_1, \dots, W_N$ ).

Importance sampling is powerful in a number of situations, including:

- (i) When expectations with respect to several different distributions (say  $f_1, \dots, f_p$ ) are of interest. All these expectations can, in principle, be estimated by using just a single set of samples!
- (ii) When rare event probabilities are of interest so ordinary Monte Carlo would take a huge number of samples for accurate estimates. In such cases, selecting  $q$  appropriately can produce much more accurate estimates with far fewer samples.

Discussions of importance sampling in astronomical Bayesian computation appear in [Lewis & Bridle](#) and [Trotta](#) for cosmological parameter estimation and [Ford](#) for extrasolar planet modeling.

R has random number generators for most standard distributions and there are many more general algorithms (such as rejection sampling) for producing independent and identically distributed (i.i.d.) draws from  $f$ . Another, very general approach for producing non i.i.d. draws (approximately) from  $f$  is the Metropolis-Hastings algorithm.

**Markov chain Monte Carlo :** For complicated distributions, producing pseudo-random i.i.d. draws from  $f$  is often infeasible. In such cases, the Metropolis-Hastings algorithm is used to produce a Markov chain say  $X_1, X_2, \dots, X_N$  where the  $X_i$ 's are *dependent* draws that are *approximately* from the desired distribution. As before, the average of  $g(X_1), g(X_2), \dots, g(X_N)$  is an estimate that converges to  $m$  as  $N$  gets large. The Metropolis-Hastings algorithm is very general and hence very useful. In the following example we will see how it can be used for inference for a model/problem where it would otherwise be impossible to compute desired expectations.

## Problem and model description

### First, a five minute review of Bayesian inference

We begin by specifying a probability model for our data  $Y$  by assuming it is generated from some distribution  $h(\theta; Y)$ , where  $\theta$  is a set of parameters for that distribution. This is written  $Y \sim h(\theta)$ . We want to infer  $\theta$  from the fixed, observed dataset  $Y$ . First, consider likelihood inference. We find a value of  $\theta$  where the likelihood  $L(\theta; Y)$  (which is obtained from the probability distribution  $h(\theta; Y)$ ) is maximized; this is the maximum likelihood estimate (MLE) for  $\theta$ . Now consider Bayesian inference. We assume a prior distribution for  $\theta$ ,  $p(\theta)$ , based on our previous knowledge. This prior may be based on astrophysical insights (e.g. no source can have negative brightness), past astronomical observation (e.g. stars have masses between 0.08-150 solar masses), and/or statistical considerations (e.g. uniform or Jeffreys priors) when it is difficult to obtain

good prior information. Inference is based on the posterior distribution  $Pi(\theta|Y)$  which is proportional to the product of the likelihood and the prior. It is only *proportional* to this product because in reality Bayes theory requires that we write down a denominator (the integral of the product of the likelihood and prior over the parameter space). Fortunately, Markov chain Monte Carlo algorithms avoid computation of this denominator while still producing samples from the posterior  $Pi(\theta|Y)$ . Note that the MCMC methods discussed here are often associated with Bayesian computation, but are really independent methods which can be used for a variety of challenging numerical problems. Essentially, any time samples from a complicated distribution are needed, MCMC may be useful.

Our example uses a dataset from the Chandra Orion Ultradeep Project (COUP). This is a time series of X-ray emission from a flaring young star in the Orion Nebula Cluster. More information on this is available at: [CASt Chandra Flares data set](#). The raw data, which arrives approximately according to a Poisson process, gives the individual photon arrival times (in seconds) and their energies (in keV). The processed data we consider here is obtained by grouping the events into evenly-spaced time bins (10,000 seconds width).

Our goal for this data analysis is to identify the change point and estimate the intensities of the Poisson process before and after the change point. We describe a Bayesian model for this change point problem (Carlin and Louis, 2000). Let  $Y_t$  be the number of occurrences of some event at time  $t$ . The process is observed for times 1 through  $n$  and we assume that there is a change at time  $k$ , i.e., after time  $k$ , the event counts are significantly different (higher or lower than before). The mathematical description of the model is provided in [change point model \(pdf\)](#). While this is a simple model, it is adequate for illustrating some basic principles for constructing an MCMC algorithm.

We first read in the data:

```
chptdat = read.table("http://www.stat.psu.edu/~mharan/MCMCtut/COUP551_rates.dat",skip=1)
```

*Note: This data set is just a convenient subset of the actual data set (see reference below.)*

We can begin with a simple time series plot as exploratory analysis.

```
Y=chptdat[,2] # store data in Y
ts.plot(Y,main="Time series plot of change point data")
```

The plot suggests that the change point may be around 10.

## Setting up the MCMC algorithm

Our goal is to simulate multiple draws from the posterior distribution which is a multidimensional distribution known only upto a (normalizing) constant. From this multidimensional distribution, we can easily derive the conditional distribution of each of the individual parameters (one dimension at a time). This is described, along with a description of the Metropolis-Hastings

# Chapter 25

## MCMC

*Notes by Tom Lored*

## Posterior Sampling & MCMC via Metropolis-Hastings

- 1 Posterior sampling
- 2 Accept-reject algorithm
- 3 Markov chains
- 4 Metropolis-Hastings algorithm

Notes for the Astrostatistics Summer School, India, July 2010  
Tom Loredo <loredo@astro.cornell.edu>

1 / 20

## Posterior Sampling & MCMC

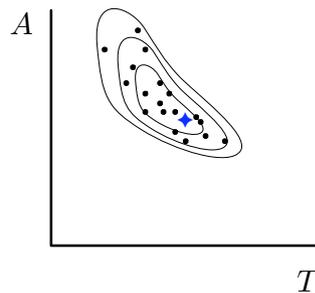
- 1 Posterior sampling
- 2 Accept-reject algorithm
- 3 Markov chains
- 4 Metropolis-Hastings algorithm

2 / 20

## Posterior Sampling

Recall the Monte Carlo algorithm for finding credible regions:

1. Create a RNG that can sample  $\mathcal{P}$  from  $p(\mathcal{P}|D_{\text{obs}})$
2. Draw  $N$  samples; record  $\mathcal{P}_i$  and  $q_i = \pi(\mathcal{P}_i)\mathcal{L}(\mu_i)$
3. Sort the samples by the  $q_i$  values
4. An HPD region of probability  $P$  is the  $\mathcal{P}$  region spanned by the 100 $P$ % of samples with highest  $q_i$



This approach is called *posterior sampling*.

Building a posterior sampler (step 1) is *hard!*

3 / 20

## Posterior Sampling & MCMC

- 1 Posterior sampling
- 2 Accept-reject algorithm
- 3 Markov chains
- 4 Metropolis-Hastings algorithm

4 / 20

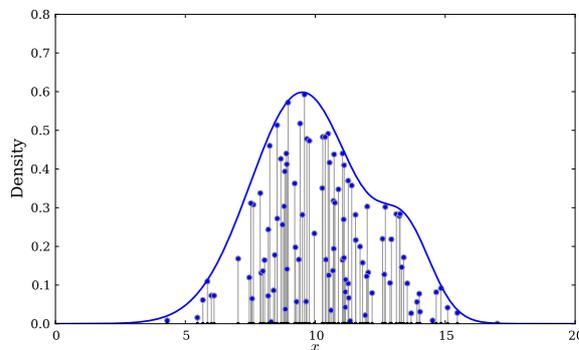
## Basic Accept-Reject Algorithm

Goal: Given  $q(\mathcal{P}) \equiv \pi(\mathcal{P})\mathcal{L}(\mathcal{P})$ , build a RNG that draws samples from the probability density function (*pdf*)

$$f(\mathcal{P}) = \frac{q(\mathcal{P})}{Z} \quad \text{with} \quad Z = \int d\mathcal{P} q(\mathcal{P})$$

The probability for a region under the *pdf* is the *area (volume) under the curve (surface)*.

→ Sample points uniformly in volume under  $q$ ; their  $\mathcal{P}$  values will be draws from  $f(\mathcal{P})$ .



The fraction of samples with  $\mathcal{P}$  ("x" in the fig) in a bin of size  $\delta\mathcal{P}$  is the fractional area of the bin.

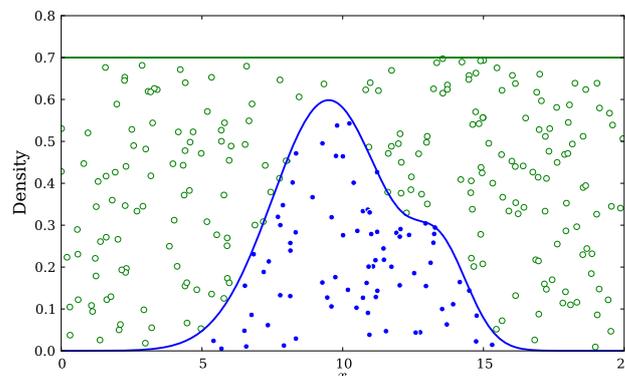
5 / 20

How can we generate points uniformly under the *pdf*?

Suppose  $q(\mathcal{P})$  has compact support: it is nonzero in a finite contiguous region of volume  $V$ .

Generate *candidate* points uniformly in a rectangle enclosing  $q(\mathcal{P})$ .

Keep the points that end up under  $q$ .



6 / 20

### Basic accept-reject algorithm

1. Find an upper bound  $Q$  for  $q(\mathcal{P})$
2. Draw a candidate parameter value  $\mathcal{P}'$  from the uniform distribution in  $V$
3. Draw a uniform random number,  $u$
4. If the ordinate  $uQ < q(\mathcal{P}')$ , record  $\mathcal{P}'$  as a sample
5. Goto 2, repeating as necessary to get the desired number of samples.

Efficiency = ratio of areas (volumes),  $Z/(QV)$ .

### Two issues

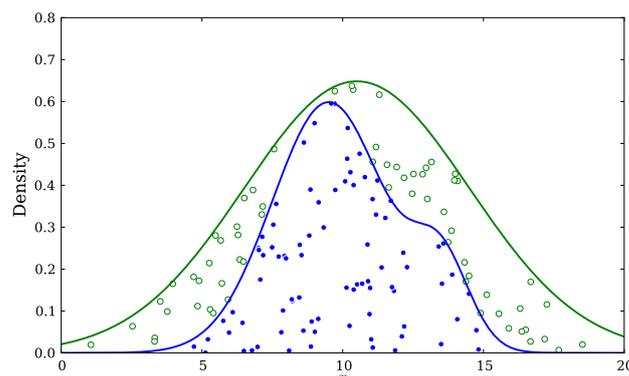
- Increasing efficiency
- Handling distributions with infinite support

7 / 20

## Envelope Functions

Suppose there is a *pdf*  $h(\mathcal{P})$  that we know how to sample from and that roughly resembles  $q(\mathcal{P})$ :

- Multiply  $h$  by a constant  $C$  so  $Ch(\mathcal{P}) \geq q(\mathcal{P})$
- Points with coordinates  $\mathcal{P}' \sim h$  and ordinate  $uCh(\mathcal{P}')$  will be distributed uniformly under  $Ch(\mathcal{P})$
- Replace the hyperrectangle in the basic algorithm with the region under  $Ch(\mathcal{P})$



8 / 20

## Accept-Reject Algorithm

1. Choose an envelope function  $h(\mathcal{P})$  and a constant  $C$  so it bounds  $q$
2. Draw a candidate parameter value  $\mathcal{P}' \sim h$
3. Draw a uniform random number,  $u$
4. If  $q(\mathcal{P}') < Ch(\mathcal{P}')$ , record  $\mathcal{P}'$  as a sample
5. Goto 2, repeating as necessary to get the desired number of samples.

Efficiency = ratio of volumes,  $Z/C$ .

In problems of realistic complexity, the efficiency is intolerably low for parameter spaces of more than a few dimensions.

Key idea: *Propose candidates that may be accepted or rejected*

9 / 20

## Posterior Sampling & MCMC

- ① Posterior sampling
- ② Accept-reject algorithm
- ③ Markov chains
- ④ Metropolis-Hastings algorithm

10 / 20

## Markov Chain Monte Carlo

Accept/Reject aims to produce *independent* samples—each new  $\mathcal{P}$  is chosen irrespective of previous draws.

To enable exploration of complex *pdfs*, let's introduce *dependence*: Choose new  $\mathcal{P}$  points in a way that

- Tends to *move toward* regions with higher probability than current
- Tends to *avoid* lower probability regions

The simplest possibility is a *Markov chain*:

$$\begin{aligned} p(\text{next location} | \text{current and previous locations}) \\ = p(\text{next location} | \text{current location}) \end{aligned}$$

A Markov chain “has no memory.”

11 / 20

## Equilibrium Distributions

Start with some (possibly random) point  $\mathcal{P}_0$ ; produce a sequence of points labeled in order by a “time” index,  $\mathcal{P}_t$ .

Ideally we'd like to have  $p(\mathcal{P}_t) = q(\mathcal{P}_t)/Z$  for each  $t$ . Can we do this with a Markov chain?

To simplify discussion, discretize parameter space into a countable number of *states*, which we'll label by  $x$  or  $y$  (i.e., cell numbers). If  $\mathcal{P}_t$  is in cell  $x$ , we say state  $S_t = x$ .

Focus on *homogeneous Markov chains*:

$$p(S_t = y | S_{t-1} = x) = T(y|x), \quad \text{transition probability (matrix)}$$

Note that  $T(y|x)$  is a probability distribution over  $y$ , and does not depend on  $t$ .

*Aside*: There is no standard notation for any of this—including the order of arguments in  $T$ !

12 / 20

What is the probability for being in state  $y$  at time  $t$ ?

$$\begin{aligned}
 p(S_t = y) &= p(\text{stay at } y) + p(\text{move to } y) - p(\text{move from } y) \\
 &= p(S_{t-1} = y) \\
 &\quad + \sum_{x \neq y} p(S_{t-1} = x) T(y|x) - \sum_{x \neq y} p(S_{t-1} = y) T(x|y) \\
 &= p(S_{t-1} = y) \\
 &\quad + \sum_{x \neq y} [p(S_{t-1} = x) T(y|x) - p(S_{t-1} = y) T(x|y)]
 \end{aligned}$$

If the sum vanishes, then there is an *equilibrium distribution*:

$$p(S_t = y) = p(S_{t-1} = y) \equiv p_{\text{eq}}(y)$$

If we *start* in a state drawn from  $p_{\text{eq}}$ , every subsequent sample will be a (dependent) draw from  $p_{\text{eq}}$ .

13 / 20

## Reversibility/Detailed Balance

A sufficient (but not necessary!) condition for there to be an equilibrium distribution is for *each* term of the sum to vanish:

$$\begin{aligned}
 p_{\text{eq}}(x) T(y|x) &= p_{\text{eq}}(y) T(x|y) \quad \text{or} \\
 \frac{T(y|x)}{T(x|y)} &= \frac{p_{\text{eq}}(y)}{p_{\text{eq}}(x)}
 \end{aligned}$$

This is called the *detailed balance* or *reversibility* condition.

If we set  $p_{\text{eq}} = q/Z$ , and we build a reversible transition distribution for this choice, then *the equilibrium distribution will be the posterior distribution*.

14 / 20

## Convergence

Problem: What about  $p(S_0 = x)$ ?

If we start the chain with a draw from the posterior, every subsequent draw will be from the posterior. But we can't do this!

### Convergence

If the chain produced by  $T(y|x)$  satisfies two conditions:

- It is *irreducible*: From any  $x$ , we can reach any  $y$  with finite probability in a finite # of steps
- It is *aperiodic*: The transitions never get trapped in cycles

then  $p(S_t = s) \rightarrow p_{\text{eq}}(x)$ .

Early samples will show evidence of whatever procedure was used to generate the starting point  $\rightarrow$  discard samples in an initial "burn-in" period.

15 / 20

## Posterior Sampling & MCMC

- 1 Posterior sampling
- 2 Accept-reject algorithm
- 3 Markov chains
- 4 Metropolis-Hastings algorithm

16 / 20

## Designing Reversible Transitions

Set  $p_{\text{eq}}(x) = q(x)/Z$ ; how can we build a  $T(y|x)$  with this as its EQ dist'n?

Steal an idea from accept/reject: Start with a proposal or candidate distribution,  $k(y|x)$ . Devise an accept/reject criterion that leads to a reversible  $T(y|x)$  for  $q/Z$ .

Using any  $k(y|x)$  will not guarantee reversibility. E.g., from a particular  $x$ , the transition rate to a particular  $y$  may be too large:

$$q(x)k(y|x) > q(y)k(x|y) \quad \text{Note: } Z \text{ dropped out!}$$

When this is true, we should use rejections to reduce the rate to  $y$ .

*Acceptance probability:* Accept  $y$  with probability  $\alpha(y|x)$ ; reject it with probability  $1 - \alpha(y|x)$  and stay at  $x$ :

$$T(y|x) = k(y|x)\alpha(y|x) + [1 - \alpha(y|x)]\delta_{y,x}$$

17 / 20

The detailed balance condition is a requirement for  $y \neq x$  transitions, for which  $\delta_{y,x} = 0$ ; it gives a condition for  $\alpha$ :

$$q(x)k(y|x)\alpha(y|x) = q(y)k(x|y)\alpha(x|y)$$

Suppose  $q(x)k(y|x) > q(y)k(x|y)$ ; then we want to suppress  $x \rightarrow y$  transitions, but we want to maximize  $y \rightarrow x$  transitions. So we should set  $\alpha(x|y) = 1$ , and the condition becomes:

$$\alpha(y|x) = \frac{q(y)k(x|y)}{q(x)k(y|x)}$$

If instead  $q(x)k(y|x) < q(y)k(x|y)$ , the situation is reversed: we want  $\alpha(y|x) = 1$ , and  $\alpha(x|y)$  should suppress  $y \rightarrow x$  transitions.

18 / 20

We can summarize the two cases as:

$$\alpha(y|x) = \begin{cases} \frac{q(y)k(x|y)}{q(x)k(y|x)} & \text{if } q(y)k(x|y) < q(x)k(y|x) \\ 1 & \text{otherwise} \end{cases}$$

or equivalently:

$$\alpha(y|x) = \min \left[ \frac{q(y)k(x|y)}{q(x)k(y|x)}, 1 \right]$$

19 / 20

## Metropolis-Hastings algorithm

Given a target quasi-distribution  $q(x)$  (it need not be normalized):

1. Specify a proposal distribution  $k(y|x)$  (make sure it is irreducible and aperiodic).
2. Choose a starting point  $x$ ; set  $t = 0$  and  $S_t = x$
3. Increment  $t$
4. Propose a new state  $y \sim k(y|x)$
5. If  $q(x)k(y|x) < q(y)k(x|y)$ , set  $S_t = y$ ; goto (3)
6. Draw a uniform random number  $u$
7. If  $u < \frac{q(y)k(x|y)}{q(x)k(y|x)}$ , set  $S_t = y$ ; else set  $S_t = x$ ; goto (3)

20 / 20



## Chapter 26

# TIME SERIES ANALYSIS

*Notes by Arnab Chakraborty*

# Time Series Analysis

## Introduction

A time series is a data set collected over time where we suspect some evolution with time. However, 10 CCD bias plates taken over 5 minutes do not qualify as time series data, because we do not expect to have any time-dependent signal embedded in them. The differences between the 10 images are purely random without any temporal evolution. On the other hand for a long 2-hour session we may very well suspect that the CCD is drifting away from its ideal behaviour. So CCD biases taken every 15 minutes over 2 hours do constitute a time series.

### A notation:

- Something that changes with time ( $t$ ) is denoted as a function  $f(t)$ . We are all familiar with this notation.
- Something that changes with chance is denoted by  $f(w)$ , where the symbol  $w$  stands for "chance". It is just a notation, *not* a variable. In a sense  $w$  is the sum total of all effects whose presence cannot be measured but is felt.
- In a time series we have a quantity that changes both with time as well as chance. So we use the notation  $X(t,w)$ .

## What is special about time series?

A very powerful weapon to tackle random variation is to take repeated measurements. But for a time series that is not possible because we cannot go back in time to take a fresh measurement of past values. So we need different techniques. These techniques start out by postulating a mechanism of how time and chance influence the data.

- **Role of time:** This is usually determined by the underlying physics where everything is assumed known except for a few unknown parameters.
- **Role of chance:** Chance enters into the picture either as **intrinsic randomness** (e.g., the experimenter's ignorance or quantum phenomena) or as **measurement error**.

## A simple situation: No intrinsic randomness

Here the sole randomness in the set up is due to measurement errors. A commonly used model for this scenario is

$$X(t,w) = f(t) + e(w),$$

where the form of the time-dependent part  $f(t)$  is governed by the physical model. It is not random. The random error term  $e(w)$  is assumed free of time. This assumption is reasonable when the measuring instrument is calibrated frequently.

A popular example of such a model is the **broken power law**

### Broken power law

R does not have any ready made command to fit a broken power law. So let us write one program to do so. Our running example will use an IR data set about GRB030329 from Bloom (2003).

```
dat = read.table('bloom.dat')
```

The first column is time in days, and the second column is the magnitude.

```
x = log(dat[,1])
y = log(dat[,2])
plot(x,y)
```

First we shall choose a break point say at  $(a,b)$ . Then we shall fit the best broken power law (which is just a broken *linear* law after taking logarithms of both variables) with a knee at that point. Since we have already fixed the knee, all that this fitting requires us to choose are the two slopes to be used on the two sides of the knee.

For this we shift our origin to  $(a,b)$  by subtracting  $a$  from  $x$ , and  $b$  from  $y$ .

```
xnew = x - a
ynew = y - b
```

Next we pick all the points to the left of the knee

```
left = xnew < 0
xleft = xnew[left]
yleft = ynew[left]
```

Similarly, we also collect all the points to the right:

```
xright = xnew[!left]
yright = ynew[!left]
```

Now we have to fit a line *through the origin* (which is now at  $(a,b)$ ) to the left hand points, and then separately to the right hand points. This will give us the best broken power law with the knee constrained to be at  $(a,b)$ .

```
leftfit = lm(yleft ~ xleft - 1)
rightfit = lm(yright ~ xright - 1)
```

How good is this fit? This can be measured by the sum of the squares of all the residuals.

```
sum(leftfit$resid ^ 2) + sum(rightfit$resid ^ 2)
```

We have to choose the knee position  $(a,b)$  to minimise this error. A good way to proceed at this stage is to differentiate the error w.r.t.  $b$ , equate the derivative to 0, and so on. (Incidentally, the error is *not* differentiable w.r.t.  $a$ .) But a quick way is to take a grid of values for  $(a,b)$ , compute the error for each value, and then choose the minimum. Here is some R code to implement this grid search.

```
agrid = seq(1.2, 2.5, 0.1)
bgrid = seq(2.75, 2.95, 0.01)
error = matrix(0, length(agrid), length(bgrid))
for(i in 1:length(agrid)) {
  for(j in 1:length(bgrid)) {
    a = agrid[i]
    b = bgrid[j]
    xnew = x - a
    ynew = y - b
    left = xnew < 0
    xleft = xnew[left]
    yleft = ynew[left]
    xright = xnew[!left]
    yright = ynew[!left]
    fitleft = lm(yleft ~ xleft - 1)
```

```

fitright = lm(yright ~ xright - 1)

error[i,j] = sum(fitleft$res ^ 2)+sum(fitright$res ^ 2)
#readline()
}
}
persp(agrid,bgrid,error)

```

Now we try to find the minimum error.

```

minpos = which.min(error)
row = (minpos-1)%nrow(error)+1
col = floor((minpos-1)/nrow(error))+1
a = agrid[row]
b = bgrid[col]

```

Finally let's plot the fitted line.

```

plot(x,y)
points(a,b,pch=22)
xnew = x - a
ynew = y - b
left = xnew<0
xleft = xnew[left]
yleft = ynew[left]
xright = xnew[!left]
yright = ynew[!left]
leftslope = lm(yleft ~ xleft - 1)$coef
rightslope = lm(yright ~ xright - 1)$coef
abline(a=b - leftslope*a ,b= leftslope,col='blue')
abline(a=b - rightslope*a ,b= rightslope,col='red')

```

## When intrinsic randomness is present

In most non-trivial problems intrinsic randomness is present, and we cannot split  $X(t,w)$  simply into two parts, one involving only  $t$  and the other involving only  $w$ . We need a more complex way to combine the effects of  $t$  and  $w$ .

In order to do this we need to have a clear idea as to the aim of our analysis. Typically the two major aims are

1. **Understanding the structure of the series:** *e.g.*, The period of waxing and waning of the brightness of a star gives us valuable information about exo-planets around it.
2. **Predicting future values:** In many applications of statistics (*e.g.*, in econometrics) this is the primary aim. But this is only of secondary importance in astronomy.

### A case study: Sunspots

Understanding the structure of a time series primarily means detecting cycles in it. Consider the sunspot data as an example.

```

dat = read.table("sspot.dat",head=T)
plot(ts(dat[,1],freq=12,start=c(1749,1)),ty="l",ylab="sunspots (monthly means)")

```

We can see a cyclical pattern. We want to find its period using Fourier Analysis/Spectral Analysis. This is the fundamental idea behind spectroscopy. But there is an important distinction between that and what we are going to do now.

Consider the two configurations for using a spectroscope. In (a) the photons are

allowed to impinge on the CCD first and the CCD signals are fed into the spectroscope.



In (b) the light is first fed to the spectroscope, and CCD comes after that. We all know that the former configuration is of no practical value, because CCD's are slow devices compared to frequency of light waves. Thus CCD's lose all color information. So we need to input the raw light to the spectroscope.

In our statistical analysis however we shall use the configuration shown in (a). The spectroscope here will be a software algorithm that needs numerical input. We shall be working with cycles of large enough periods, so that the slow response of CCD's will pose no problem.

The basic math behind Fourier Transforms (or Fast Fourier Transforms, as its commonly used efficient implementation is called) is beyond the scope of this tutorial. We shall nevertheless remind ourselves of the concept using a toy example.

```

t = 1:200
y1 = sin(2*pi*t/20)
y2 = sin(2*pi*t/12)
y3 = sin(2*pi*t/25)

y123 = y1+y2+y3

par(mfrow=c(4,1))
plot(t,y1,ty="l")
plot(t,y2,ty="l")
plot(t,y3,ty="l")
plot(t,y123,ty="l")
  
```

Notice how chaotic the plot of  $y_{123}$  looks even though it is just a smooth mathematical function. Mere inspection has no hope of ever finding that it is actually made of 3 different sinusoids! Now we shall feed this into our software spectroscope.

```

par(mfrow=c(2,1))
sp = spectrum(y123)
plot(sp$freq, sp$spec, ty="l")
  
```

Observe how the 3 components clearly stands out now!

Now we are in a position to apply our software spectroscope to the sunspot data.

```
dat = read.table("sspot.dat", head=T)
names(dat)
attach(dat)

plot(sspot, ty="l")

sp = spectrum(sspot)
plot(sp$freq, sp$spec, ty="l")
```

We are interested in the strongest period present. This is indicated by the highest peak. We can find it by inspection, or we may let R find it for us:

```
index = which.max(sp$spec)
1/sp$freq[index]
```

## Unevenly sampled data

Its appealing mathematical properties notwithstanding, the Fast Fourier Transformation suffers from one great drawback: it expects the data points to evenly spaced along the time line. But in an observational science like astronomy, where physical conditions like clouds govern availability of data, such data sets are extremely rare. Various modifications to spectral analysis have been suggested to cope with this situation. We shall discuss one of them here: the Lomb-Scargle approach, which is available as a collection of R functions freely downloadable from the web. A good reference for the implementation details is *Numerical Recipes in C* by Press *et al.* To learn the details of R version please see [here](#). The following example is taken from that page, and is explained there.

```
source("LombScargle.R")

unit = "hour" # hourly data
set.seed(19) # make this example reproducible

time = sort( runif(48, 0, 48) )

Expression = cos(2*pi*time/24 + pi/3) # 24 hour period

M = 4*length(time) # often 2 or 4

# Use test frequencies corresponding to
# periods from 8 hours to 96 hours
TestFrequencies = (1/96) + (1/8 - 1/96) * (1:M / M)

# Use Horne & Baliunas' estimate of independent frequencies
Nindependent = NHorneBaliunas(length(Expression))

ComputeAndPlotLombScargle(time, Expression,
  TestFrequencies, Nindependent,
  "Cosine Curve (24 hour period)")
```

## Prediction

Predicting a time series is not often the most important concern in astronomy. Understanding the forces governing the dynamics of a time series is of more importance to the observational astronomer. Nevertheless prediction has its uses, and we shall now learn about some techniques that are geared towards this aim.



Any prediction technique needs to assume, fit and extrapolate some temporal pattern based on the data. It must be borne in mind that such

a temporal pattern is purely for the purpose of prediction. Trying to attach structural significance to it may lead to misleading conclusions. Compare the situation with that of approximating a smooth function with splines. The coefficients used in the spline curves have usually no physical significance.

### Holt-Winters method

This is a very simple (if crude) technique that assumes that the time series is composed of a slowly varying signal plus a cyclical fluctuation of *known period* plus random noise.

The Holt-Winters method is basically a tracking method that tracks the signal and cyclical fluctuation over time using a feedback loop. To understand the algorithm imagine how a robot arm that is capable of only linear movement can trace a curve. It makes a sequence of small linear steps updating the slope at each point based on feedback from its sensor.

Similarly, the Holt-Winters method approximates a complicated time series using a linear component and a cyclical term:

$$a + b h + s_{t+h},$$

where the linear coefficients  $a, b$  and the cyclical term  $s_{t+h}$  are automatically updated using three different feedback loops. These three loops are controlled by three tuning parameters  $\alpha$ ,  $\beta$  and  $\gamma$ . These take values between 0 and 1. If a tuning parameter is chosen to be 0 then the corresponding feedback loop is switched off. The following command, for example, applies the Holt-Winters method to the sun spot data without any cyclical term:

```
hw = HoltWinters(sspot, gamma = 0)
```

To predict the next 200 observations we can use

```
p = predict(hw, 200)
par(mfrow=c(1,1))
ts.plot(sspot, p, col=c("black", "red"))
```

Of course, this is pretty silly prediction, since we suppressed the cyclical fluctuations by force. A more reasonable prediction may be had if we allow default values for all the three tuning parameters (the defaults are not fixed values, they are computed by R based on the data).

```
hw = HoltWinters(sspot)
p = predict(hw, 200)
ts.plot(sspot, p, col=c("black", "red"))
```

This prediction looks much better. But it must be borne in mind that it is a very crude method, and does not produce any error bar. Its absolute simplicity makes it an ideal choice for one-step-ahead predictions computed online (*i.e.*, where the prediction is updated at each step as more data arrive).

## Box-Jenkin's method

The method that we are now going to learn is the most popular statistical method of time series analysis, though its applicability to problems in astronomy is somewhat limited. At the heart of this method sits the somewhat tricky concept of a **stationary** time series. We shall first acquaint ourselves with this concept through some examples.

### Stationarity

In all the time series models encountered so far we have some idea about the temporal evolution of the series (like presence of some linear trend and/or cyclical fluctuations etc). We usually have such an idea from prior knowledge about the physical process generating the data. If we are looking at the light curve of a star with an exo-planet we expect to see periodic ups and downs. But what if we do not have any such prior idea? Then we assume that the distribution of the random variable  $X(t,w)$  is free of  $t$ . Such a time series is called **stationary**.

One must not forget that stationarity is not a property of the data *per se*, rather it describes the state of background information we have about the data. When we say

"a particular time series is stationary"

we are saying that

"we have no background information that leads us to believe that the random behaviour of the series evolves with time."

Considered superficially this may seem to indicate that there is no useful temporal information in the series, and that no meaningful prediction is possible. That this naive interpretation is not correct is the most remarkable fact about Box-Jenkins modelling.

### AutoRegressive Moving Average (ARMA) model

Many time series data are modeled as

$$X_t = \mu + \phi_1 X_{t-1} + \dots + \phi_p X_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q}.$$

This rather hairy object is indeed less pretentious than it looks. Here  $X_t$  is the value of the series at time  $t$ . The model says that part of the randomness in  $X_t$  comes from its dependence on the past values:  $X_{t-1}, \dots, X_{t-p}$ ; and the rest comes from fresh random errors  $\epsilon_t, \dots, \epsilon_{t-q}$  from now and the  $q$  most recent time points.

Viewed in this light, the model (which bears the impressive name **ARMA(p,q)**) says what is applicable in general to most time series. Exactly how the past  $X$ 's and the  $\epsilon$ 's influence  $X_t$  is governed by the coefficients  $\phi$ 's and  $\theta$ 's, which we shall estimate from the data. The only arbitrary part in the model seems to be the choice of  $p$  and  $q$ . While it is reasonable to assume that  $X_t$  is influenced by the past values  $X_{t-1}, \dots, X_{t-p}$ , how do we exactly choose  $p$ ? The same question may be asked for  $q$ .

To make informed guesses about  $p$  and  $q$  based on data, one usually computes two functions called the **Autocorrelation Function (ACF)** and **Partial Autocorrelation Function (PACF)** from the data. Both these functions have

somewhat complicated formulae that do not shed much light on their nature. So instead we shall take an intuitive approach.

## ACF

When we want to measure the degree of association between two quantities a common method is to collect data about them, and compute the correlation. Suppose that we are interested in knowing how much association is there between an observation in a time series and the observation immediately preceding it. The simplest way would be consider all consecutive pairs in the time series data, and compute the correlation based on them. This is basically what is done in time series analysis, and the resulting correlation is called the **autocorrelation at lag 1**.

Had we paired each observation with the one 2 time points behind it, then the resulting correlation would have been called autocorrelation at lag 2.

The autocorrelation function (ACF) is a function  $\gamma(h)$  for  $h=1,2,\dots$  where  $\gamma(h)$  is the autocorrelation at lag  $h$ . One main utility of the ACF is for guessing the value of  $q$ . For an **ARMA(0,q)** model (which is also called an **MA(q)** model) typically the first  $q$  values of the ACF tend to be large, while the remaining are smaller in comparison. In practice it is often not clear at all how to decide which ACF's are large and which are not. So the ACF only provides a very rough guidance, that we shall learn to polish soon.

To compute ACF for a time series data set we can use the `acf` command as follows.

```
|acf(sspot)
```

Well, here the ACF is dying out, but rather slowly. This is indicative of  $p \neq 0$  and  $q = 0$ . We shall get a further confirmation for this guess when we consider the *partial autocorrelation function* below.

## PACF

Correlation measures association between two variables. However, it does not tell us if the association is direct or indirect. An example if indirect association is where two variables  $X, Y$  are both associated with a common third variable  $Z$ . For example, temperature ( $X$ ) outside the telescope may found to have association with the amount ( $Y$ ) of noise in the CCD. A scientist may guess that the association occurs indirectly via the temperature ( $Z$ ) of the CCD unit. Merely finding the correlation between  $X$  and  $Y$  is not enough to test the scientist's guess. We can, instead, wrap the CCD unit in very good temperature control system (making  $Z$  a rock-solid constant) and *then* compute the correlation between  $X$  and  $Y$ . If now the correlation drops down to zero, we indeed have reason to support the scientist's guess. The correlation we computed just now is called the **partial correlation** between  $X$  and  $Y$  fixing  $Z$ . It measures the association between  $X$  and  $Y$  *other than via*  $Z$ .

In the context of time series we compute partial correlation between  $X_t$  and  $X_{t+h}$  fixing all the  $X$ 's inbetween. This is the value partial autocorrelation function (PACF) at lag  $h$ .

Just as the ACF helps us to guess  $q$ , the PACF helps us guess the value of  $p$ : the first  $p$  PACF's tend to be larger than the rest.

The R command `pacf` computes PACF (surprise!).

```
|pacf(sspot)
```

The two blue margins provide an aid to deciding whether a PACF value is "large" or not. Values within or barely peeping out of the margins may be considered "large", the rest being "small".

Our plot indicates that  $p=5$  and  $q=0$  might be good guesses to start with.

### Fitting the model

Once we have made a rough guess about  $p$  and  $q$  we need to estimate the coefficients  $\theta$ 's and  $\phi$ 's. The elaborate mathematical machinery required for this is conveniently hidden from our view by the `arima` command of R. Note the extra 'i' in `arima`. Actually R works with a more general model called ARIMA (AutoRegressive Integrated Moving Average) of which ARMA is a special case. Here is an example of the `arima` command in use:

```
|fit=arima(sspot,order=c(5,0,0))
```

Note the triple `c(5,0,0)`. The first 5 is the value of  $p$ , the last 0 is that of  $q$ . The middle 0 is not used in an ARMA model (it is used in the more general ARIMA model).

This rather benign-looking command launches a lot of computation in the background, and if successful, dumps the results in the variable `fit`.

There are various things that we can do with this object. Possibly the first thing that we should do is to take a look at what is inside it.

```
|fit
```

But before we put it to any serious use like prediction, we should better make sure that it is indeed a good fit. One way to check this is by plotting the residuals:

```
|plot(fit$resid)
```

Preferably the plot should show a series of small values without any pattern. Any pattern in the residual plot indicates that the fitted model has been inadequate to explain that pattern. This is the case here. The regular periodic fluctuations are too strong to be called "patternless".

What we do to remedy a bad fit depends heavily on the particular data set, and is something of an art. One thing to try out is to change  $p$  or  $q$  slightly and see if the fit improves. This comparison is facilitated by the **Akaike Information Criterion (AIC)** which is a summarisation of the goodness of fit of the fitted model in terms of a single number. If we have two competing models for the same data set, then the one with the smaller AIC is the better fit. The command to compute AIC in R for the fitted model stored in `fit` is

```
|AIC(fit)
```

AIC cannot be compared for models based on different data sets. As a result there is no universal threshold against which to rate a single AIC value on its own.

It may happen that no ARMA model fits the data at hand. In this case we have to

migrate to some different (and more complicated) class of models. And above all, we must not forget that there are just not enough types of models in statistics to fit all kinds of time series data occurring in practice!



## Chapter 27

# TIME SERIES ANALYSIS

*Notes by Eric Feigelson*

## Time Series Analysis

### Outline

- 1 Time series in astronomy
- 2 Frequency domain methods
- 3 Time domain methods
- 4 References

### Time series in astronomy

- Periodic phenomena: binary orbits (stars, extrasolar planets); stellar rotation (radio pulsars); pulsation (helioseismology, Cepheids)
- Stochastic phenomena: accretion (CVs, X-ray binaries, Seyfert gals, quasars); scintillation (interplanetary & interstellar media); jet variations (blazars)
- Explosive phenomena: thermonuclear (novae, X-ray bursts), magnetic reconnection (solar/stellar flares), star death (supernovae, gamma-ray bursts)

### Difficulties in astronomical time series

#### Gapped data streams:

Diurnal & monthly cycles; satellite orbital cycles; telescope allocations

#### Heteroscedastic measurement errors:

Signal-to-noise ratio differs from point to point

#### Poisson processes:

Individual photon/particle events in high-energy astronomy

## Important Fourier Functions

### Discrete Fourier Transform

$$d(\omega_j) = n^{-1/2} \sum_{t=1}^n x_t \exp(-2\pi i t \omega_j)$$

$$d(\omega_j) = n^{-1/2} \sum_{t=1}^n x_t \cos(2\pi i \omega_j t) - i n^{-1/2} \sum_{t=1}^n x_t \sin(2\pi i \omega_j t)$$

### Classical (Schuster) Periodogram

$$I(\omega_j) = |d(\omega_j)|^2$$

### Spectral Density

$$f(\omega) = \sum_{h=-\infty}^{h=\infty} \exp(-2\pi i \omega h) \gamma(h)$$

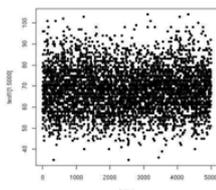
Fourier analysis reveals nothing of the evolution in time, but rather reveals the variance of the signal at different frequencies.

It can be proved that the classical periodogram is an estimator of the spectral density, the Fourier transform of the autocovariance function.

Formally, the probability of a periodic signal in Gaussian noise is  $P \propto e^{d(\omega_j)/\sigma^2}$ .

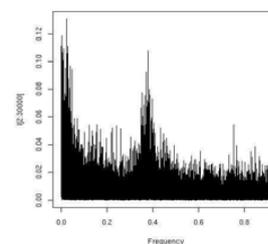
## Ginga observations of X-ray binary GX 5-1

GX 5-1 is a binary star system with gas from a normal companion accreting onto a neutron star. Highly variable X-rays are produced in the inner accretion disk. XRB time series often show 'red noise' and 'quasi-periodic oscillations', probably from inhomogeneities in the disk. We plot below the first 5000 of 65,536 count rates from Ginga satellite observations during the 1980s.



```
gx=scan("../Desktop/CASt/SumSch/TSA/GX.dat")
t=1:5000
plot(t,gx[1:5000],pch=20)
```

Fast Fourier Transform of the GX 5-1 time series reveals the 'red noise' (high spectral amplitude at small frequencies), the QPO (broadened spectral peak around 0.35), and white noise.



```
f = 0.32768/65536
I = (4/65536)*abs(fft(gx)/sqrt(65536))^2
plot(f[2:60000],I[2:60000],type="l",xlab="Frequency")
```

### Limitations of the spectral density

But the classical periodogram is not a good estimator! E.g. it is formally 'inconsistent' because the number of parameters grows with the number of datapoints. The discrete Fourier transform and its probabilities also depends on several strong assumptions which are rarely achieved in real astronomical data: evenly spaced data of infinite duration with a high sampling rate (Nyquist frequency), Gaussian noise, single frequency periodicity with sinusoidal shape and stationary behavior. Formal statement of strict stationarity:

$$P\{x_{t_1} \leq c_1, \dots, x_{t_K} \leq c_K\} = P\{x_{t_1+h} \leq c+1, \dots, x_{t_K+h} \leq c_K\}.$$

Each of these constraints is violated in various astronomical problems. Data spacing may be affected by daily/monthly/orbital cycles. Period may be comparable to the sampling time. Noise may be Poissonian or quasi-Gaussian with heavy tails. Several periods may be present (e.g. helioseismology). Shape may be non-sinusoidal (e.g. elliptical orbits, eclipses). Periods may not be constant (e.g. QPOs in an accretion disk).

### Improving the spectral density I

The estimator can be improved with **smoothing**.

$$\hat{f}(\omega_j) = \frac{1}{2m_1} \sum_{k=-m}^m I(\omega_{j-k}).$$

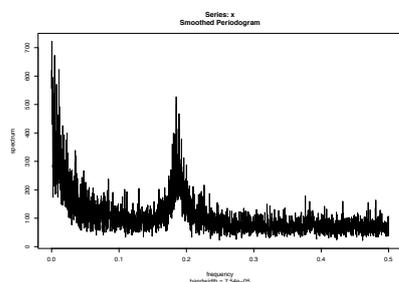
This reduces variance but introduces bias. It is not obvious how to choose the smoothing bandwidth  $m$  or the smoothing function (e.g. Daniell or boxcar kernel).

**Tapering** reduces the signal amplitude at the ends of the dataset to alleviate the bias due to leakage between frequencies in the spectral density produced by the finite length of the dataset.

Consider for example the cosine taper

$$h_t = 0.5[1 + \cos(2\pi(t - \bar{t})/n)]$$

applied as a weight to the initial and terminal  $n$  datapoints. The Fourier transform of the taper function is known as the spectral window. Other widely used options include the Fejer and Parzen windows and multitapering. Tapering decreases bias but increases



```
postscript(file="/Desktop/GX_em_tap_fft.eps")
k = kernel("modified.daniell", c(7,7))
spec = spectrum(gx, k, method="pgram", taper=0.3, fast=TRUE, detrend=TRUE, log="no")
dev.off()
```

### Improving the spectral density II

**Pre-whitening** is another bias reduction technique based on removing (filtering) strong signals from the dataset. It is widely used in radio astronomy imaging where it is known as the CLEAN algorithm, and has been adapted to astronomical time series (Roberts et al. 1987).

A variety of **linear filters** can be applied to the time domain data prior to spectral analysis. When aperiodic long-term trends are present, they can be removed by spline fitting (high-pass filter). A kernel smoother, such as the moving average, will reduce the high-frequency noise (low-pass filter). Use of a parametric autoregressive model instead of a nonparametric smoother allows likelihood-based model selection (e.g. BIC).

### Improving the spectral density III

Harmonic analysis of unevenly spaced data is problematic due to the loss of information and increase in aliasing.

The **Lomb-Scargle periodogram** is widely used in astronomy to alleviate aliasing from unevenly spaced:

$$d_{LS}(\omega) = \frac{1}{2} \left( \frac{[\sum_{t=1}^n x_t \cos \omega(x_t - \tau)]^2}{\sum_{i=1}^n \cos^2 \omega(x_t - \tau)} + \frac{[\sum_{t=1}^n x_t \sin \omega(x_t - \tau)]^2}{\sum_{i=1}^n \sin^2 \omega(x_t - \tau)} \right)$$

where  $\tan(2\omega\tau) = (\sum_{i=1}^n \sin 2\omega x_t) (\sum_{i=1}^n \cos 2\omega x_t)^{-1}$

$d_{LS}$  reduces to the classical periodogram  $d$  for evenly-spaced data. Bretthorst (2003) demonstrates that the Lomb-Scargle periodogram is the unique sufficient statistic for a single stationary sinusoidal signal in Gaussian noise based on Bayes theorem assuming simple priors.

### Some other methods for periodicity searching

**Phase dispersion measure** (Stellingwerf 1972) Data are folded modulo many periods, grouped into phase bins, and intra-bin variance is compared to inter-bin variance using  $\chi^2$ . Non-parametric procedure well-adapted to unevenly spaced data and non-sinusoidal shapes (e.g. eclipses). Very widely used in variable star research, although there is difficulty in deciding which periods to search (Collura et al. 1987).

**Minimum string length** (Dworetzky 1983) Similar to PDM but simpler: plots length of string connecting datapoints for each period. Related to the Durbin-Watson roughness statistic in econometrics.

**Rayleigh** and  $Z_n^2$  tests (Leahy et al. 1983) for periodicity search Poisson distributed photon arrival events. Equivalent to Fourier spectrum at high count rates.

**Bayesian periodicity search** (Gregory & Loredó 1992) Designed for non-sinusoidal periodic shapes observed with Poisson events. Calculates odds ratio for periodic over constant model and most probable shape.

### Conclusions on spectral analysis

For challenging problems, smoothing, multitapering, linear filtering, (repeated) pre-whitening and Lomb-Scargle can be used together. Beware that aperiodic but autoregressive processes produce peaks in the spectral densities. Harmonic analysis is a complicated 'art' rather than a straightforward 'procedure'.

It is extremely difficult to derive the significance of a weak periodicity from harmonic analysis. Do not believe analytical estimates (e.g. exponential probability), as they rarely apply to real data. It is essential to make simulations, typically permuting or bootstrapping the data keeping the observing times fixed. Simulations of the final model with the observation times is also advised.

### Nonstationary time series

Non-stationary periodic behaviors can be studied using **time-frequency Fourier analysis**. Here the spectral density is calculated in time bins and displayed in a 3-dimensional plot.

**Wavelets** are now well-developed for non-stationary time series, either periodic or aperiodic. Here the data are transformed using a family of non-sinusoidal orthogonal basis functions with flexibility both in amplitude and temporal scale. The resulting wavelet decomposition is a 3-dimensional plot showing the amplitude of the signal at each scale at each time. Wavelet analysis is often very useful for noise thresholding and low-pass filtering.

## Nonparametric time domain methods

### Autocorrelation function

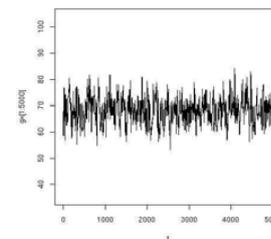
$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)} \text{ where}$$

$$\hat{\gamma}(h) = \frac{\sum_{i=1}^{n-h} (x_{i+h} - \bar{x})(x_i - \bar{x})}{n}$$

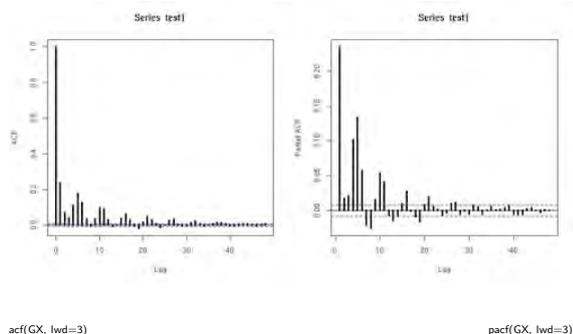
This sample ACF is an estimator of the correlation between the  $x_t$  and  $x_{t-h}$  in an evenly-spaced time series lags. For zero mean, the ACF variance is  $Var \hat{\rho} = [n-h]/[n(n+2)]$ .

The partial autocorrelation function (PACF) estimates the correlation with the linear effect of the intermediate observations,  $x_{t-1}, \dots, x_{t-h+1}$ , removed. Calculate with the Durbin-Levinson algorithm based on an autoregressive model.

### Kernel smoothing of GX 5+1 time series Normal kernel, bandwidth = 7 bins



### Autocorrelation functions



## Parametric time domain methods: ARMA models

### Autoregressive moving average model

Very common model in human and engineering sciences, designed for aperiodic autocorrelated time series (e.g. 1/f-type 'red noise'). Easily fit by maximum-likelihood. Disadvantage: parameter values are difficult to interpret physically.

$$\text{AR}(p) \text{ model } x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + w_t$$

$$\text{MA}(q) \text{ model } x_t = w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q}$$

The AR model is recursive with memory of past values. The MA model is the moving average across a window of size  $q + 1$ . ARMA(p,q) combines these two characteristics.

## State space models

Often we cannot directly detect  $x_t$ , the system variable, but rather indirectly with an observed variable  $y_t$ . This commonly occurs in astronomy where  $y$  is observed with measurement error (errors-in-variable or EIV model). For AR(1) and errors  $v_t = N(\mu, \sigma)$  and  $w_t = N(\nu, \tau)$ ,

$$y_t = Ax_t + v_t \quad x_t = \phi_1 x_{t-1} + w_t$$

This is a state space model where the goal is to estimate  $x_t$  from  $y_t$ ,  $p(x_t|y_t, \dots, y_1)$ . Parameters are estimated by maximum likelihood, Bayesian estimation, Kalman filtering, or prediction.

## GX 5+1 modeling

```
ar(x = GX, method = "mle")
```

Coefficients:

```
1 2 3 4 5 6 7 8
```

```
0.21 0.01 0.00 0.07 0.11 0.05 -0.02 -0.03
```

```
arima(x = GX, order = c(6, 2, 2))
```

Coefficients:

```
ar1 ar2 ar3 ar4 ar5 ar6 ma1 ma2
```

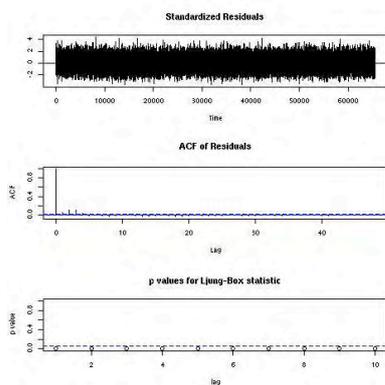
```
0.12 -0.13 -0.13 0.01 0.09 0.03 -1.93 0.93
```

```
Coeff s.e. = 0.004
```

```
 $\sigma^2 = 10^2$ 
```

```
log L = -244446.5
```

```
AIC = 488911.1
```



Although the scatter is reduced by a factor of 30, the chosen model is not adequate: Ljung-Box test shows significant correlation in the residuals. Use AIC for model selection.

## Other time domain models

- Extended ARMA models: VAR (vector autoregressive), ARFIMA (ARIMA with long-memory component), GARCH (generalized autoregressive conditional heteroscedastic for stochastic volatility)
- Extended state space models: non-stationarity, hidden Markov chains, etc. MCMC evaluation of nonlinear and non-normal (e.g. Poisson) models

### Statistical texts and monographs

- D. Brillinger, Time Series: Data Analysis and Theory, 2001  
 C. Chatfield, The Analysis of Time Series: An Introduction, 6th ed., 2003  
 G. Kitagawa & W. Gersch, Smoothness Priors Analysis of Time Series, 1996  
 J. F. C. Kingman, Poisson Processes, 1993  
 J. K. Lindsey, Statistical Analysis of Stochastic Processes in Time, 2004  
 S. Mallat, A Wavelet Tour of Signal Processing, 2nd ed, 1999  
 M. B. Priestley, Spectral Analysis and Time Series, 2 vol, 1981  
 R. H. Shumway and D. S. Stoffer, Time Series Analysis and Its Applications  
 (with R examples), 2nd Ed., 2006

### Astronomical references

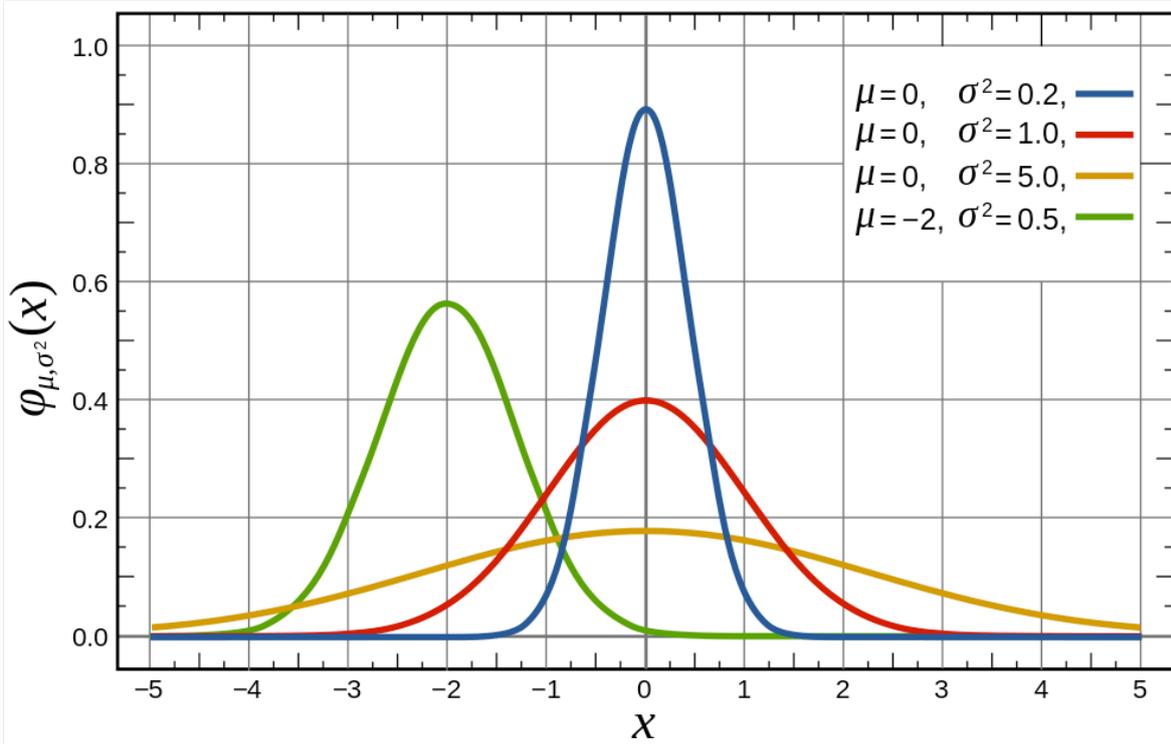
- Bretthorst 2003, "Frequency estimation and generalized Lomb-Scargle periodograms", in Statistical Challenges in Modern Astronomy  
 Collura et al 1987, "Variability analysis in low count rate sources", ApJ 315, 340  
 Dworetsky 1983, "A period-finding method for sparse randomly spaced observations ....", MNRAS 203, 917  
 Gregory & Loredó 1992, "A new method for the detection of a periodic signal of unknown shape and period", ApJ 398, 146  
 Kovacs et al. 2002, "A box-fitting algorithm in the search for periodic transits", A&A 391, 369  
 Leahy et al. 1983, "On searches for periodic pulsed emission: The Rayleigh test compared to epoch folding", ApJ 272, 256  
 Roberts et al. 1987, "Time series analysis with CLEAN ...", AJ 93, 968  
 Scargle 1982, "Studies in astronomical time series, II. Statistical aspects of spectral analysis of unevenly spaced data", ApJ 263, 835  
 Scargle 1998, "Studies in astronomical time series, V. Bayesian Blocks, a new method to analyze structure in photon counting data, ApJ 504, 405

- Stellingwerf 1972, "Period determination using phase dispersion measure", ApJ 224, 953  
 Vio et al. 2005, "Time series analysis in astronomy: Limits and potentialities, A&A 435, 773

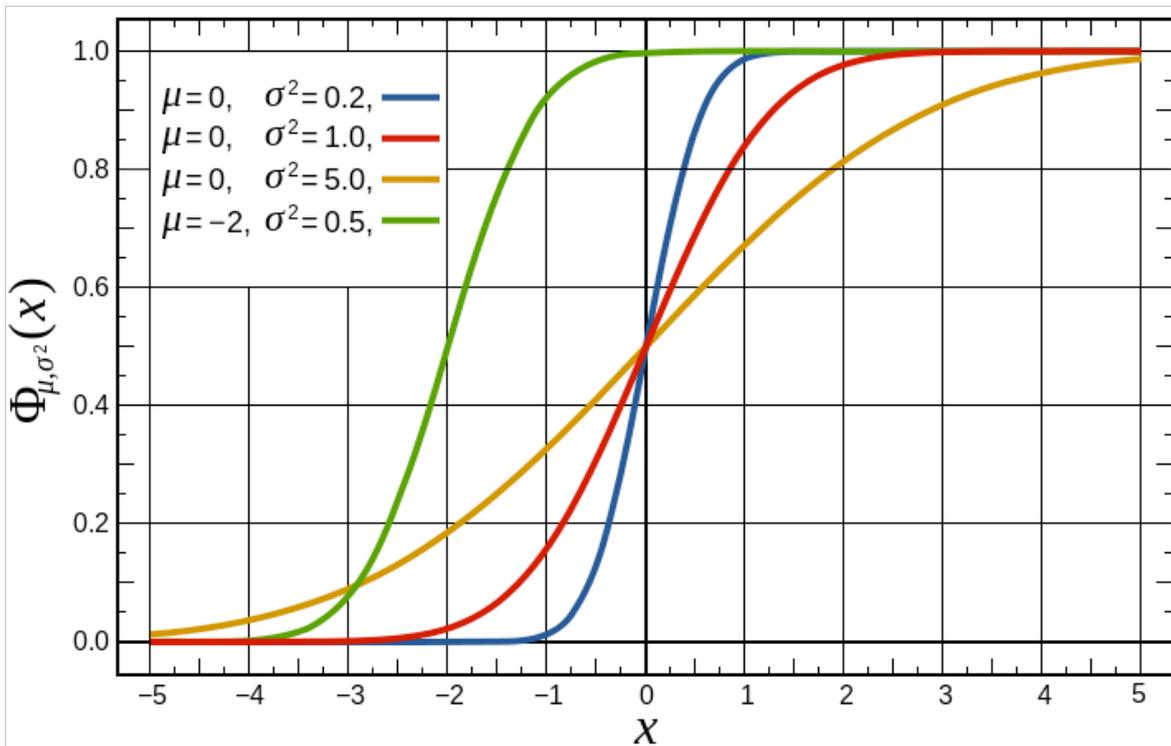
## Chapter 28

### APPENDIX A: DISTRIBUTIONS

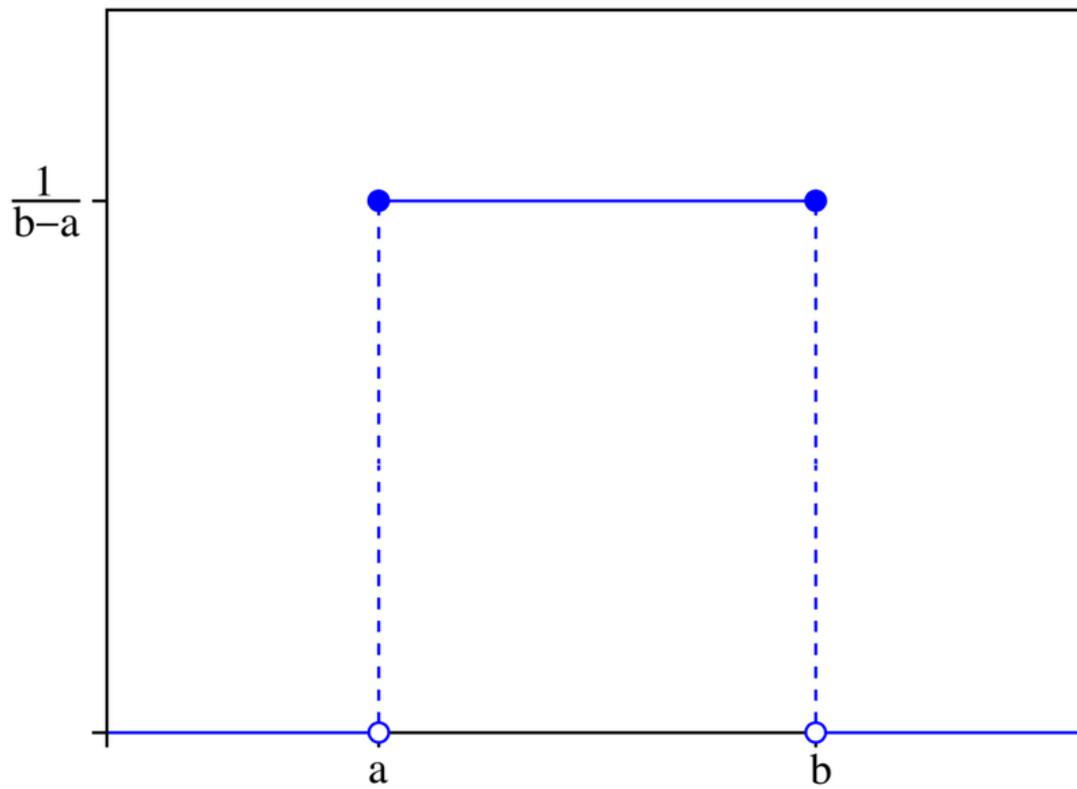
## Normal Distribution



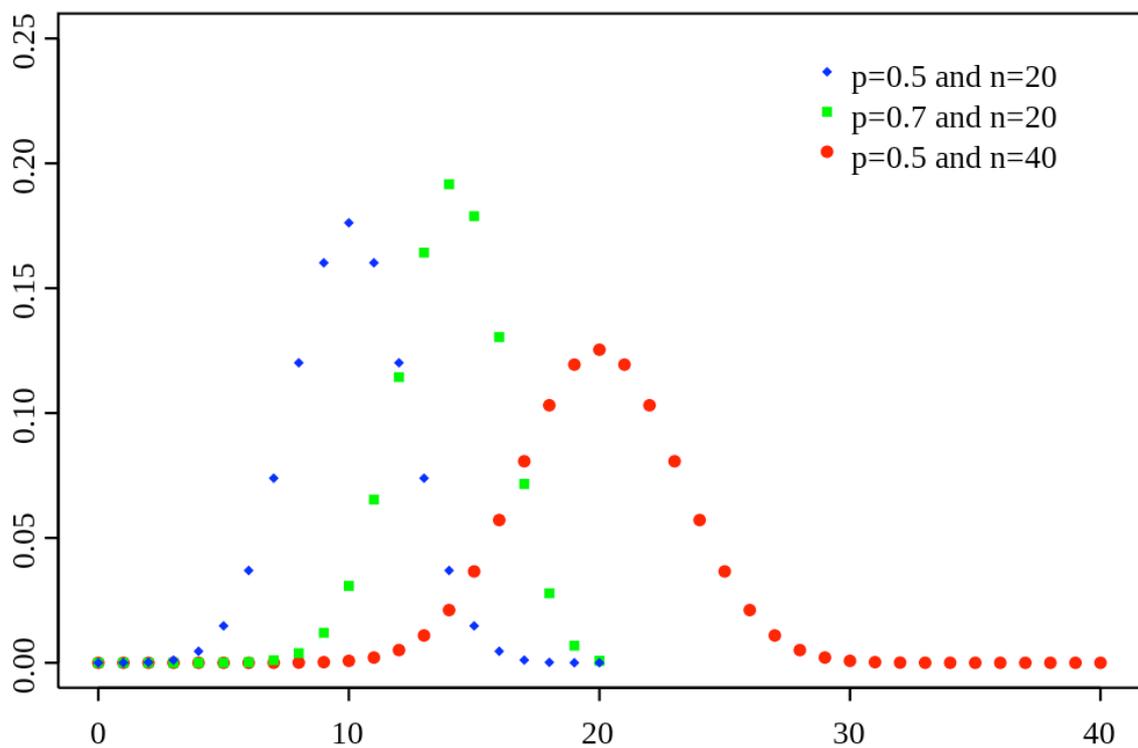
## Cumulative Distribution Function



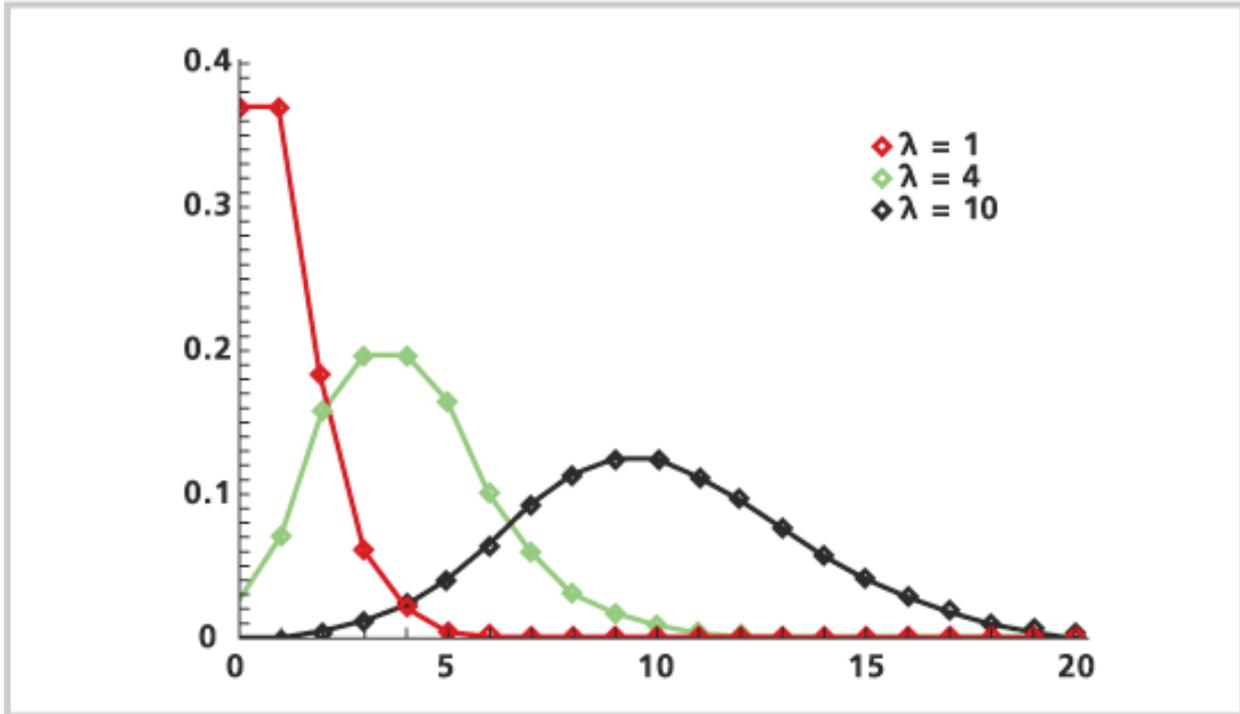
## Uniform Distribution



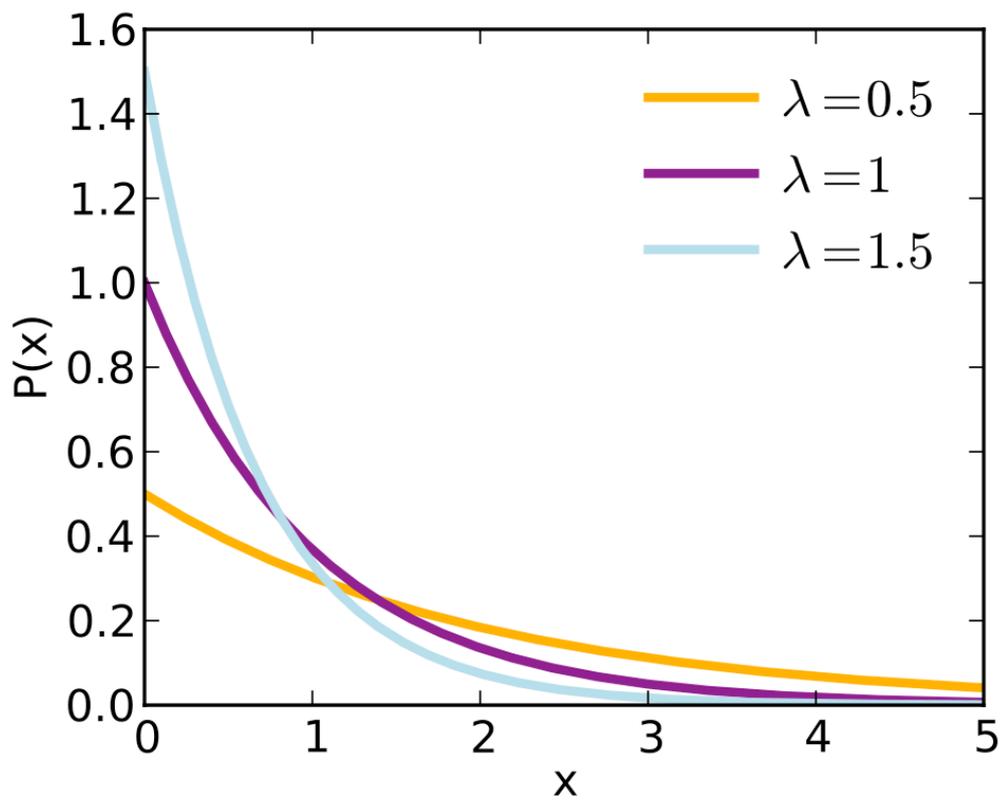
## Binomial Distribution



## Poisson Distribution



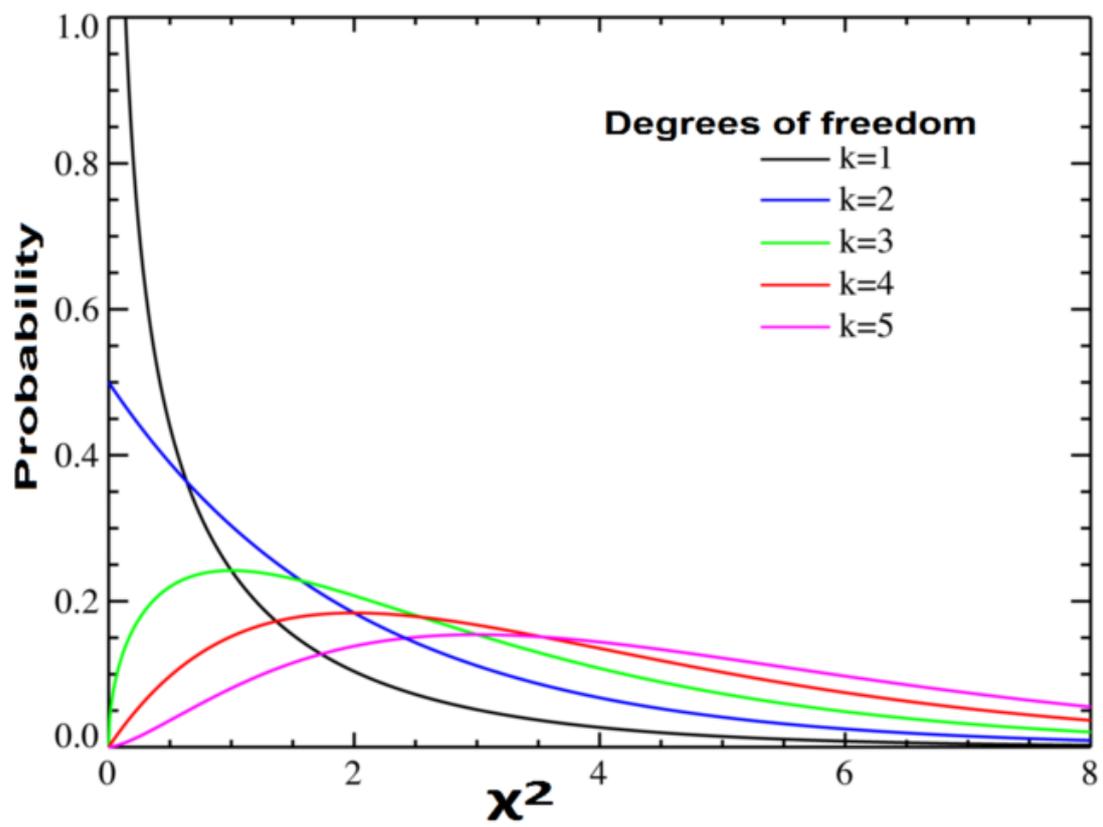
## Exponential Distribution



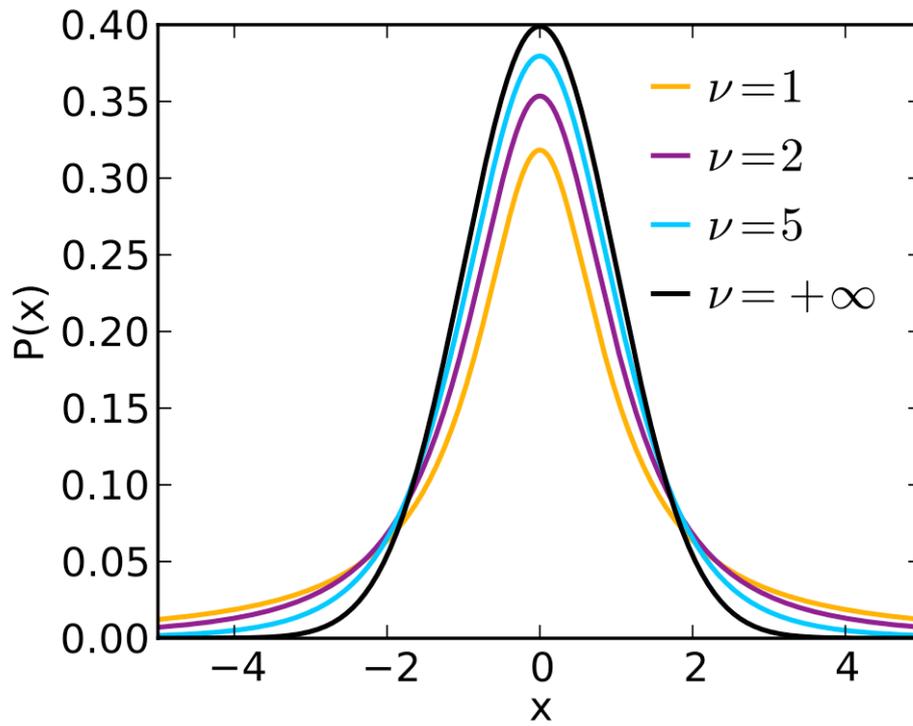
## Power-law Distribution



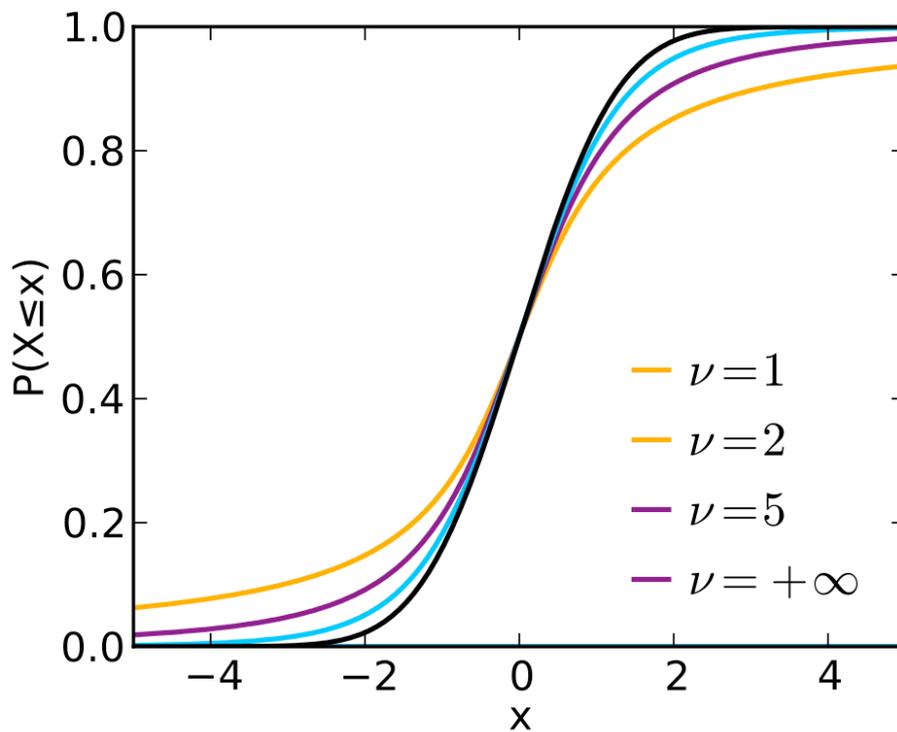
## Chi-square Distribution



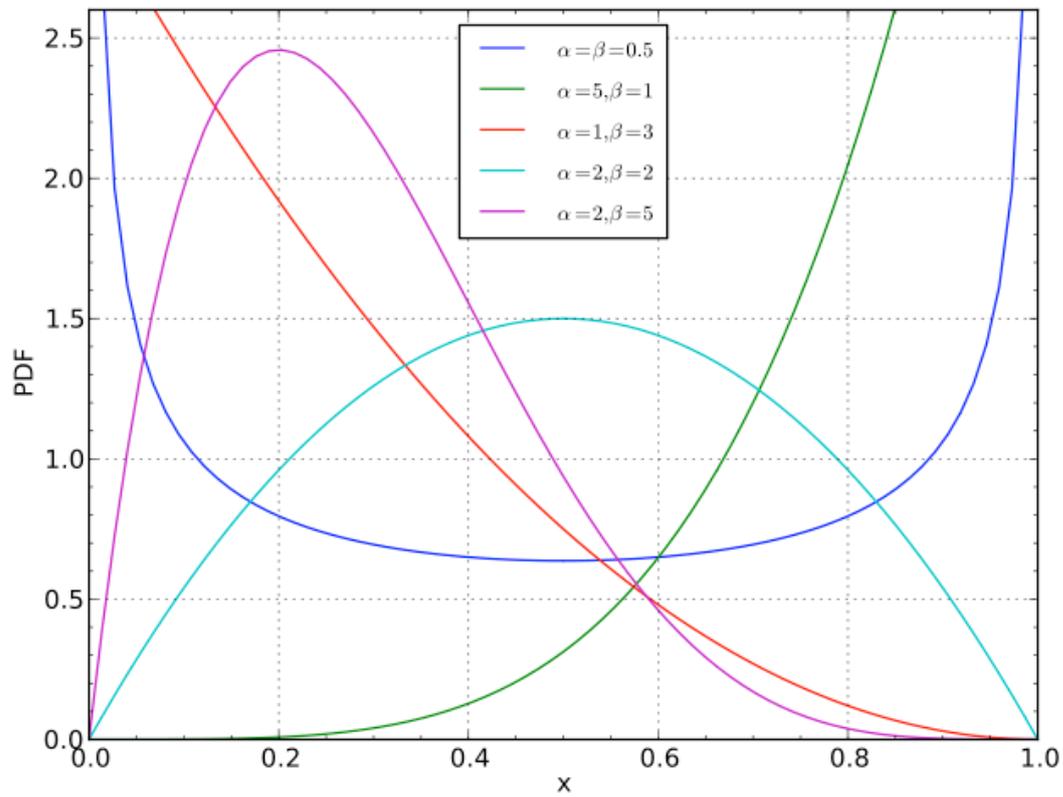
## Student-t Probability Density



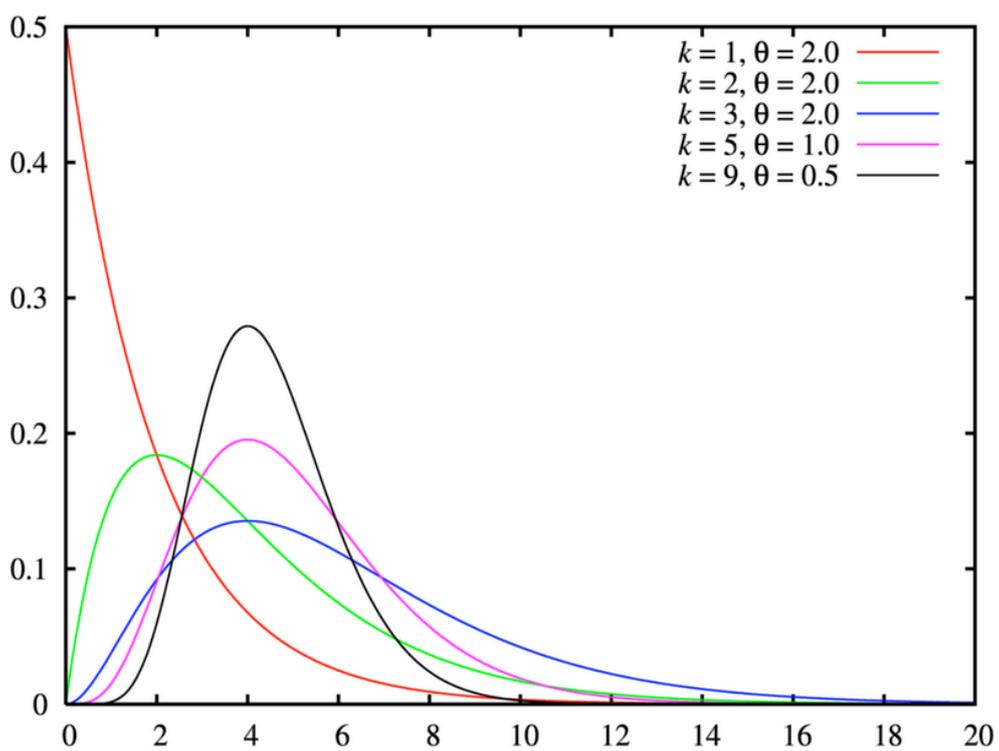
## Student-t Cumulative Distribution



## Beta Distribution



## Gamma Distribution





## Chapter 29

### APPENDIX B: JARGON

# Astro Jargon for Statisticians

8/31/2004

**Note: An equivalent list, with Statistics terms described for Astronomers, is available at [www.ics.uci.edu/~dvd/Astro/stat-jargon-for-astro.pdf](http://www.ics.uci.edu/~dvd/Astro/stat-jargon-for-astro.pdf)**

## [Å] and [keV]

Ångstrom, a unit of length equal to  $10^{-8}$  cm. X-ray wavelengths generally range from O(1) to O(100) Å. The equivalent unit of photon energy is [keV], where  $1 \text{ eV} = 1.609 \cdot 10^{-12}$  ergs, and  $[\text{Å}] = 12.3985 / [\text{keV}]$ .

## Abundance and Metallicity

The relative number of an element wrt that of Hydrogen is the abundance. It is usually written  $A(Z) = N(Z) / N(H)$  for a simple ratio, or  $[Z] = \log_{10}(A(Z) / A(Z)_{\text{Sun}})$  when denoting abundance relative to the Solar composition. The metallicity is the abundance of Fe, which is the most dominant source of emission. Abundance variations are usually reported with reference to Fe, e.g.,  $[Ne/Fe]$ ,  $[O/Fe]$ , etc. (Note that astronomers refer to all elements other than H and He as "metals".)

## ARF

Auxiliary Response File (aka Ancillary Response File). Encodes the telescope effective area. Includes the combined telescope/filter/detector areas and efficiencies as a function of energy averaged over time. When the input flux spectrum is multiplied by the ARF the result is the distribution of counts that would be seen by a detector with perfect (i.e. infinite) energy resolution. The RMF is then needed to produce the final observed spectrum.

## Continuum

Spectra usually consist of emission lines, absorption features, and continua. The continuum can arise due to blackbody emission, atomic bound-free transitions, bremsstrahlung radiation, cyclotron or synchrotron emission.

## Dither

While observing a celestial source, X-ray telescopes do not maintain a fixed pointing direction, but instead 'dithers' around the Sky. *Chandra*'s dither pattern is a Lissajous figure, with irrational periods in the cardinal directions to make a non-closed Lissajous pattern.

## Effective Area

The product of the telescope mirror geometric area, reflectivity (which is a strong function of energy), off-axis vignetting (also a function of energy as well as off-axis angle), detector quantum efficiency (including any filters), which depends on energy and position on the detector, and [if applicable] diffraction grating efficiency (which is a function of order and energy).

## Emissivity

The intrinsic strength of an atomic transition that produces a spectral line. The term encapsulates all the atomic data information needed to calculate the flux. It is often generalized to also refer to continuum processes. Units are usually  $[10^{-23} \text{ ergs cm}^3 \text{ s}^{-1}]$ . Sometimes also presented as  $[\text{ph cm}^3 \text{ s}^{-1}]$ , and especially for

continuum emissivities, [(erg|ph) cm<sup>3</sup> s<sup>-1</sup> Å<sup>-1</sup>]

### **Emission Measure**

A measure of the "amount of material" of a plasma available to produce the observed flux, the product of the square of the electron number density and the volume of emission, with units [cm<sup>-3</sup>]. Often, because observations are carried out along a line of sight, the cross-section area is taken out of the expression and the units become [cm<sup>-5</sup>].

### **Event**

X-ray astronomy instruments record a distinct signal from each individual photon they detect. As a result X-ray data are stored event by event, which retains all the information and allows great flexibility of analysis. Every X-ray "event" (a general term for a detection; may refer to a celestial photon or a background cosmic ray) is characterized by a "pulse height" (PHA) that encodes the energy of the incoming photon; a time of arrival; a grade describing the quality of the event; and typically two position coordinates.

### **Exposure maps and Instrument maps**

An array specifying the 'amount of exposure' at each image pixel for a given observation. For Chandra, the exposure map includes the effective area and has units of [cm<sup>2</sup>] or [cm<sup>2</sup> s]. This quantity is essential to determine the flux or brightness of a celestial source from the observed counts. A related item is the Instrument map, which describes the efficiency of the detector at each detector pixel. Generally, an exposure map is derived after applying aspect dither to the instrument map.

### **Image**

A two-way frequency table of observed counts as a function of location. The location coordinates are either rooted in physical detector space, or in inferred direction of arrival of the photons in angular coordinates on the sky.

### **lambda**

*lambda* refers to the wavelength of a photon, usually in [Å], and *E* refers to its energy, usually in [keV]. For grating data, because the RMF is almost diagonal, these also refer to the detector bin. Low-resolution spectra are placed in channel bins, whose boundaries are mapped to approximate ranges of photon energies via a gain map.

### **Lines**

Atoms, ions, and molecules emit photons at characteristic energies, with each producing a unique set of lines that serve as "bar codes" that identify the element. Lines can be seen either in emission (as enhancements over a smooth continuum) or in absorption (as dips from a smooth continuum).

### **PHA and PI**

Pulse Height Amplitude/Pulse Invariant Channel

1. Engineering unit describing the integrated charge per pixel from an event recorded in a detector. In early electronic devices, this was the size of the pulse.
2. The PHA value in event files is the total pulse height of an event. For a given location, a gain table is used to map the PHA of an event to a nominal energy value, converting the PHA into a Pulse Invariant (PI) channel.
3. "PHA File": Standard file type for a histogram of counts vs. spectral channel

(PHA, ADU, diffraction angle, wavelength, or other).

### **Photons v/s Counts**

By convention, the term "photon" usually refers to the photons *before* they pass through the telescope, while "counts" refer to the *observed* signal in the detector. That is, "counts" are the result of "photons" passing through the telescope/detector system. Thus, a count spectrum is the incident photon spectrum modified by the instrument ARF and RMF.

### **Pileup**

Pileup occurs when two or more photons are detected in a CCD pixel within one read-put period, so the detector electronics are fooled into mixing them into a single event.

### **Plasma**

A highly ionized state of matter achieved either by heating some material to very high temperatures (hundreds and thousands of degrees), or bombarding a material by a strong flux of high energy photons.

### **PSF (or PRF)**

The Point Spread Function describes the shape of the image produced on the detector by a delta function (point) source. Also known as 'Point Response Function' or PRF. A related term is the Line Spread Function (LSF), which applies to the response of a grating to a spectral line of delta-function shape.

### **Quantum Efficiency (QE)**

The QE is the fraction of incident photons registered by a detector. A strong, highly structured function of energy, originally used to describe CCD detectors such as *Chandra's* ACIS, and generalized to include other types of detectors such as multichannel plates (e.g., HRC).

### **Quantum Efficiency Uniformity (QEU)**

The QE is usually defined as a function of energy for a single point on the detector, and deviations from it at different detector locations are mapped in a QEU file.

### **RMF**

Redistribution Matrix File, maps from photon space into detector counts (pulse height or position) space. The redistribution matrix contains the information about how the incoming photons are spread out over detector channels by the detector resolution. In high resolution instruments (e.g., diffraction gratings such as HETG and LETG) the matrix is almost diagonal. In proportional counters the matrix elements are non-zero over a large area. CCD detectors, such as ACIS, are an intermediate case, with most of the response being almost diagonal, but escape peaks and low energy tails add significant contributions.

### **Sources**

Some types of common astronomical X-ray sources are described below.

- **Hot Stars:** Massive stars of spectral class earlier than type A (bluer stars like Vega) produce X-rays via shocks in stellar winds.
- **Cool Stars:** Stars like the Sun that show evidence of magnetic activity have hot atmospheres confined by magnetic fields; these hot atmospheres (aka coronae) are thought to be heated via magnetic reconnection.
- **Supernova Remnants:** The diffuse emission from the detritus of a supernova explosion, heated by shocks as the ejecta plows into the interstellar medium.

- **X-ray binaries:** Powered by accretion of matter from a companion star, which forms a disk around a central compact object and is heated (mostly) by viscous dissipation.
- **QSO/Quasar:** Highly luminous (luminosities a trillion times greater than that of the Sun) unresolved emission from the core of galaxies at high redshifts.
- **Active Galactic Nuclei:** Galaxies with high luminosity of the central, compact region, emitting into the entire spectral energy range from radio to X-rays and in some cases gamma-rays. Outflows and jets observed often.
- **Cluster of galaxies:** Group of galaxies at nearly the same distance from the Earth, exhibits X-ray emission from hot intergalactic gas.
- **Cooling Flows:** Hot intergalactic gas falling in towards the central galaxy of a cluster, increasing in density and cooling in the process.

### Spectral Class

Most stars are classified into a sequence of types organized by their photospheric (surface) temperature, denoted by the letters O, B, A, F, G, K, and M. The Sun is a G type star. Hotter stars (like O, B, etc.) are also called "earlier type stars" and cooler stars (like M, K) are also called "later type stars".

### Spectrum

A frequency distribution of observed counts as a function of detector channels. For low-resolution spectra, the detector channels are PHA or PI (mapped onto photon energy space via the RMF). For high-resolution grating spectra, the detector channels are wavelengths derived from pixel location.

### Units

- Astronomers tend to use mostly CGS units, such as [ergs] for energy, [ergs/s/cm<sup>2</sup>] for flux, etc.
- Spectra are displayed as functions of either energy (in [keV], for low-resolution spectra) or wavelength (in [Å] for high-resolution grating spectra). Spectral intensities are variously shown in units of [counts/channel], [counts/sec/keV], [photons/sec/cm<sup>2</sup>/keV], etc.
- Angular separations are measured in [degrees], [arcminutes], and [arcseconds], while angular locations (positions on the sky) are denoted either in decimal degrees or sexagesimal notation as Right Ascension ([hours:minutes:seconds] of time; 15 degrees of arc == 1 hour of time) and Declination ([degrees:minutes:seconds] of arc).

### WCS

World Coordinate System, a standardized format for storing coordinate information in image files, that allows the translation of pixel positions to true sky coordinates.

### X-Ray Telescope

Unlike in the optical, it is difficult to make mirrors or lenses to focus X-ray light. X-ray mirrors are usually built as nested paraboloid and hyperboloids that bring photons to a focus by deflecting them at small angles. Often, mirrors are entirely dispensed with and collimators are used. Also, because the Earth's atmosphere absorbs X-rays, X-ray telescopes must be placed in outer space. Some selected X-ray and gamma-ray missions are listed below.

- **Chandra**

The *Chandra* X-Ray Observatory, one of the series of Great Observatories launched by NASA (others are: Hubble, *Compton*, SIRTf). Named after Prof.

Subrahmanya Chandrasekhar, the Indian-American Nobel laureate in Physics.

See <http://chandra.harvard.edu/about/> for more details.

*Chandra* has two detectors on board, the Advanced Camera for Imaging and Spectroscopy (ACIS), a CCD detector, and the High Resolution Camera (HRC), a multichannel plate detector. There are two gratings that may be optionally used to obtain high resolution spectra: the HETGS (High Energy Transmission Grating Spectrometer) and the LETGS (Low Energy Transmission Grating Spectrometer). The HRMA (High Resolution Mirror Assembly) consists of 4 shells of nested mirrors with 2 segments - paraboloid and hyperboloid - each.

- **Compton**

The *Compton* Gamma-Ray Observatory (*CGRO*) was the first gamma ray observatory, launched in 1991 and deliberately "deorbited" in June 2000. Carried telescopes COMPTEL (0.5-30 MeV) and EGRET (30 MeV-100 GeV) that surveyed the whole sky.

- **Einstein**

The first true imaging X-ray telescope, launched in 1978, first non-solar X-ray telescope to map diffuse and point sources.

Energy Range : 2 keV - 100 MeV

- **GLAST**

Very large area, large field of view gamma-ray experiment, expected to be launched in late 2005, covering an energy range of 20 MeV to 300 GeV.

- **ROSAT**

First X-ray telescope to do all-sky survey; "pathfinder" for *Chandra*.

Lifetime : 1 June 1990 - 12 February 1999

Energy Range : X-ray 0.1 - 2.5 keV , EUV 62-206 eV

- **TRACE and SOHO**

Transition Region and Coronal Explorer and the Solar and Heliospheric Observatory, EUV telescopes devoted to solar observations, and are unique in that they use normal incidence mirrors over narrow wavelength bands to make images of the Sun.

- **XMM-Newton**

The X-Ray Multi-Mirror Mission, a large area X-ray telescope operated by ESA. Has higher count rates and energy range than *Chandra*, but less spectral resolution.

**See also:**

The *Chandra*/CIAO Dictionary: <http://asc.harvard.edu/ciao/dictionary/>

The CIAO Why Topics: <http://asc.harvard.edu/ciao/why/>

The Astronomy Cafe: <http://www.astronomycafe.net/>

Imagine the Universe: <http://imagine.gsfc.nasa.gov/>

Astro Picture of the Day: <http://antwrp.gsfc.nasa.gov/apod/astropix.html>

# Statistics Jargon for Astronomers

Under Construction!

## Akaike Information Criterion (AIC)

...

## Background Marginalization

...

## Bayes Factor (BF)

Bayes factor is defined by

$$BF = P(M_2|X_n)/P(M_1|X_n) * P(M_1)/P(M_2),$$

where  $M_i$  ( $i=1,2$ ) indicates models/distributions,  $P(M_1)/P(M_2)$  is the prior odds, and  $P(M_2|X_n)/P(M_1|X_n)$  is the posterior odds.

## Bayes' Theorem

A means to update the model parameter probability distributions based on data. Since

$$\begin{aligned} p(M|D) &= p(M|D) p(D) = p(M) p(D|M), \\ p(M|D) &= p(M) p(D|M) / p(D). \end{aligned}$$

This is Bayes' Theorem. Usually,  $M$  represents the model parameters,  $D$  represents the data, and the  $|$  symbol indicates a statement of conditional probability. Here,  $p(M)$  are the a priori probabilities on the model parameters,  $p(D|M)$  is the likelihood, and  $p(D)$  is a normalizing factor.

## Bayesian Information Criterion (BIC)

Also, called Schwarz Information Criterion (SIC).

## Bayesian v/s Frequentist

...

## Bootstrap

Bootstrapping is a well known resampling method for estimating ...

## Cash statistic

Cash statistics explains statistical inference methods based on the likelihood of a parametric model. Those methods are point estimation via the methods of maximum likelihood and the asymptotic convergence of the likelihood ratio test to Chi-square distribution, which allows to obtain a confidence interval in addition to hypothesis testing. A given parametric model does not have to be Gaussian. Any parametric model that explains underline physics can be plugged in the likelihood based methods for statistical inference on parameters. Its name is generally used among X-ray astrophysicists.

## Chi-square statistic

Widely used measure of goodness of a fit. There are many forms of this expression, starting with the

-- Model variance  $\chi^2$  :  $\chi^2 = (D - M)^2/M$

-- Data variance  $\chi^2$  :  $\chi^2 = (D - M)^2/D$

-- Iterative Primini approximation :  $\chi^2_{\{i\}} = (D - M_{\{i\}})^2/M_{\{i-1\}}$

## Confidence Interval

A confidence interval is a plausible range of values for  $\mu$ , the parameter with a quantifiable measure of its plausibility (like 95%, 99%, the level of confidence)

**Cramer-Rao Lower Bound**

Counterpart of Heisenberg's uncertainty (physics) or Kraft's inequality (information theory).

**Credible Region**

...

**Cumulative Distribution Function (CDF)**

...

**Data Augmentation**

Data augmentation is an elegant computational construct that allows one to take advantage of the fact that if it were possible to collect additional data, statistical analysis would be greatly simplified. This is true regardless of why the so-called ``missing data'' are not observed. For example, if we were able to record the counts due to background contamination in addition to total counts in each bin, it would, of course, be a trivial task to account for the background. There is a large class of powerful statistical methods designed for ``missing data'' problems. With the insight that ``true'' values of quantities recorded with measurement error can be regarded as ``missing data,'' these methods can usefully be applied to almost any astrophysical problem.

**Data Depth, Statistical**

...

**EM Algorithm**

The EM Algorithm is a computational tool that uses the method of [data augmentation](#) and can be used to optimize a [likelihood](#) function in order to compute the [Maximum Likelihood Estimate \(MLE\)](#) of an unknown parameter. Bayesians also use the EM algorithm to optimize the [a posteriori distribution](#) to compute the Maximum A Posteriori (MAP) estimate of an unknown parameter. The EM algorithm tends to be easy to implement and enjoys monotone convergence in the objective function: When optimizing a function the EM algorithm is guaranteed to go uphill.

**Estimator**

An estimator is a function of [random variable](#),  $X$ . For example, the maximum likelihood estimator (MLE) of a given parameter is the function of  $X$  where the likelihood is maximized. Plugging in data into the estimator provides an estimate (value) of the parameter. Estimators are acquired from judicious guessing, the method of maximum likelihood, the method of moments, bayesian methods, decision theoretic methods, unbiased or consistent properties in the estimator.

M.A.Hendry and J.F.L. Simmons (1995), *Distance Estimation in Cosmology*, [Vistas in Astronomy, Vol.39, pp297-314](#) explains these estimators from the astrophysicist point of views.

**Gamma Distribution**

When the parameter of an exponential distribution describes the expected rate of a physical interest such as atomic decay rate, the parameters of a gamma distribution describe the expected rate as well as the expected size of the target sample.

Therefore, Gamma distribution is a superset of Exponential distribution.

**Gibbs Sampler**

The Gibbs Sampler is a [MCMC](#) sampler that constructs a Markov chain by dividing the set of unknown parameters into a number of groups and then simulating each group in turn conditional on the current values of all the other groups.

**Hypothesis Testing**

Hypothesis Testing is a statistical decision making process with respect to an uncertain hypothesis. In general, to know the truth of the given hypothesis, some

evidence (data) is collected with an assumption that this data set was generated from the hypothesis, where summaries of the data set (statistics) should support the hypothesis. If the statistic is not consistent with the hypothesis, one can conclude that the hypothesis can be rejected based on the data. There are many test statistics, summarizing data to make a statistical decision on a given hypothesis.

### **Information, Fisher's**

...

### **Information Theory**

...

### **Informative and Non-informative Priors**

...

### **Kullback-Leibler Distance**

...

### **Kurtosis**

...

### **Likelihood**

The likelihood (or sampling distribution) quantifies the likelihood of the data given the unknown model parameters.

### **Likelihood Ratio Test (LRT)**

...

### **Markov-Chain Monte Carlo (MCMC)**

MCMC is a computational tool that is used to generate simulations from a probability distribution. Because the simulations are generated with a Markov chain, care must be taken to insure that the chain has converged to the target probability distribution and to account for the autocorrelation in this simulations. Bayesians use MCMC to explore high-dimension posterior distributions in order to estimate unknown parameters and to construct error bars for these estimates.

### **Marginalization**

...

### **Martingale**

In probability theory, a martingale satisfies the following,

$$E(X_{n+1} | X_1, \dots, X_n) = X_n,$$

where  $X_1, \dots, X_n$  are a sequence of random variables. In other words, the conditional expectation of  $X_{n+1}$ , given with all past observations, only depends on the immediate previous observation.

### **Mean, Median, Mode**

- Mean:
- Arithmetic mean:
- Geometric mean:
- Harmonic mean:
- Median: ...
- Mode:

### **Metropolis-Hastings**

The Metropolis-Hastings sampler is a MCMC sampler that uses a convenient rule to generate simulations and then uses a accept-reject step to correct the simulation to match the target probability distribution.

### **Minimum Descriptive Length (MDL)**

MDL shares commonality with BIC due to the same shape of criteria. However, their origins are different.  $p \log n$  in MDL is a measure of model complexity when

candidate models are from the exponential family (In information theory, most likely we treat binary systems). In BIC,  $p \log n$  is a by product of Laplace transformation.

### **Model Averaging**

According to Wasserman (2000), *model averaging* refers to the process of estimating some quantity under each model and then averaging the estimates according to how likely each model is; on the other hand, *model selection* refers to the problem of using the data to select one model from the list of candidate models.

### **Model Selection**

Well known statistical model selection criteria are Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and their modifications which satisfy particular conditions such as small sample size. The best model is considered to minimize the distance to the true model, which, in general, is unknown. Various distance measures exist to suite data properties. In astronomy, these model selection criteria have been used to determine the number of components in star populations and to choose the best theoretical model among candidates in cosmology.

### **Most Compact Region**

...

### **Neyman-Pearson Lemma**

...

### **Normal Distribution**

Normal distribution is known as Gaussian distribution in astronomy.

### **Odds Ratio**

...

### **Poisson Likelihood**

...

### **Posterior Distribution**

The posterior distribution represents the updated knowledge regarding the unknown model parameters after observing the data and other information pertaining to the unknown parameters. Thus, the posterior distribution combines the information in the prior distribution and the likelihood (via Bayes theorem) and is a complete summary of the knowledge regarding the unknown model parameters.

### **Posteriors**

...

### **Power, Statistical**

...

### **Principal Components Analysis (PCA)**

...

### **Prior Distribution**

The prior distribution is a probability distribution that quantifies knowledge regarding unknown quantities (e.g., model parameters) prior to observing the data or other information pertaining to the the unknown quantities.

### **Probability**

...

### **Probability Density Function (pdf)**

...

### **Random Variable (r.v.)**

A random variable is a function from  $S$ , the sample space, to  $R$ , the real line; in other words, a numerical value calculated from the outcome of a random experiment.

### **Sampling Distribution**

Sampling distribution is the probability distribution of a point estimate.

## Skewness

...

## Symbols

Some commonly used symbols

- |
  - indicates conditional probability, e.g.,  $p(\mathbf{A}|\mathbf{B})$  is read as the "the probability that  $\mathbf{A}$  is true *given that*  $\mathbf{B}$  is true."
- $\sim$ 
  - Usually written as  $\mathbf{x} \sim \mathbf{f}(\dots)$ , denotes that the variable  $\mathbf{x}$  is distributed as a function of the specified form. e.g.,
    - counts  $\sim \text{Po}(\text{lambda})$
    - flux  $\sim \text{N}(\text{mean}, \text{stddev})$
- $\mathbf{E}(\cdot)$ 
  - Expectation, or mean.
- **lambda**
  - Usually describes the Poisson intensity, unlike Astrophysical usage, where it is shorthand for wavelength.

## Unbiased Estimator

...

## Variance and Standard Deviation (SD)

Standard deviation is a measure of average fluctuation of  $X$  around  $\mu$  and also the square root of variance, which is defined as  $V(X) = E[(X - \mu)^2]$ .

### See also:

A description and usage of some of the terms listed here is available at

<http://www.ics.uci.edu/~dvd/Astro/stat-jargon-for-astro.pdf>

The Astro Jargon for Statisticians: <http://hea-www.harvard.edu/AstroStat/astrojargon.html>

The Chandra/CIAO Dictionary: <http://asc.harvard.edu/ciao/dictionary/>

The CIAO Why Topics: <http://asc.harvard.edu/ciao/why/>

The [Statistics Glossary](#) at [IPAC/Level5](#)

---

[CHASC: The California-Harvard AstroStatistics Collaboration](#)

--Last viewed on 07/19/2013 16:46:15.

