

Lecture notes assembled by

G. Jogesh Babu

Director, Center for Astrostatistics
The Pennsylvania State University

Contents

Title	Presenter	page no.
Overview of Summer School	<i>G. Jogesh Babu</i>	1
Overview of astrostatistics	<i>Eric Feigelson</i>	5
Descriptive statistics	<i>Derek Young</i>	13
Correlation and regression	<i>Mosuk Chow</i>	21
Linear regression issues in astronomy	<i>Eric Feigelson</i>	53
Exploratory data analysis and regression	<i>Derek Young</i>	59
Laws of probability, Bayes' theorem and Central Limit Theorem	<i>Mosuk Chow</i>	67
Estimation, confidence intervals and tests of hypotheses	<i>James Rosenberger</i>	89
MLEs, Cramer-Rao inequality, BIC	<i>Thomas Hettmansperger</i>	125
Mixture Models and the EM Algorithm	<i>Thomas Hettmansperger</i>	161
Likelihood Computations and Random Numbers	<i>Derek Young</i>	179
Nonparametric.Zip	<i>Thomas Hettmansperger</i>	189
Bayesian analysis	<i>Thomas Loredo</i>	211
Multivariate analysis	<i>James Rosenberger</i>	263
Multivariate Computations	<i>Derek Young</i>	317
Bootstrap	<i>G. Jogesh Babu</i>	325
Bootstrap for Goodness of fit	<i>G. Jogesh Babu</i>	337
Hypothesis testing and bootstrapping	<i>Derek Young</i>	347
Model selection, evaluation and likelihood ratio tests	<i>Bruce Lindsay</i>	355
Time Series and Stochastic Processes I	<i>John Fricks</i>	397
MCMC	<i>Murali Haran</i>	431
Spatial Statistics	<i>Murali Haran</i>	449
Time Series and Stochastic Processes II	<i>Eric Feigelson</i>	481
Cluster analysis	<i>Jia Li</i>	489

Acknowledgments: The Summer School is supported in part by NSF grant AST-0808877.

Overview of Summer School in Statistics for Astronomers IV

June 9-14, 2008

G. Jogesh Babu

This is an overview of statistical concepts and methods covered in the summer school. Eric Feigelson starts with an [overview of astrostatistics](#) giving a brief description of modern astronomy and astrophysics. He describes how the roots of many statistical concepts originated in astronomy, starting with Hipparchus in 4th c. BC. He discusses:

- Relevance of statistics in astronomy today
- State of astrostatistics today
- Methodological challenges for astrostatistics in 2000s

Derek Young starts the computer lab session with an introduction to **R** programming language and [Descriptive statistics](#). **R** is an integrated suite of software facilities for data manipulation, calculation and graphical display. Descriptive statistics describe the basic features of data in an observational study and provide simple summaries about the sample and the measures. Various commonly used techniques such as, graphical description, tabular description, and summary statistics, are illustrated through **R**.

Derek Young also presents [exploratory data analysis](#) (EDA). It is an approach to analyzing data for the purpose of formulating hypotheses worth testing, complementing the tools of conventional statistics for testing hypotheses. EDA is an approach/philosophy for data analysis that employs a variety of techniques (mostly graphical) to:

- maximize insight into a data set
- uncover underlying structure
- extract important variables
- detect outliers and anomalies
- test underlying assumptions
- develop parsimonious models, and
- provide a basis for further data collection through surveys or experiments

Mosuk Chow introduces basic principles of [probability theory](#), which is at the heart of statistical analysis. The topics include conditional probability, Bayes theorem (on which the Bayesian analysis is based), expectation, variance, standard deviation (which helps in constructing units free estimates), density of a continuous random variable (as opposed to density defined in physics), normal (Gaussian) distribution, Chi-square distribution (not to be confused with Chi-square statistic), and other important distributions. They also include some probability inequalities and the Central Limit Theorem.

Mosuk Chow also lectures on [correlation & regression](#), including correlation coefficient, the underlying principles of linear and multiple linear regression, least squares estimation, ridge regression, and principal components among others. This lecture is followed by a discussion of [linear regression issues in astronomy](#) by Eric Feigelson. He compares different regression lines used in astronomy, and illustrates them with Faber-Jackson relation.

Descriptive statistics are typically distinguished from inferential statistics. While the lab sessions on descriptive statistics provide tools to describe what the data shows, the inferential statistics helps to reach conclusions that extend beyond the immediate data alone. For instance, statistical inference is used to make judgments of an observed difference between groups is a dependable

one or one that might have happened by chance in a study. James Rosenberger's lecture on [**statistical inference**](#) focusses on methods of point estimation, confidence intervals for unknown parameters, and basic principles of testing of hypotheses.

Likelihood is the hypothetical probability that an event that has already occurred would yield a specific outcome. The concept differs from that of a probability in that a probability refers to the occurrence of future events, while a likelihood refers to past events with known outcomes.

Maximum [**likelihood estimation \(MLE\)**](#) is a popular statistical method used for fitting a mathematical model to data. Modeling real world data by estimating maximum likelihood offers a way of tuning the free parameters of the model to provide a good fit. Thomas Hettmansperger's lecture includes maximum likelihood method for linear regression, an alternative to least squares method. He also presents Cramer-Rao inequality, which sets a lower bound on the error (variance) of an estimator of parameter. It helps in finding the 'best' estimator. Hettmansperger also discusses analysis of data from two or more different populations by considering [**mixture models**](#). Here the likelihood calculations are difficult, so he introduces an iterative device called EM algorithm. Derek Yung illustrates [**likelihood computations**](#) and EM algorithm using **R**.

Thomas Hettmansperger's second lecture is on [**Nonparametric statistics**](#). Non-parametric (or distribution-free) inferential statistical methods are procedures which, unlike parametric statistics, make no assumptions about the probability distributions of the population. Here, the model structure is not specified a priori but is instead determined from data. As non-parametric methods make fewer assumptions, their applicability is much wider than the corresponding parametric methods. In this lecture he describes some simple non-parametric procedures such as sign test, Mann-Whitney two sample test and Kruskal-Wallis test for comparing several samples.

[**Bayesian inference**](#) is statistical inference in which evidence or observations are used to update or to newly infer the probability that a hypothesis may be true. The name "Bayesian" comes from the frequent use of Bayes' theorem. As evidence accumulates, the degree of belief in a hypothesis ought to change. In many statistical problems, failure to take prior knowledge into account can lead to inferior conclusions. Of course, the quality of Bayesian analysis depends on how best one can convert the prior information into mathematical prior probability. Thomas Loredo describes various methods for parameter estimation, model assessment etc, and illustrates them with examples from astronomy.

The lecture on [**Multivariate analysis**](#) by James Rosenberger introduces the statistical analysis of data containing observations of two or more variables that may depend on each other. The methods include principle components analysis, to reduce the number of variables and canonical correlation. The lecture covers many important topics including testing of hypotheses, constructing confidence regions for multivariate parameters, multivariate regression, and discriminant analysis (supervised learning). Derek Young covers computational aspects of [**Multivariate analysis**](#) in an interactive lab session.

G. J. Babu introduces a [**resampling procedure**](#) called bootstrap. It is essentially about how to get most out of repeated use of the data. Bootstrap is similar to Monte Carlo method but the 'simulation' is carried out from the data itself. It is a very general, mostly non-parametric procedure, and is widely applicable. Applications to regression, cases where the procedure fails, and where it outperforms traditional procedures are also discussed. The lecture also covers curve fitting (model fitting or [**goodness of fit**](#)) using bootstrap procedure. This procedure is important as the commonly used Kolmogorov-Smirnov procedure does not work in multidimensional case, or when the parameters of the curve is estimated. Some of these procedures are illustrated using **R** in a lab session on [**Hypothesis testing and bootstrapping**](#) by Derek Young.

The lecture on [**Model selection, evaluation, and likelihood ratio tests**](#) by Bruce Lindsay covers model selection procedures starting with Chi-square test, Rao's score test and likelihood ratio test. The discussion also includes cross validation.

The two lectures on Time Series & Stochastic Processes by [**John Fricks**](#) and [**Eric Feigelson**](#) provide an overview of Time series analysis and, more generally stochastic processes, including

time domain procedures, state space models, kernel smoothing and illustrations with examples from astronomy. The first lecture also includes a number of commonly used examples, such as Poisson processes and focuses on spectral methods for inference. A brief discussion of Kalman filter is also included.

Monte Carlo methods are a collection of techniques that use pseudo-random (computer simulated) values to estimate solutions to mathematical problems. In the tutorial on [MCMC](#), Murali Haran discusses Monte Carlo for Bayesian inference. In particular, MCMC method for the evaluation of expectations with respect to a probability distribution is illustrated. 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). MCMC procedures are successfully used in the search for extra-solar planets.

In his lecture on [Spatial Statistics](#), Murali Haran teaches spatial point processes, intensity function, homogeneous and inhomogeneous poisson processes, and estimation of Ripley's K function (statistic useful for point pattern analysis).

Jia Li covers data mining techniques, classifying data into clusters (including k-means, model clustering, single (friends of friends) and complete linkage clustering algorithms in her lecture on [Cluster analysis](#).

Overview of astrostatistics

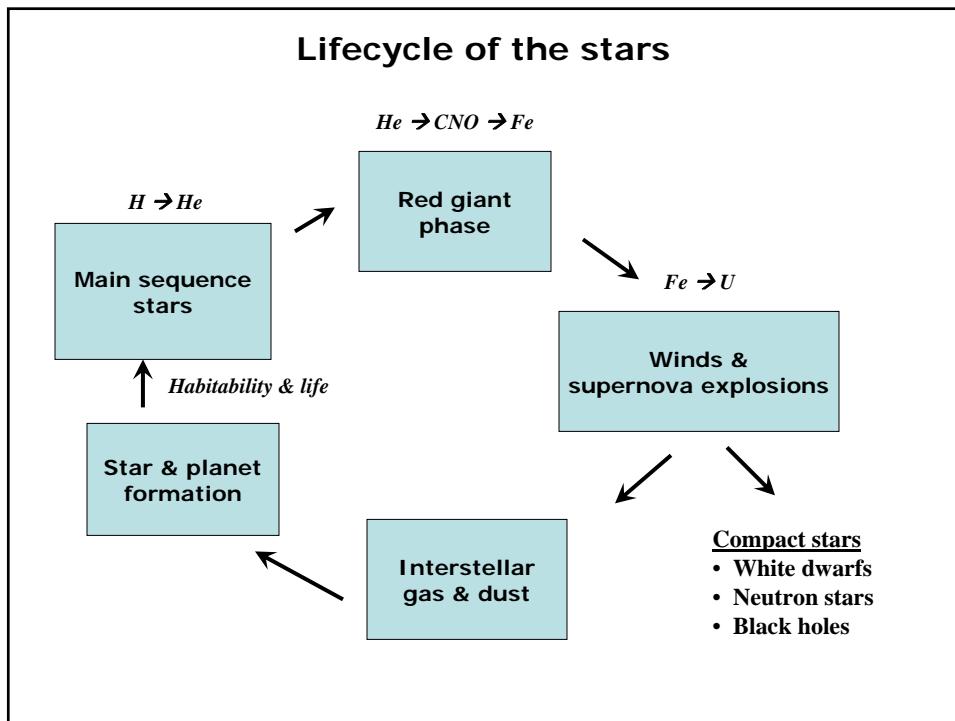
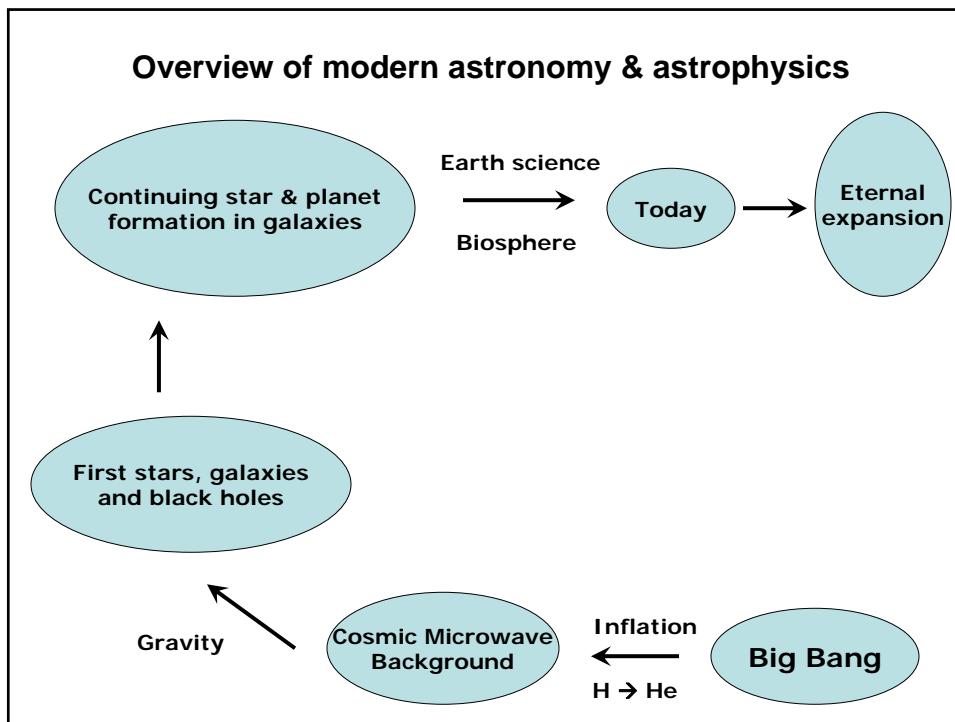
Eric Feigelson (Astro & Astrophys)
&
Jogesh Babu (Stat)

Penn State University

What is astronomy

Astronomy (astro = star, nomen = name in Greek) is the observational study of matter beyond Earth – planets in the Solar System, stars in the Milky Way Galaxy, galaxies in the Universe, and diffuse matter between these concentrations. The perspective is rooted from our viewpoint on or near Earth using telescopes or robotic probes.

Astrophysics (astro = star, physis = nature) is the study of the intrinsic nature of astronomical bodies and the processes by which they interact and evolve. This is an indirect, inferential intellectual effort based on the assumption that gravity, electromagnetism, quantum mechanics, plasma physics, chemistry, and so forth – apply universally to distant cosmic phenomena.



What is astrostatistics?

What is astronomy?

The properties of planets, stars, galaxies and the Universe, and the processes that govern them

What is statistics?

- “The first task of a statistician is cross-examination of data” (R. A. Fisher)
- “[Statistics is] the study of algorithms for data analysis” (R. Beran)
- “A statistical inference carries us from observations to conclusions about the populations sampled” (D. R. Cox)
- “Some statistical models are helpful in a given context, and some are not” (T. Speed, addressing astronomers)
- “There is no need for these hypotheses to be true, or even to be at all like the truth; rather ... they should yield calculations which agree with observations” (Osiander’s Preface to Copernicus’ *De Revolutionibus*, quoted by C. R. Rao)

“The goal of science is to unlock nature’s secrets. ... Our understanding comes through the development of theoretical models which are capable of explaining the existing observations as well as making testable predictions. ... Fortunately, a variety of sophisticated mathematical and computational approaches have been developed to help us through this interface, these go under the general heading of statistical inference.” (P. C. Gregory, *Bayesian Logical Data Analysis for the Physical Sciences*, 2005)

My conclusion:

The application of statistics to high-energy astronomical data is not a straightforward, mechanical enterprise. It requires careful statement of the problem, model formulation, choice of statistical method(s), and judicious evaluation of the result.

Astronomy & statistics: A glorious history

Hipparchus (4th c. BC): Average via midrange of observations

Galileo (1572): Average via mean of observations

Halley (1693): Foundations of actuarial science

Legendre (1805): Cometary orbits via least squares regression

Gauss (1809): Normal distribution of errors in planetary orbits

Quetelet (1835): Statistics applied to human affairs

*But the fields diverged in the late 19-20th centuries,
astronomy → astrophysics (EM, QM)
statistics → social sciences & industries*

Do we need statistics in astronomy today?

- Are these stars/galaxies/sources an unbiased sample of the vast underlying population?
- When should these objects be divided into 2/3/... classes?
- What is the intrinsic relationship between two properties of a class (especially with confounding variables)?
- Can we answer such questions in the presence of observations with measurement errors & flux limits?

Do we need statistics in astronomy today?

- Are these stars/galaxies/sources an unbiased sample of the vast underlying population? **Sampling**
- When should these objects be divided into 2/3/... classes? **Multivariate classification**
- What is the intrinsic relationship between two properties of a class (especially with confounding variables)? **Multivariate regression**
- Can we answer such questions in the presence of observations with measurement errors & flux limits?
Censoring, truncation & measurement errors

- When is a blip in a spectrum, image or datastream a real signal?
Statistical inference
- How do we model the vast range of variable objects (extrasolar planets, BH accretion, GRBs, ...)?
Time series analysis
- How do we model the 2-6-dimensional points representing galaxies in the Universe or photons in a detector?
Spatial point processes & image processing
- How do we model continuous structures (CMB fluctuations, interstellar/intergalactic media)?
Density estimation, regression

How often do astronomers need statistics? (*a bibliometric measure*)

Of ~15,000 refereed papers annually:

- 1% have 'statistics' in title or keywords
- 5% have 'statistics' in abstract
- 10% treat variable objects
- 5-10% (est) analyze data tables
- 5-10% (est) fit parametric models

The state of astrostatistics today

The typical astronomical study uses:

- Fourier transform for temporal analysis (Fourier 1807)
- Least squares regression (Legendre 1805, Pearson 1901)
- Kolmogorov-Smirnov goodness-of-fit test (Kolmogorov, 1933)
- Principal components analysis for tables (Hotelling 1936)

Even traditional methods are often misused:

- Six unweighted bivariate least squares fits are used interchangeably in H_o studies with wrong confidence intervals
Feigelson & Babu ApJ 1992
- Likelihood ratio test (F test) usage typically inconsistent with asymptotic statistical theory
Protassov et al. ApJ 2002
- K-S g.o.f. probabilities are inapplicable when the model is derived from the data
Babu & Feigelson ADASS 2006

A new imperative: Virtual Observatory

Huge, uniform, multivariate databases are emerging from specialized survey projects & telescopes:

- 10⁹-object catalogs from USNO, 2MASS & SDSS opt/IR surveys
- 10⁶- galaxy redshift catalogs from 2dF & SDSS
- 10⁵-source radio/infrared/X-ray catalogs
- 10³⁻⁴-samples of well-characterized stars & galaxies with dozens of measured properties
- Many on-line collections of 10²-10⁶ images & spectra
- Planned Large-aperture Synoptic Survey Telescope will generate ~10 Pby

The Virtual Observatory is an international effort underway to federate these distributed on-line astronomical databases.

Powerful statistical tools are needed to derive scientific insights from extracted VO datasets
 (NSF FRG involving PSU/CMU/Caltech)

But astrostatistics is an emerging discipline

- We organize cross-disciplinary conferences at Penn State *Statistical Challenges in Modern Astronomy* (1991/1996, 2001/06)
- Fionn Murtagh & Jean-Luc Starck run methodological meetings & write monographs
- We organize Summer Schools at Penn State and astrostatistics workshops at SAMS
- Powerful astro-stat collaborations appearing in the 1990s:
 - Penn State CAST (Jogesh Babu, Eric Feigelson)
 - Harvard/Smithsonian (David van Dyk, Chandra scientists, students)
 - CMU/Pitt = PICA (Larry Wasserman, Chris Genovese, ...)
 - NASA-ARC/Stanford (Jeffrey Scargle, David Donoho)
 - Efron/Petrosian, Berger/Jeffreys/Loredo/Connors, Stark/GONG, ...

Some methodological challenges for astrostatistics in the 2000s

- Simultaneous treatment of measurement errors and censoring (esp. multivariate)
- Statistical inference and visualization with very-large-N datasets too large for computer memories
- A user-friendly cookbook for construction of likelihoods & Bayesian computation of astronomical problems
- Links between astrophysical theory and wavelet coefficients (spatial & temporal)
- Rich families of time series models to treat accretion and explosive phenomena

Structural challenges for astrostatistics

Cross-training of astronomers & statisticians

New curriculum, summer workshops
Effective statistical consulting

Enthusiasm for astro-stat collaborative research

Recognition within communities & agencies
More funding (astrostat gets <0.1% of astro+stat)

Implementation software

StatCodes Web metasite (www.astro.psu.edu/statcodes)
Standardized in R, MatLab or VOStat? (www.r-project.org)

Inreach & outreach

A Center for Astrostatistics to help attain these goals

Descriptive Statistics

In the course of learning a bit about how to generate data summaries in R, one will inevitably learn some useful R syntax and commands. Thus, this first tutorial on descriptive statistics serves a dual role as a brief introduction to R. When this tutorial is used online, the bolded, indented lines

```
# like this one
```

are meant to be copied and pasted directly into R at the command prompt.

Obtaining astronomical datasets

The astronomical community has a vast complex of on-line databases. Many databases are hosted by data centers such as the [Centre des Données 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 [Vizier Catalogue Service](#) as follows:

- In Web browser, go to 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
- Save ASCII file on disk for ingestion into R

Reading data into R

Enter R by typing "R" (UNIX) or double-clicking to execute Rgui.exe (Windows) or R.app (Mac). In the commands below, we start by extracting some [system and user information](#), the [R.version](#) you are using, and some of its [capabilities](#). [citation](#) tells how to cite R in publications. R is released under the [GNU Public Licence](#), as indicated by [copyright](#). Typing a question mark in front of a command opens the help file for that command.

```
Sys.info()
R.version
capabilities()
citation()
?copyright
```

The various capitalizations above are important as R is case-sensitive. When using R interactively, it is very helpful to know that the up-arrow key can retrieve previous commands, which may be edited using the left- and right-arrow keys and the delete key.

The last command above, `?copyright`, is equivalent to `help(copyright)` or `help("copyright")`. However, to use this command you have to know that the function called "copyright" exists. Suppose that you knew only that there was a function in R that returned copyright information but you could not remember what it was called. In this case, the [help.search](#) function provides a handy reference tool:

```
help.search("copyright")
```

Last but certainly not least, a vast array of documentation and reference materials may be accessed via a simple command:

```
help.start()
```

The initial working directory in R is set by default or by the directory from which R is invoked (if it is invoked on the command line). It is possible to read and set this working directory using the [getwd](#) or [setwd](#) commands. A list of the files in the current working directory is given by [list.files](#), which has a variety of useful options and is only one of several utilities interfacing to the computer's [files](#). In the [setwd](#) command, note that in Windows, path (directory) names are not case-sensitive and may contain either forward slashes or backward slashes; in the latter case, a backward slash must be written as "\\" when enclosed in quotation marks.

```
getwd()
list.files() # what's in this directory?
# The # symbol means that the rest of that line is a comment.
```

We wish to read an ASCII data file into an R object using the [read.table](#) command or one of its variants. Let's begin with a cleaned-up version of the Hipparcos dataset described above, a description of which is given at http://astrostatistics.psu.edu/datasets/HIP_star.html.

```
hip <- read.table("http://astrostatistics.psu.edu/datasets/HIP_star.dat",
header=T,fill=T) # T is short for TRUE
```

The "<-", which is actually "less than" followed by "minus", is the R assignment operator. Admittedly, this is a bit hard to type repeatedly, so fortunately R also allows the use of a single equals sign (=) for assignment.

Note that no special character must be typed when a command is broken across lines as in the example above. Whenever a line is entered that is not yet syntactically complete, R will replace the usual prompt, ">" with a + sign to indicate that more input is expected. The [read.table](#) function can refer to a location on the web, though a filename (of a file in the working directory) or a pathname would have sufficed. The "header=TRUE" option is used because the first row of the file is a header containing the names of the columns. We used the "fill=TRUE" option because some of the columns have only 8 of the 9 columns filled, and "fill=TRUE" instructs R to fill in blank fields at the end of the line with missing values, denoted by the special R constant [NA](#) ("not available"). **Warning:** This only works in this example because all of the empty cells are in the last column of the table. (You can verify this by checking the ASCII file [HIP_star.dat](#).) Because the [read.table](#) function uses delimiters to determine where to break between columns, any row containing only 8 values would always put the [NA](#) in the 9th column, regardless of where it was intended to be placed. As a general rule, data files with explicit delimiters are to be preferred to files that use line position to indicate column number, particularly when missing data are present. If you must use line position, R provides the [read.fortran](#) and [read.fwf](#) functions for reading fixed width format files.

Summarizing the dataset

The following R commands list the [dimensions](#) of the dataset and print the variable [names](#) (from the single-line header). Then we list the first row, the first 20 rows for the 7th column, the second row for all but the 7th column, and the [sum](#) of the 3rd column.

```
dim(hip)
names(hip)
hip[1,]
hip[1:20,7]
hip[2,-7]
sum(hip[,3])
```

Note that vectors, matrices, and arrays are indexed using the square brackets and that "1:20" is shorthand for the vector containing integers 1 through 20, inclusive. Even punctuation marks such as the colon have help entries, which may be accessed using [help\(":"\)](#).

Next, list the [maximum](#), [minimum](#), [median](#), and [mean absolute deviation](#) (similar to standard deviation) of each column. First we do this using a [for](#)-loop, which is a slow process in R. Inside the loop, [c](#) is a generic R function that combines its arguments into a vector and [print](#) is a generic R command that prints the contents of an object. After the inefficient but intuitively clear approach using a [for](#)-loop, we then do the same job in a more efficient fashion using the [apply](#) command. Here the "2" refers to columns in the `x` array; a "1" would refer to rows.

```
for(i in 1:ncol(hip)) {
  print(c(max(hip[,i]), min(hip[,i]), median(hip[,i]), mad(hip[,i])))
}
apply(hip, 2, max)
apply(hip, 2, min)
apply(hip, 2, median)
apply(hip, 2, mad)
```

The curly braces {} in the for loop above are optional because there is only a single command inside. Notice that the output gives only NA for the last column's statistics. This is because a few values in this column are missing. We can tell how many are missing and which rows they come from as follows:

```
sum(is.na(hip[,9]))
which(is.na(hip[,9]))
```

There are a couple of ways to deal with the NA problem. One is to repeat all of the above calculations on a new R object consisting of only those rows containing no NAs:

```
y <- na.omit(hip)
for(i in 1:ncol(y)) {
  print(c(max(y[,i]), min(y[,i]), median(y[,i]), mad(y[,i])))
}
```

Another possibility is to use the `na.rm` (remove NA) option of the summary functions. This solution gives slightly different answers from the solution above; can you see why?

```
for(i in 1:ncol(hip)) {
  print(c(max(hip[,i],na.rm=T), min(hip[,i],na.rm=T), median(hip[,i],na.rm=T),
mad(hip[,i],na.rm=T)))}
```

A vector can be [sorted](#) using the Shellsort or Quicksort algorithms; [rank](#) returns the order of values in a numeric vector; and [order](#) returns a vector of indices that will sort a vector. The last of these functions, [order](#), is often the most useful of the three, because it allows one to reorder all of the rows of a matrix according to one of the columns:

```
sort(hip[1:10,3])
hip[order(hip[1:10,3]),]
```

Each of the above lines gives the sorted values of the first ten entries of the third column, but the second line reorders *each* of the ten rows in this order. Note that neither of these commands actually alters the value of `x`, but we could reassign `x` to equal its sorted values if desired.

Standard errors and confidence intervals

The standard error of an estimator is, by definition, an estimate of the standard deviation of that estimator. Let's consider an example.

Perhaps the most commonly used estimator is the sample mean (called a *statistic* because it depends only on the data), which is an estimator of the population mean (called a *parameter*). Assuming that our sample of data truly consists of independent observations of a random variable X , the true standard deviation of the sample mean equals $\text{stdev}(X)/\sqrt{n}$, where n is the sample size. However, we do not usually know $\text{stdev}(X)$, so we estimate the standard deviation of the sample mean by replacing $\text{stdev}(X)$ by an estimate thereof.

If the $V\text{mag}$ column (the 2nd column) of our dataset may be considered a random sample from some larger population, then we may estimate the true mean of this population by

```
mean(hip[,2])
```

and the standard error of this estimator is

```
sd(hip[,2]) / sqrt(2719)
```

We know that our estimator of the true population mean is not exactly correct, so a common way to incorporate the uncertainty in our measurements into reporting estimates is by reporting a confidence interval. A confidence interval for some population quantity is always a set of "reasonable" values for that quantity. In this case, the Central Limit Theorem tells us that the sample mean has a roughly Gaussian, or normal, distribution centered at the true population mean. Thus, we may use the fact that 95% of the mass of any Gaussian distribution is contained within 1.96 standard deviations of its mean to construct the following 95% confidence interval for the true population mean of $V\text{mag}$:

```
mean(hip[,2]) + c(-1.96,1.96)*sd(hip[,2]) / sqrt(2719)
```

In fact, many confidence intervals in statistics have exactly the form above, namely, (estimator) \pm (critical value) * (standard error of estimator).

The precise interpretation of a confidence interval is a bit tricky. For instance, notice that the above interval is centered not at the true mean (which is unknown), but at the sample mean. If we were to take a different random sample of the same size, *the confidence interval would change even though the true mean remains fixed*. Thus, the correct way to interpret the "95%" in "95% confidence interval" is to say that roughly 95% of all such hypothetical intervals will contain the true mean. In particular, it is **not** correct to claim, based on the previous output, that there is a 95% probability that the true mean lies between 8.189 and 8.330. Although this latter interpretation is incorrect, if one chooses to use Bayesian estimation procedures, then the analogue of a confidence interval is a so-called "credible interval"; and the incorrect interpretation of a confidence interval is actually the correct interpretation of a credible interval (!).

As a brief illustration, suppose that we draw a random sample of size 100 from the $V\text{mag}$ column (which we will treat as a random sample from the true population of $V\text{mag}$ measurements) and construct a 95% confidence interval for the true population mean:

```
temp.hip=sample(hip[,2],100)
mean(temp.hip) + c(-1.96,1.96)*sd(temp.hip) / sqrt(100)
```

Run the above again and compare the two confidence intervals that you obtain.

More R syntax

[Arithmetic](#) in R is straightforward. Some common operators are: + for addition, - for subtraction, * for multiplication, / for division, %/% for integer division, %% for modular arithmetic, ^ for exponentiation. The help page for these operators may accessed by typing, say,

```
? '+'
```

Some common built-in functions are [exp](#) for the exponential function, [sqrt](#) for square root, [log10](#) for base-10 logarithms, and [cos](#) for cosine. The syntax resembles "sqrt(z)". [Comparisons](#) are made using < (less than), <= (less than or equal), == (equal to) with the syntax "a >= b". To test whether a and b are exactly equal and return a TRUE/FALSE value (for instance, in an "if" statement), use the command [identical\(a,b\)](#) rather a==b. Compare the following two ways of comparing the vectors a and b:

```
a <- c(1,2);b <- c(1,3)
a==b
identical(a,b)
```

Also note that in the above example, 'all(a==b)' is equivalent to 'identical(a,b)'.

R also has other [logical](#) operators such as & (AND), | (OR), ! (NOT). There is also an xor (exclusive or) function. Each of these four functions performs elementwise comparisons in much the same way as arithmetic operators:

```
a <- c(TRUE,TRUE,FALSE,FALSE);b <- c(TRUE,FALSE,TRUE,FALSE)
!a
a & b
a | b
xor(a,b)
```

However, when 'and' and 'or' are used in programming, say in 'if' statements, generally the '&&' and '||' forms are preferable. These longer forms of 'and' and 'or' evaluate left to right, examining only the first element of each vector, and evaluation terminates when a result is determined. Some other operators are listed [here](#).

The expression "y == x^2" evaluates as TRUE or FALSE, depending upon whether y equals x squared, and performs no assignment (if either y or x does not currently exist as an R object, an error results).

Let us continue with simple characterization of the dataset: find the row number of the object with the smallest value of the 4th column using [which.min](#). A longer, but instructive, way to accomplish this task creates a long vector of logical constants (tmp), mostly FALSE with one TRUE, then pick out the row with "TRUE".

```
which.min(hip[,4])
tmp <- (hip[,4]==min(hip[,4]))
(1:nrow(hip))[tmp]      #   or equivalently,
which(tmp)
```

The [cut](#) function divides the range of x into intervals and codes the values of x according to which interval they fall. It this is a quick way to group a vector into bins. Use the "breaks" argument to either specify a vector of bin boundaries, or give the number of intervals into which x should be cut. Bin string labels can be specified. Cut converts numeric vectors into an R object of class "[factor](#)" which can be ordered and otherwise manipulated; e.g. with command [levels](#). A more flexible method for dividing a vector into groups using user-specified rules is given by [split](#).

```
table(cut(hip[, "Plx"],breaks=20:25))
```

The command above uses several tricks. Note that a column in a matrix may be referred to by its name (e.g., "Plx") instead of its number. The notation '20:25' is short for 'c(20,21,22,23,24,25)' and in general, 'a:b' is the vector of consecutive integers starting with a and ending with b (this also works if a is larger than b). Finally, the [table](#) command tabulates the values in a vector or factor.

Although R makes it easy for experienced users to invoke multiple functions in a single line, it may help to recognize that the previous command accomplishes the same task as following a string of commands:

```
p <- hip[, "Plx"]
cuts <- cut(p, breaks=20:25)
table(cuts)
```

The only difference is that the string of three separate commands creates two additional R objects, p and cuts. The preferred method of carrying out these operations depends on whether there will later be any use for these additional objects.

Finally, suppose that you performed a calculation resulting from many lines of code, but you forgot to assign it a name. You can use the [Last.value](#) command to recall that value.

```
p+rnorm(length(p))
p.r <- .Last.value
p.r
```

Univariate plots

Recall the variable names in the Hipparcos dataset using the [names](#) function. By using [attach](#), we can automatically create temporary variables with these names (these variables are not saved as part of the R session, and they are superseded by any other R objects of the same names).

```
names(hip)
attach(hip)
```

After using the attach command, we can obtain, say, individual [summaries](#) of the variables:

```
summary(Vmag)
summary(B.V)
```

Next, summarize some of this information graphically using a simple yet sometimes effective visualization tool called a dotplot or dotchart, which lets us view all observations of a quantitative variable simultaneously:

```
dotchart(B.V)
```

The shape of the distribution of the B.V variable may be viewed using a traditional histogram. If we use the prob=TRUE option for the histogram so that the vertical axis is on the probability scale (i.e., the histogram has total area 1), then a so-called *kernel density estimate*, or histogram smoother, can be overlaid:

```
hist(B.V, prob=T)
d <- density(B.V, na.rm=T)
lines(d, col=2, lwd=2, lty=2)
```

The topic of density estimation will be covered in a later tutorial. For now, it is important to remember that even though histograms and density estimates are drawn in two-dimensional space, they are intrinsically *univariate* analysis techniques here. We are only studying the single variable B.V in this example (though there are multivariate versions of these techniques as well).

Check the help file for the [par](#) function (by typing "?par") to see what the col, lwd, and lty options accomplish in the [lines](#) function above.

A simplistic histogram-like object for small datasets, which both gives the shape of a distribution and displays each observation, is called a stem-and-leaf plot. It is easy to create by hand, but R will create a text

version:

```
stem(sample(B.V,100))
```

The sample command was used above to obtain a random sample of 100, without replacement, from the B.V vector.

Finally, we consider [box-and-whisker plots](#) (or "boxplots") for the four variables Vmag, pmRA, pmDE, and B.V (the last variable used to be B-V, or B minus V, but R does not allow certain characters). These are the 2nd, 6th, 7th, and 9th columns of 'hip'.

```
boxplot(hip[,c(2,6,7,9)])
```

Our first attempt above looks pretty bad due to the different scales of the variables, so we construct an array of four single-variable plots:

```
par(mfrow=c(2,2))
for(i in c(2,6,7,9))
  boxplot(hip[,i],main=names(hip)[i])
par(mfrow=c(1,1))
```

The [boxplot](#) command does more than produce plots; it also returns output that can be more closely examined. Below, we produce boxplots *and* save the output.

```
b <- boxplot(hip[,c(2,6,7,9)])
names(b)
```

'b' is an object called a list. To understand its contents, read the help for [boxplot](#). Finally, suppose we wish to see all of the outliers in the pmRA variable, which is the second of the four variables in the current boxplot:

```
b$names[2]
b$out[b$group==2]
```

R scripts

While R is often run interactively, one often wants to carefully construct R scripts and run them later. A file containing R code can be run using the [source](#) command.

In addition, R may be run in batch mode. The editor [Emacs](#), together with "[Emacs speaks statistics](#)", provides a nice way to produce R scripts.

Summer School in Statistics for
Astronomers & Physicists, IV
June 9-14, 2008

Correlation and Regression

June 9, 1:30-3:00pm

Random Variables

- Let X and Y be random variables (RV).
- A RV is specified by a distribution
 - Continuous: *probability density function (pdf)*, $f_X(x), f_Y(y)$.
 - Discrete: *probability mass function (pmf)*, $p_X(x_i), p_Y(y_i)$.

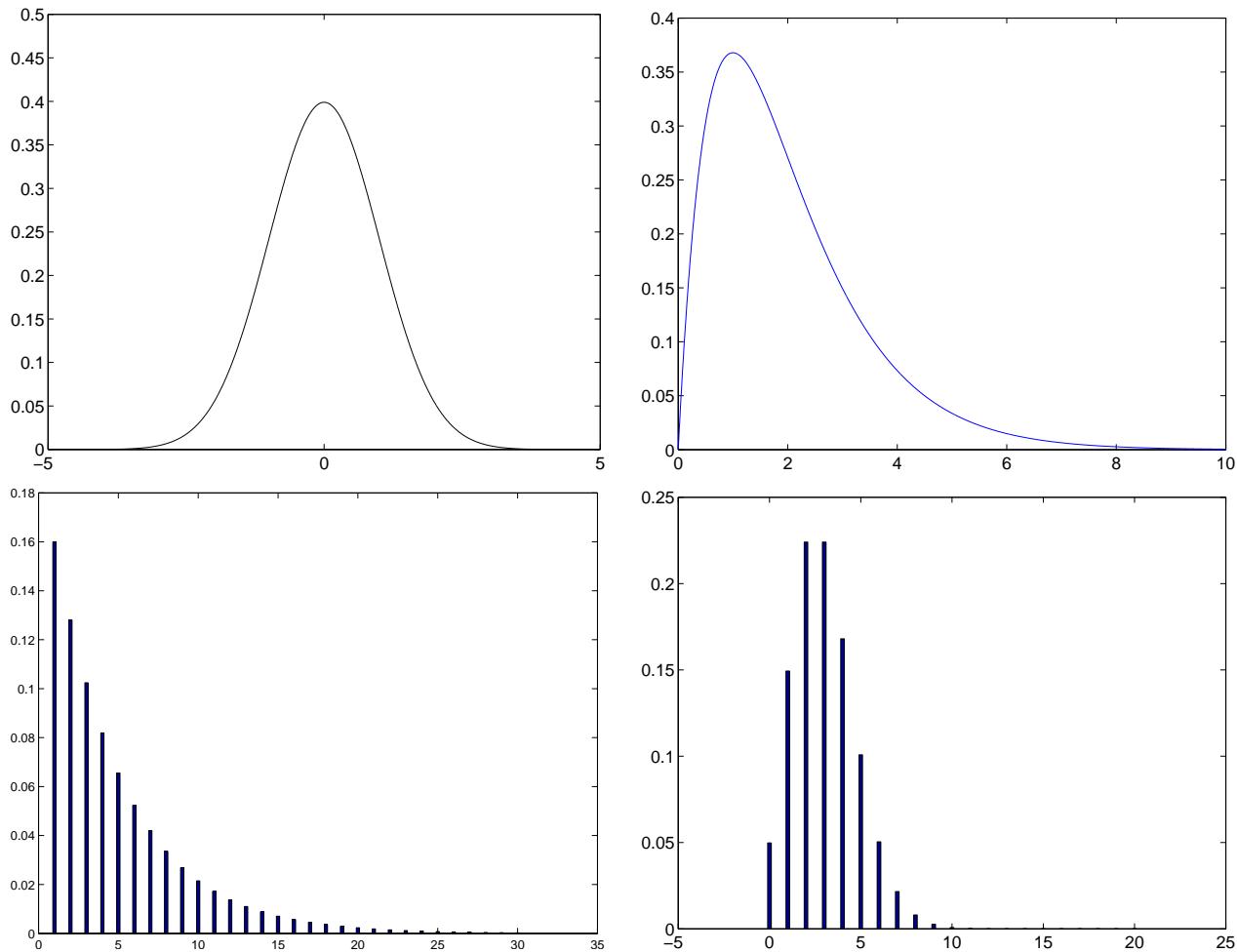


Figure 1: Distributions of random variables

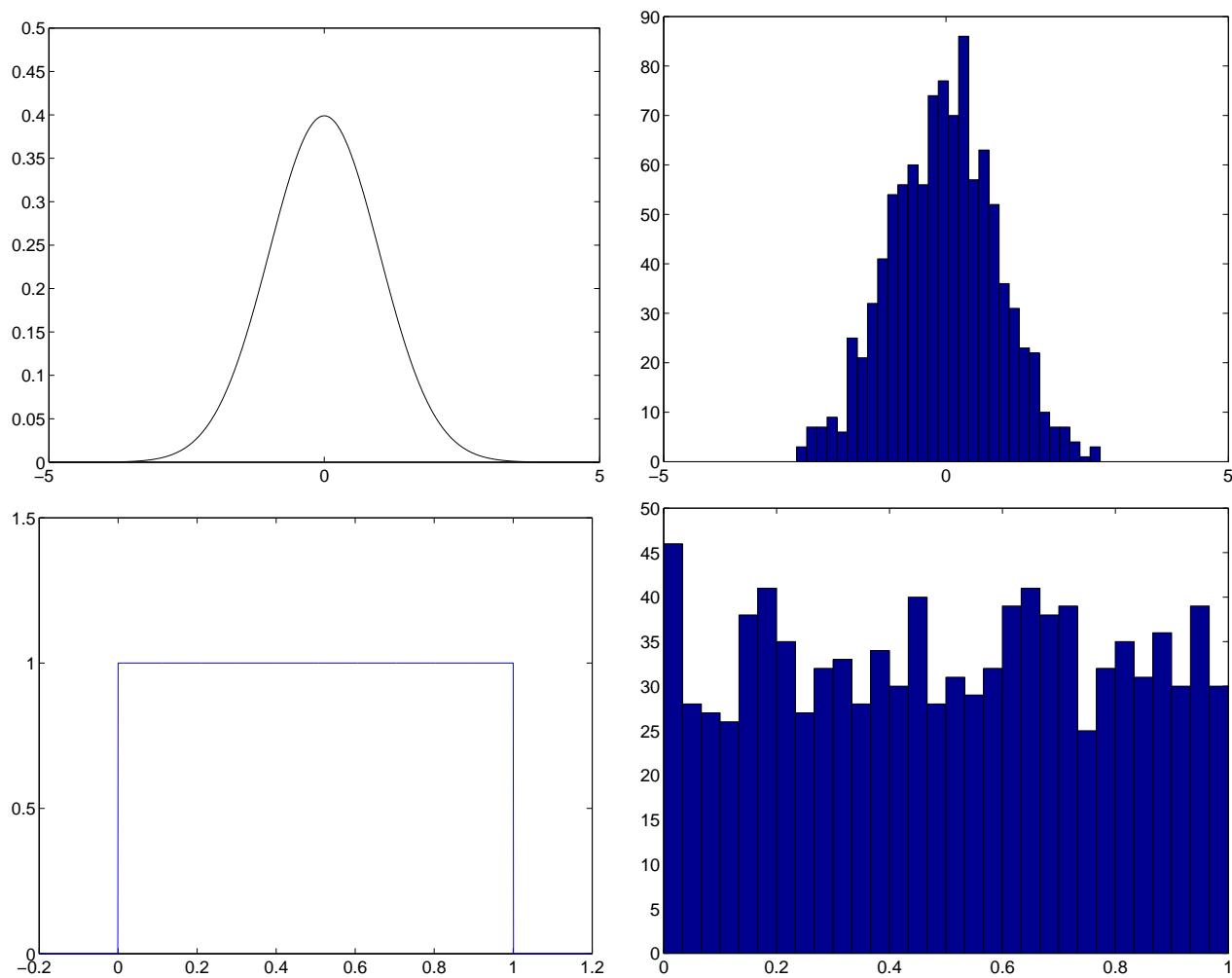


Figure 2: Histograms vs. pdf

- Expectation: $E(X) = \int_{-\infty}^{\infty} xf_X(x)dx.$
 - Linear property: $E(aX + bY) = aE(X) + bE(Y).$
- Variance: $Var(X) = E[(X - E(X))^2].$
- Joint Distribution
 - The joint pdf of X and Y : $f(x, y).$

$$f_X(x) = \int_{-\infty}^{\infty} f(x, y)dy$$

$$f_Y(y) = \int_{-\infty}^{\infty} f(x, y)dx$$

- Covariance:

$$Cov(X, Y) = E[(X - E(X))(Y - E(Y))]$$

- Correlation Coefficient:

$$\rho(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}}$$

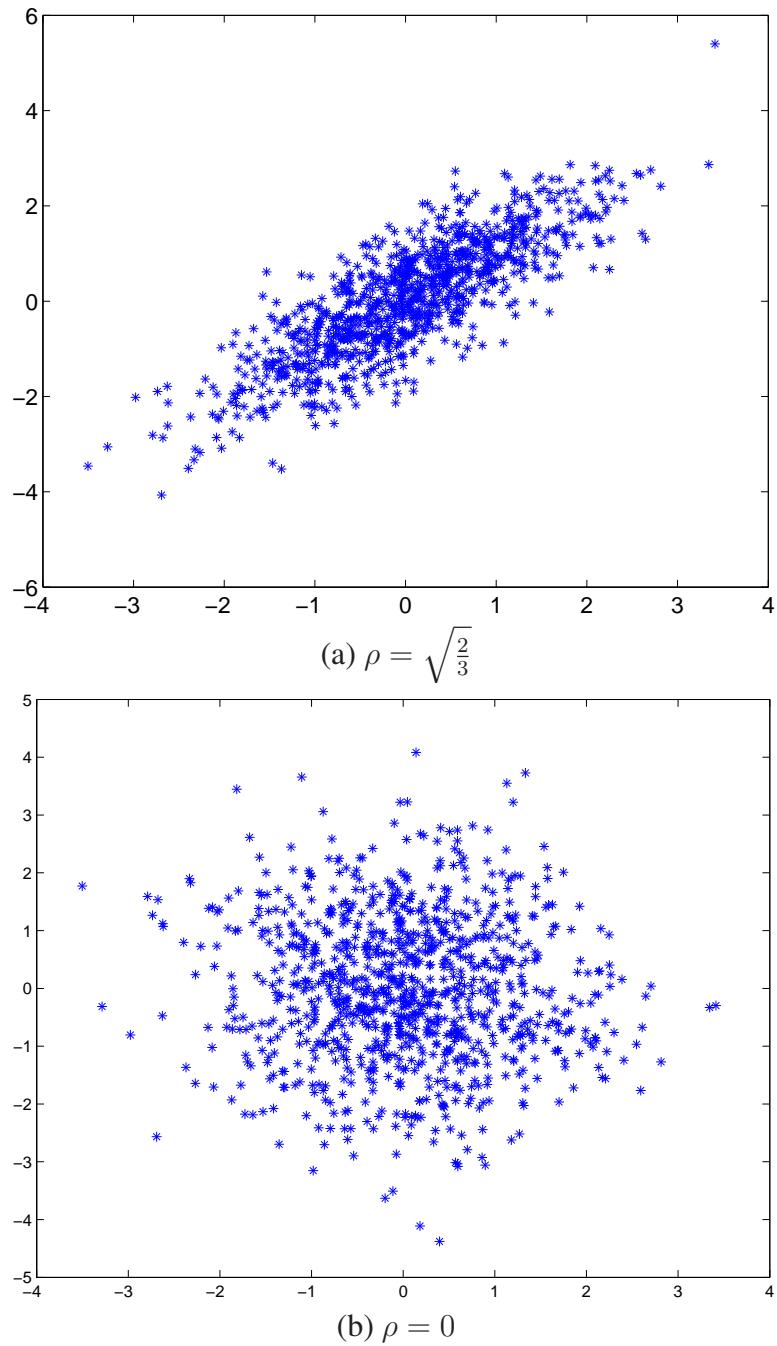


Figure 3: Different joint distributions with identical marginal distributions.

- The Hipparcos data:

1. HIP: Hipparcos star number
2. Vmag: Visual band magnitude. This is an inverted logarithmic measure of brightness
3. RA: Right Ascension (degrees), positional coordinate in the sky equivalent to longitude on the Earth
4. DE: Declination (degrees), positional coordinate in the sky equivalent to latitude on the Earth
5. Plx: Parallactic angle (mas = milliarcseconds). $1000/\text{Plx}$ gives the distance in parsecs (pc)
6. pmRA: Proper motion in RA (mas/yr). RA component of the motion of the star across the sky
7. pmDE: Proper motion in DE (mas/yr). DE component of the motion of the star across the sky
8. e-Plx: Measurement error in Plx (mas)
9. B-V: Color of star (mag)

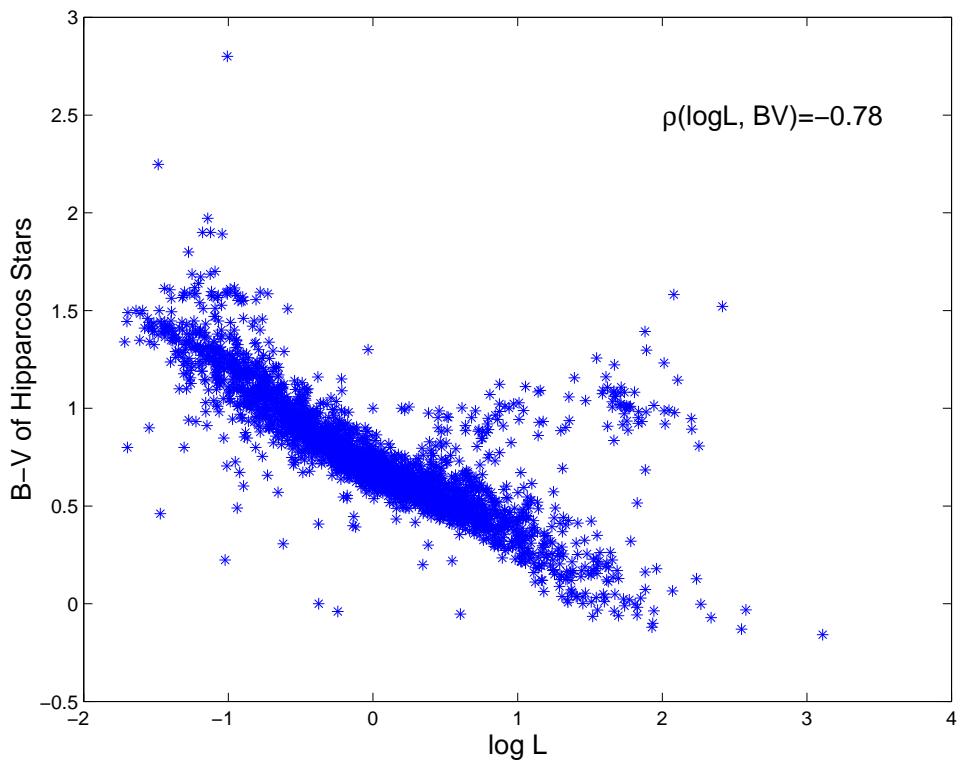


Figure 4: Scatter plot of the Hipparcos data. $\log L = (15 - \text{Vmag} - 5 \log \text{Plx})/2.5$.

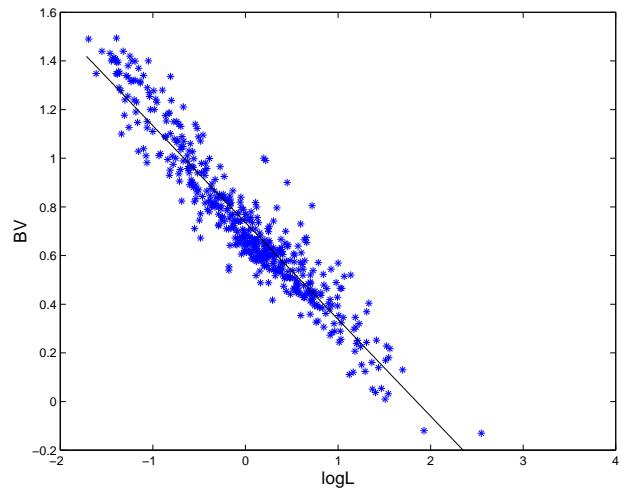
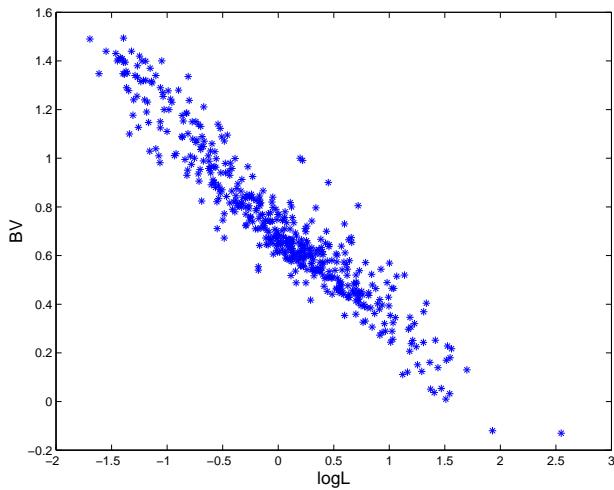
Linear Regression

- Let X be the *predictor variable* and Y the *response variable*.
- Suppose $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$
- Regression function: $E(Y|X) = \beta_0 + \beta_1 X$
- Least square estimation:

$$\arg \min_{\hat{\beta}_0, \hat{\beta}_1} \sum_i (Y_i - (\hat{\beta}_0 + \hat{\beta}_1 X_i))^2$$

- Let

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i , \quad e_i = Y_i - \hat{Y}_i$$



Linear Regression

- Let

$$\begin{aligned}
 \bar{X} &= \frac{\sum_{i=1}^n X_i}{n} \\
 \bar{Y} &= \frac{\sum_{i=1}^n Y_i}{n} \\
 \hat{\sigma}_X^2 &= \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n} \\
 \hat{\sigma}_Y^2 &= \frac{\sum_{i=1}^n (Y_i - \bar{Y})^2}{n} \\
 \hat{\sigma}_{X,Y} &= \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n} \\
 \hat{\rho}_{X,Y} &= \frac{\hat{\sigma}_{X,Y}}{\hat{\sigma}_X \hat{\sigma}_Y}
 \end{aligned}$$

- Regression $\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i$

$$\begin{aligned}
 \hat{\beta}_1 &= \hat{\rho}_{X,Y} \cdot \frac{\sigma_Y}{\sigma_X} \\
 \hat{\beta}_0 &= \bar{Y} - \hat{\beta}_1 \bar{X}
 \end{aligned}$$

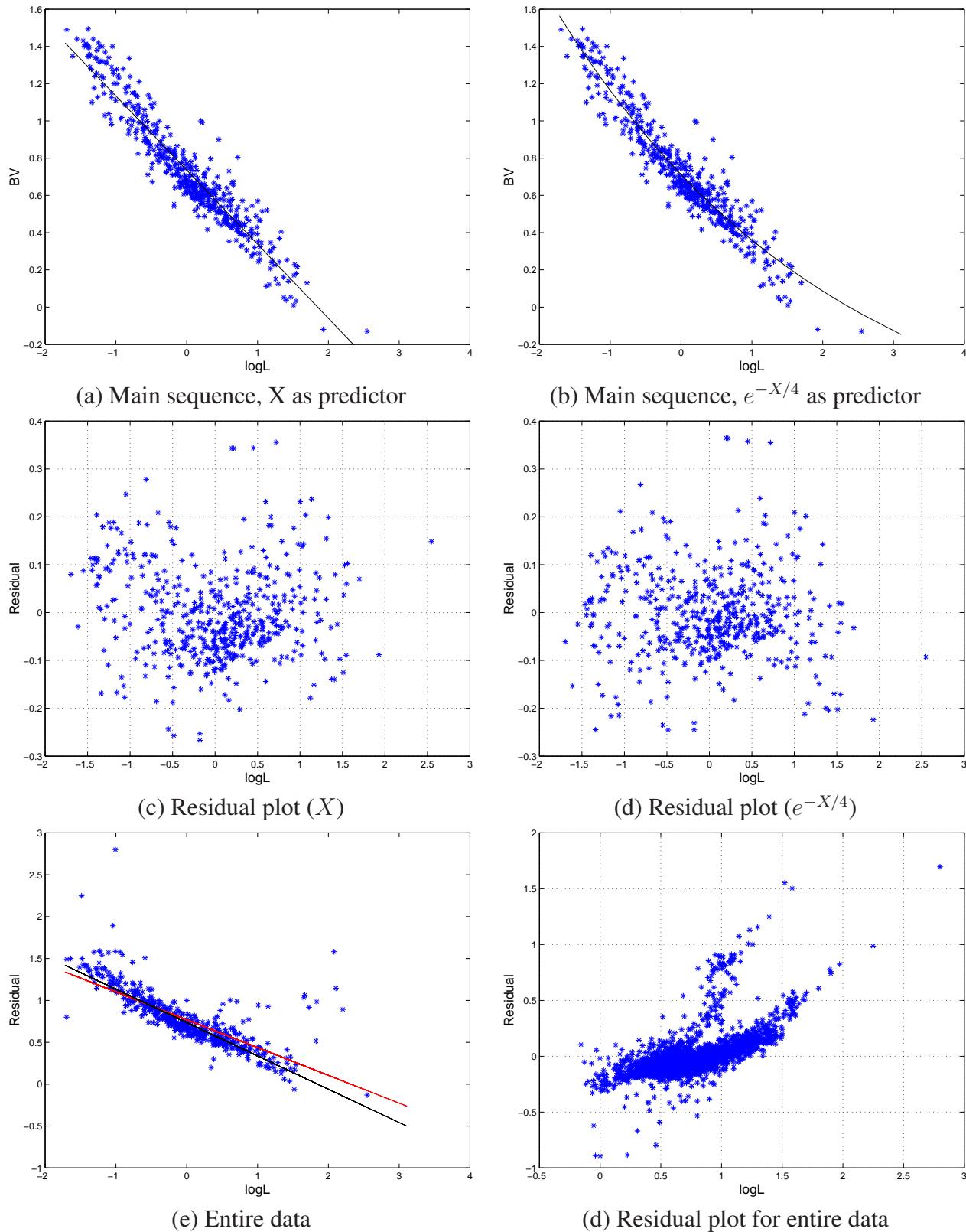


Figure 5: Linear regression on the Hipparcos data

Multiple Linear Regression

- Input vector: $X = (X_1, X_2, \dots, X_p)$.
- Output Y is real-valued.
- Predict Y from X by $f(X)$ so that the expected loss function

$$E(L(Y, f(X)))$$

is minimized.

- Square loss:

$$L(Y, f(X)) = (Y - f(X))^2 .$$

- The optimal predictor

$$\begin{aligned} f^*(X) &= \operatorname{argmin}_{f(X)} E(Y - f(X))^2 \\ &= E(Y \mid X) . \end{aligned}$$

- The function $E(Y \mid X)$ is the *regression function*.

Example

Problem:

The number of active physicians in a Standard Metropolitan Statistical Area (SMSA), denoted by Y , is expected to be related to total population (X_1 , measured in thousands), land area (X_2 , measured in square miles), and total personal income (X_3 , measured in millions of dollars). Data are collected for 141 SMSAs, as shown in the following table.

$i :$	1	2	3	...	139	140	141
X_1	9387	7031	7017	...	233	232	231
X_2	1348	4069	3719	...	1011	813	654
X_3	72100	52737	54542	...	1337	1589	1148
Y	25627	15389	13326	...	264	371	140

Goal: Predict Y from X_1 , X_2 , and X_3 .

Linear Methods

- The linear regression model

$$f(X) = \beta_0 + \sum_{j=1}^p X_j \beta_j .$$

- What if the model is not true?
 - It is a good approximation
 - Because of the lack of training data/or smarter algorithms, it is the most we can extract robustly from the data.
- Comments on X_j :
 - Quantitative inputs
 - Transformations of quantitative inputs, e.g., $\log(\cdot)$, $\sqrt{(\cdot)}$.
 - Basis expansions: $X_2 = X_1^2$, $X_3 = X_1^3$, $X_3 = X_1 \cdot X_2$.

Estimation

- The issue of finding the regression function $E(Y \mid X)$ is converted to estimating β_j , $j = 0, 1, \dots, p$.
- Training data:

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}, \\ \text{where } x_i = (x_{i1}, x_{i2}, \dots, x_{ip}).$$

- Denote $\beta = (\beta_0, \beta_1, \dots, \beta_p)^T$.
- The loss function $E(Y - f(X))^2$ is approximated by the empirical loss $RSS(\beta)/N$:

$$\begin{aligned} RSS(\beta) &= \sum_{i=1}^N (y_i - f(x_i))^2 \\ &= \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2. \end{aligned}$$

Notation

- The input matrix \mathbf{X} of dimension $N \times (p + 1)$:

$$\begin{pmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,p} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{N,1} & x_{N,2} & \dots & x_{N,p} \end{pmatrix}$$

- Output vector \mathbf{y} :

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{pmatrix}$$

- The estimated β is $\hat{\beta}$.
- The fitted values at the training inputs are

$$\hat{y}_i = \hat{\beta}_0 + \sum_{j=1}^p x_{ij} \hat{\beta}_j$$

and

$$\hat{\mathbf{y}} = \begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \dots \\ \hat{y}_N \end{pmatrix}$$

Point Estimate

- The *least square estimation* of $\hat{\beta}$ is

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

- The fitted value vector is

$$\hat{\mathbf{y}} = \mathbf{X} \hat{\beta} = \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

- Hat matrix:

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

Geometric Interpretation

- Each column of \mathbf{X} is a vector in an N -dimensional space (NOT the p -dimensional feature vector space).

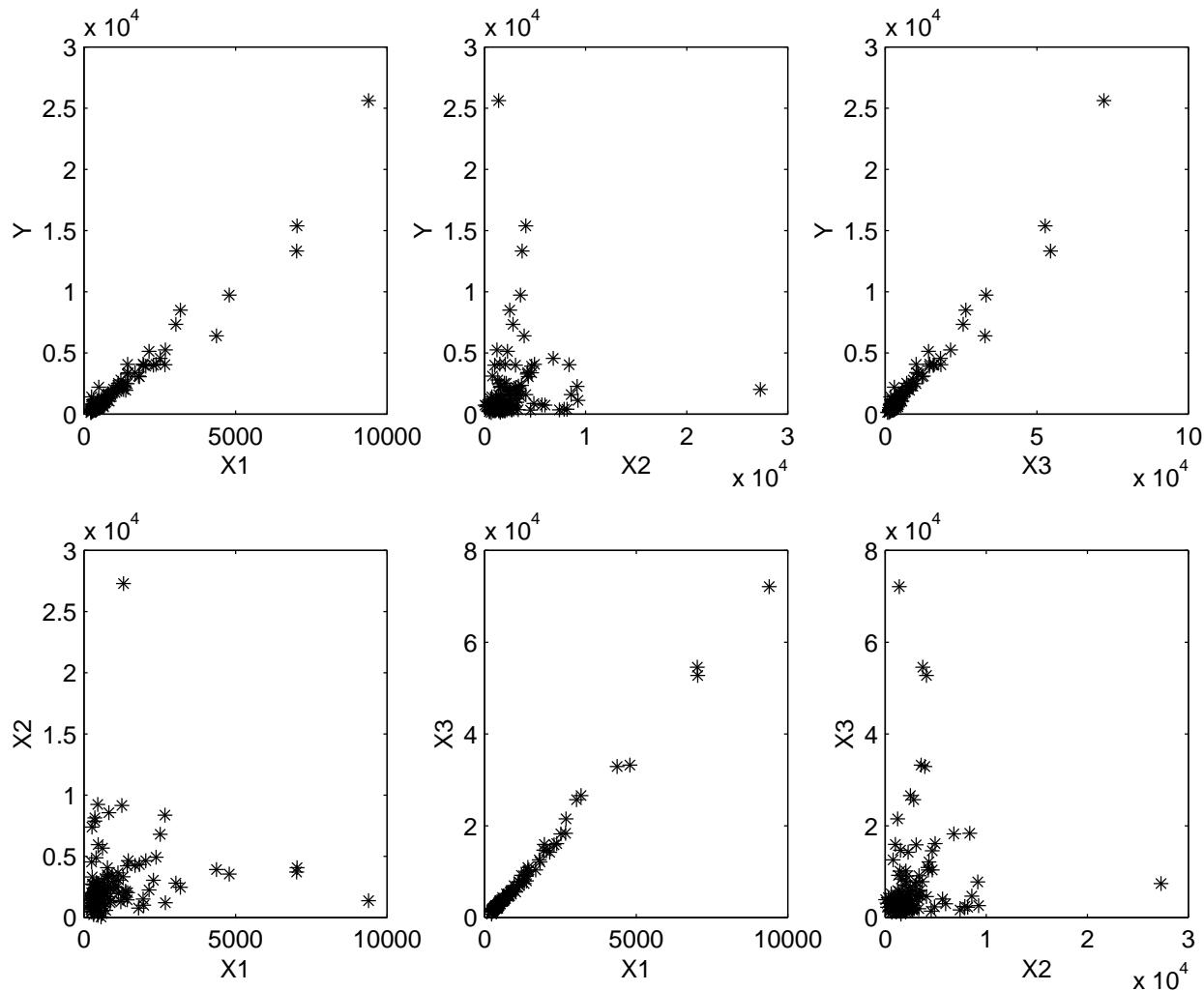
$$\mathbf{X} = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_p)$$

- The fitted output vector $\hat{\mathbf{y}}$ is a linear combination of the column vectors \mathbf{x}_j , $j = 0, 1, \dots, p$.
- $\hat{\mathbf{y}}$ lies in the subspace spanned by \mathbf{x}_j , $j = 0, 1, \dots, p$.
- $RSS(\hat{\beta}) = \| \mathbf{y} - \hat{\mathbf{y}} \|^2$.
- $\mathbf{y} - \hat{\mathbf{y}}$ is perpendicular to the subspace, i.e., $\hat{\mathbf{y}}$ is the projection of \mathbf{y} on the subspace.
- The geometric interpretation is very helpful for understanding coefficient shrinkage and subset selection.

Example Results

The SMSA problem

- $\hat{Y}_i = -143.89 + 0.341X_{i1} - 0.0193X_{i2} + 0.255X_{i3}$.
- $RSS(\hat{\beta}) = 52942336$.



If the Linear Model Is True

- $E(Y | X) = \beta_0 + \sum_{j=1}^p X_j \beta_j$
- The least square estimation of β is unbiased,

$$E(\hat{\beta}_j) = \beta_j \quad j = 0, 1, \dots, p .$$

- To draw inferences about β , further assume:

$$Y = E(Y | X) + \epsilon$$

where $\epsilon \sim N(0, \sigma^2)$ and is independent of X .

- X_{ij} are regarded as fixed, Y_i are random due to ϵ .
- Estimation accuracy:

$$Var(\hat{\beta}) = (\mathbf{X}^T \mathbf{X})^{-1} \sigma^2 .$$

- Under the assumption,

$$\hat{\beta} \sim N(\beta, (\mathbf{X}^T \mathbf{X})^{-1} \sigma^2) .$$

- Confidence intervals can be computed and significant tests can be done.

Gauss-Markov Theorem

- Assume the linear model is true.
- For any linear combination of the parameters β_0, \dots, β_p , denoted by $\theta = a^T \beta$, $a^T \hat{\beta}$ is an unbiased estimation since $\hat{\beta}$ is unbiased.
- The least squares estimate of θ is

$$\begin{aligned}\hat{\theta} &= a^T \hat{\beta} \\ &= a^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X} \mathbf{y} \\ &\triangleq \tilde{a}^T \mathbf{y},\end{aligned}$$

which is linear in \mathbf{y} .

- Suppose $c^T \mathbf{y}$ is another unbiased linear estimate of θ , i.e., $E(c^T \mathbf{y}) = \theta$.
- The least square estimate yields the minimum variance among all linear unbiased estimate.

$$Var(\tilde{a}^T \mathbf{y}) \leq Var(c^T \mathbf{y}).$$

- β_j , $j = 0, 1, \dots, p$ are special cases of $a^T \beta$, where a^T only has one non-zero element that equals 1.

Ridge Regression

Centered inputs

- Suppose \mathbf{x}_j , $j = 1, \dots, p$, are mean removed.
- $\hat{\beta}_0 = \bar{y} = \sum_{i=1}^N y_i / N$.
- If we remove the mean of y_i , we can assume

$$E(Y | X) = \sum_{j=1}^p \beta_j X_j$$

- Input matrix \mathbf{X} has p (rather than $p + 1$) columns.
- $\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$
- $\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$

Singular Value Decomposition (SVD)

- If the column vectors of \mathbf{X} are orthonormal, i.e., the variables $X_j, j = 1, 2, \dots, p$, are uncorrelated and have unit norm.

– $\hat{\beta}_j$ are the coordinates of \mathbf{y} on the orthonormal basis \mathbf{X} .

- In general

$$\boxed{\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T} .$$

- $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p)$ is an $N \times p$ orthogonal matrix.
 $\mathbf{u}_j, j = 1, \dots, p$ form an orthonormal basis for the space spanned by the column vectors of \mathbf{X} .
- $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p)$ is an $p \times p$ orthogonal matrix.
 $\mathbf{v}_j, j = 1, \dots, p$ form an orthonormal basis for the space spanned by the row vectors of \mathbf{X} .
- $\mathbf{D} = diag(d_1, d_2, \dots, d_p)$, $d_1 \geq d_2 \geq \dots \geq d_p \geq 0$ are the singular values of \mathbf{X} .

Principal Components

- The sample covariance matrix of \mathbf{X} is

$$\mathbf{S} = \mathbf{X}^T \mathbf{X} / N .$$

- Eigen decomposition of $\mathbf{X}^T \mathbf{X}$:

$$\begin{aligned}\mathbf{X}^T \mathbf{X} &= (\mathbf{U} \mathbf{D} \mathbf{V}^T)^T (\mathbf{U} \mathbf{D} \mathbf{V}^T) \\ &= \mathbf{V} \mathbf{D} \mathbf{U}^T \mathbf{U} \mathbf{D} \mathbf{V}^T \\ &= \mathbf{V} \mathbf{D}^2 \mathbf{V}^T\end{aligned}$$

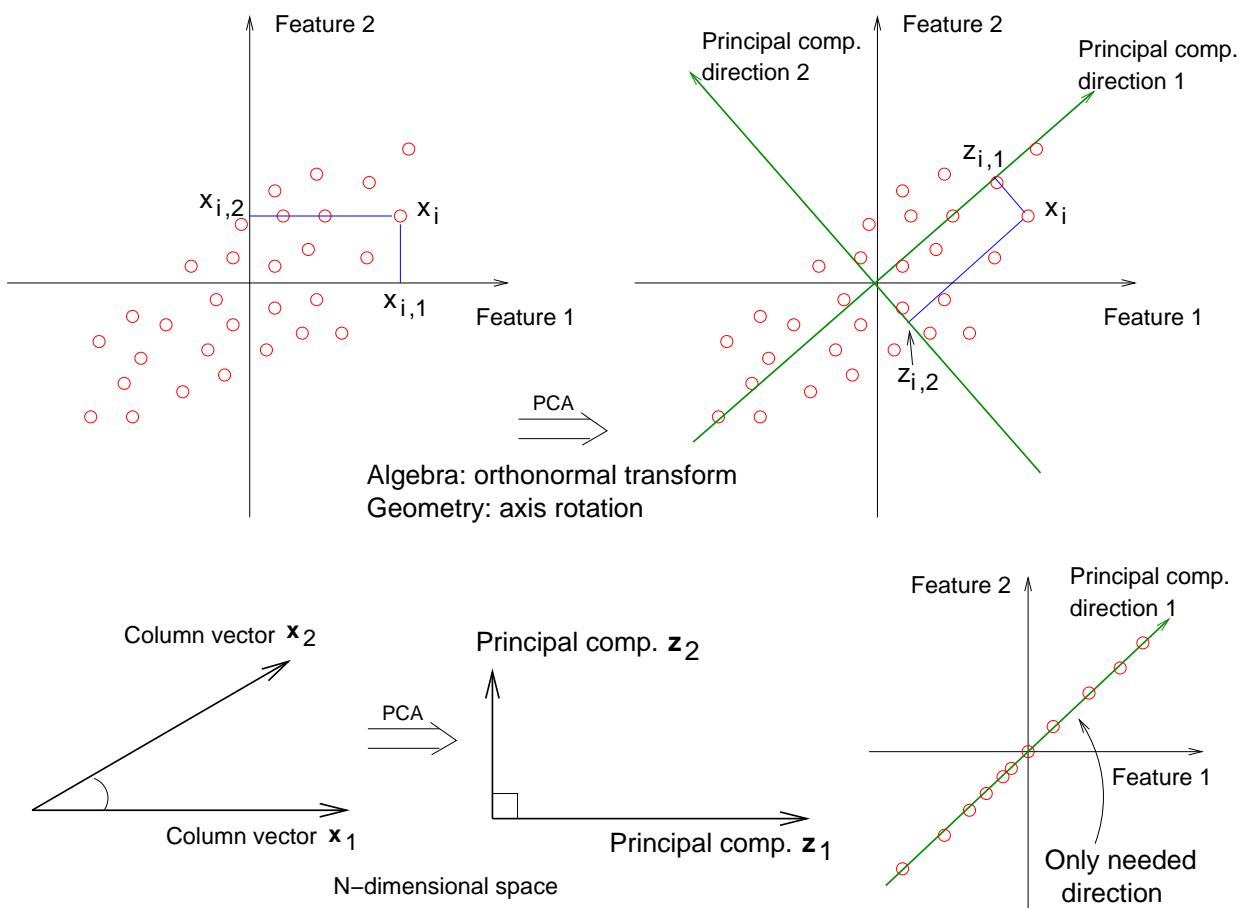
- The eigenvectors of $\mathbf{X}^T \mathbf{X}$, \mathbf{v}_j , are called *principal component direction* of \mathbf{X} .
- It's easy to see that $\mathbf{z}_j = \mathbf{X} \mathbf{v}_j = \mathbf{u}_j d_j$. Hence \mathbf{u}_j , is simply the projection of the row vectors of \mathbf{X} , i.e., the input predictor vectors, on the direction \mathbf{v}_j , scaled by d_j . For example

$$\mathbf{z}_1 = \begin{pmatrix} X_{1,1}v_{1,1} + X_{1,2}v_{1,2} + \cdots + X_{1,p}v_{1,p} \\ X_{2,1}v_{1,1} + X_{2,2}v_{1,2} + \cdots + X_{2,p}v_{1,p} \\ \vdots & \vdots & \vdots \\ X_{N,1}v_{1,1} + X_{N,2}v_{1,2} + \cdots + X_{N,p}v_{1,p} \end{pmatrix}$$

- The *principal components* of \mathbf{X} are $\mathbf{z}_j = d_j \mathbf{u}_j$, $j = 1, \dots, p$.
- The first principal component of \mathbf{X} , \mathbf{z}_1 , has the largest sample variance amongst all normalized linear combinations of the columns of \mathbf{X} .

$$Var(\mathbf{z}_1) = d_1^2/N .$$

- Subsequent principal components \mathbf{z}_j have maximum variance d_j^2/N , subject to being orthogonal to the earlier ones.



Ridge Regression

- Minimize a penalized residual sum of squares

$$\hat{\beta}^{ridge} = \operatorname{argmin}_{\beta} \left\{ \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\}$$

- Equivalently

$$\begin{aligned} \hat{\beta}^{ridge} &= \operatorname{argmin}_{\beta} \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2 \\ \text{subject to } & \sum_{j=1}^p \beta_j^2 \leq s . \end{aligned}$$

- λ or s controls the model complexity.

Solution

- With centered inputs,

$$RSS(\lambda) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta^T\beta ,$$

and

$$\hat{\beta}^{ridge} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

- Solution exists even when $\mathbf{X}^T\mathbf{X}$ is singular, i.e., has zero eigen values.
- When $\mathbf{X}^T\mathbf{X}$ is ill-conditioned (nearly singular), the ridge regression solution is more robust.

Geometric Interpretation

- Center inputs.
- Consider the fitted response

$$\begin{aligned}
 \hat{\mathbf{y}} &= \mathbf{X} \hat{\beta}^{ridge} \\
 &= \mathbf{X}(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y} \\
 &= \mathbf{U} \mathbf{D} (\mathbf{D}^2 + \lambda \mathbf{I})^{-1} \mathbf{D} \mathbf{U}^T \mathbf{y} \\
 &= \sum_{j=1}^p \mathbf{u}_j \frac{d_j^2}{d_j^2 + \lambda} \mathbf{u}_j^T \mathbf{y} ,
 \end{aligned}$$

where \mathbf{u}_j are the normalized principal components of \mathbf{X} .

- Ridge regression shrinks the coordinates with respect to the orthonormal basis formed by the principal components.
- Coordinate with respect to the principal component with a smaller variance is shrunk more.

- Instead of using $X = (X_1, X_2, \dots, X_p)$ as predicting variables, use the transformed variables

$$(X\mathbf{v}_1, X\mathbf{v}_2, \dots, X\mathbf{v}_p)$$

as predictors.

- The input matrix is $\tilde{\mathbf{X}} = \mathbf{U}\mathbf{D}$ (Note $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$).
- Then for the new inputs

$$\hat{\beta}_j^{ridge} = \frac{d_j}{d_j^2 + \lambda} \mathbf{u}_j^T \mathbf{y} .$$

$$Var(\hat{\beta}_j) = \frac{\sigma^2}{d_j^2}$$

where σ^2 is the variance of the error term ϵ in the linear model.

- The factor of shrinkage given by ridge regression is

$$\frac{d_j^2}{d_j^2 + \lambda} .$$

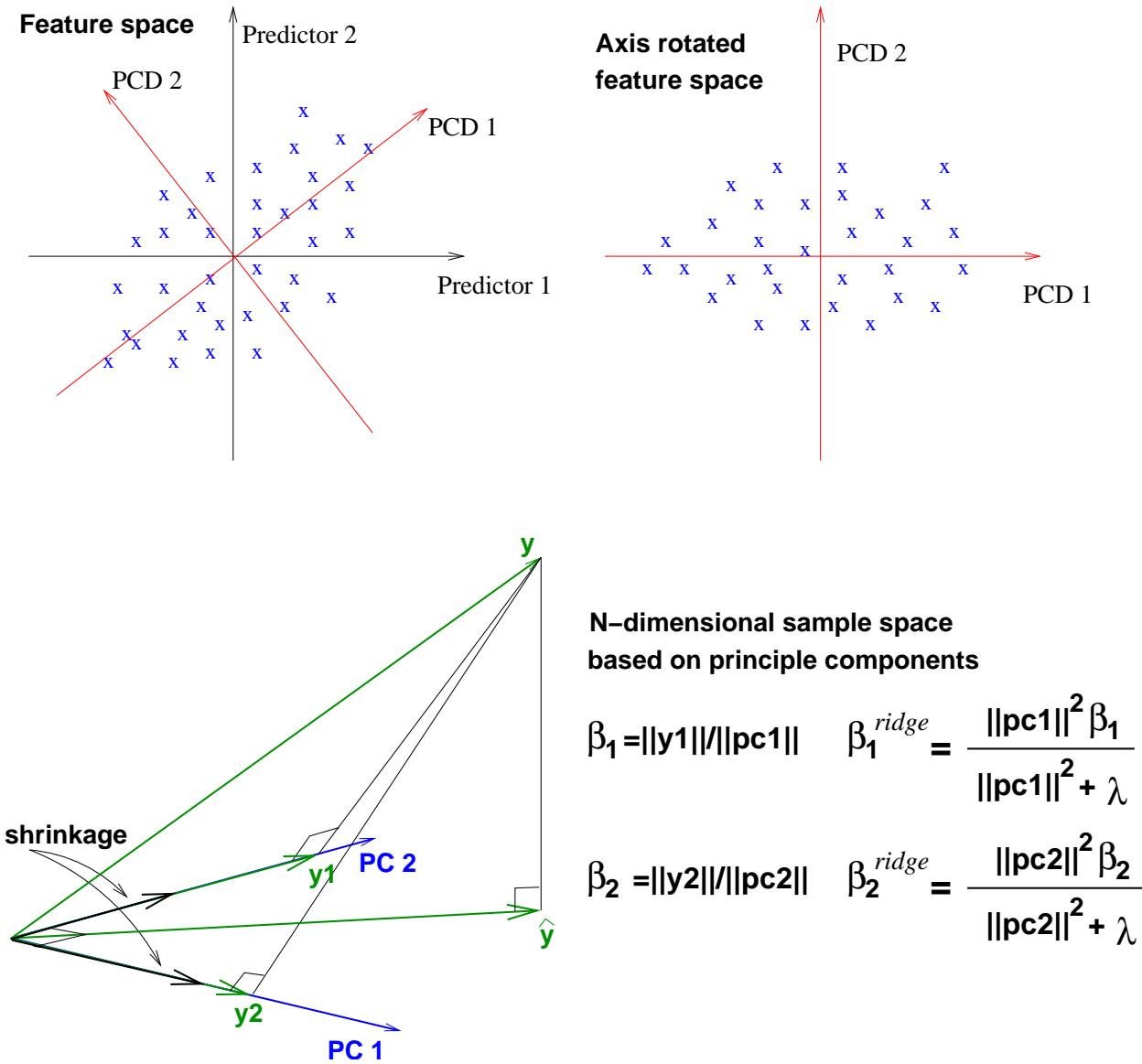


Figure 6: The Geometric interpretation of principal components and shrinkage by ridge regression.

Compare squared loss $E(\beta_j - \hat{\beta}_j)^2$

- Without shrinkage: σ^2/d_j^2 .
- With shrinkage: $Bias^2 + Variance$.

$$\begin{aligned}
 & (\beta_j - \beta_j \cdot \frac{d_j^2}{d_j^2 + \lambda})^2 + \frac{\sigma^2}{d_j^2} \cdot (\frac{d_j^2}{d_j^2 + \lambda})^2 \\
 &= \frac{\sigma^2}{d_j^2} \cdot \frac{d_j^2(d_j^2 + \lambda^2 \frac{\beta_j^2}{\sigma^2})}{(d_j^2 + \lambda)^2}
 \end{aligned}$$

- Consider the ratio between squared loss

$$\frac{d_j^2(d_j^2 + \lambda^2 \frac{\beta_j^2}{\sigma^2})}{(d_j^2 + \lambda)^2} .$$

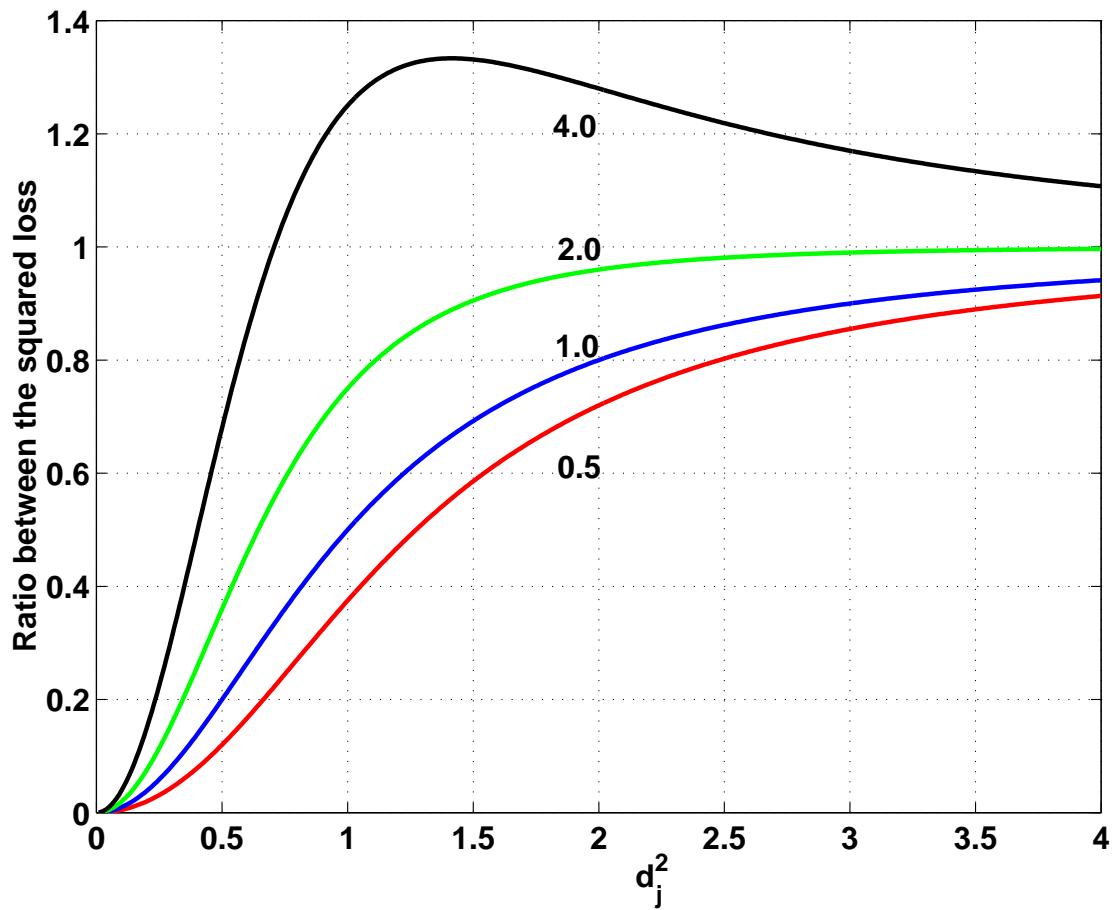


Figure 7: The ratio between the squared loss with and without shrinkage. The amount of shrinkage is set by $\lambda = 1.0$. The four curves correspond to $\beta^2/\sigma^2 = 0.5, 1.0, 2.0, 4.0$ respectively. When $\beta^2/\sigma^2 = 0.5, 1.0, 2.0$, shrinkage always leads to lower squared loss. When $\beta^2/\sigma^2 = 4.0$, shrinkage leads to lower squared loss when $d_j^2 \leq 0.71$. Shrinkage is more beneficial when d_j^2 is small.

Principal Components Regression (PCR)

- Instead of smoothly shrinking the coordinates on the principal components, PCR either does not shrink a coordinate at all or shrinks it to zero.
- Principal component regression forms the derived input columns $\mathbf{z}_m = \mathbf{X}\mathbf{v}_m$, and then regresses \mathbf{y} on $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_M$ for some $M \leq p$.
- Principal components regression discards the $p - M$ smallest eigenvalue components.

The Lasso

- The lasso estimate is defined by

$$\hat{\beta}^{lasso} = \operatorname{argmin}_{\beta} \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2$$

subject to $\sum_{j=1}^p |\beta_j| \leq s$

- Comparison with ridge regression: L_2 penalty $\sum_{j=1}^p \beta_j^2$ is replaced by the L_1 lasso penalty $\sum_{j=1}^p |\beta_j|$.
- Some of the coefficients may be shrunk to exactly zero.
- Orthonormal columns in \mathbf{X} are assumed in the following figure.

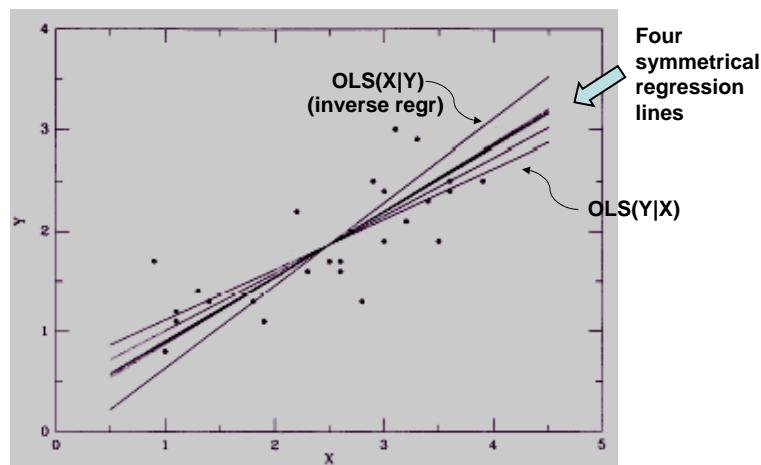
Linear regression issues in astronomy

Eric Feigelson
Summer School in astrostatistics

References
Isobe, Feigelson, Akritas & Babu, ApJ 364, 105 1990
Feigelson & Babu, ApJ 397, 55 1992

Structural regression

Seeking the intrinsic relationship between two properties
without specifying ‘dependent’ and ‘independent’ variables



Analytical formulae for slopes of the 6 OLS lines

16

ISOBE, FEIGELSON, AKRITAS, AND BABU

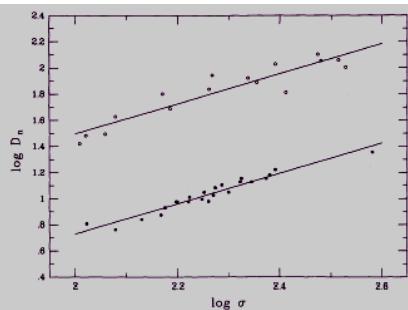
Vol. 364

TABLE I
LINEAR REGRESSION FORMULAE FOR SLOPES

Method	Expression for Slope	Estimate of the Variance of the Slope $\widehat{\text{Var}}(\beta_i)$
OLS($X Y$)	$\beta_1 = \frac{S_{xy}}{S_{xx}}$	$\frac{1}{S_{xx}^2} \left[\sum_{i=1}^n (x_i - \bar{x})^2 (y_i - \beta_1 x_i - \bar{y} + \beta_1 \bar{x})^2 \right]$
OLS($Y X$)	$\beta_2 = \frac{S_{yy}}{S_{xx}}$	$\frac{1}{S_{yy}^2} \left[\sum_{i=1}^n (y_i - \bar{y})^2 (y_i - \beta_2 x_i - \bar{y} + \beta_2 \bar{x})^2 \right]$
OLS bisector	$\beta_3 = (\beta_1 + \beta_2)^{-1} [\beta_1 \beta_2 - 1 + \sqrt{(1 + \beta_1^2)(1 + \beta_2^2)}]$	$\frac{\beta_3^2}{(\beta_1 + \beta_2)^2(1 + \beta_1^2)(1 + \beta_2^2)} [(1 + \beta_3^2)^2 \widehat{\text{Var}}(\beta_1) + 2(1 + \beta_1^2)(1 + \beta_2^2) \widehat{\text{Cov}}(\beta_1, \beta_2) + (1 + \beta_2^2)^2 \widehat{\text{Var}}(\beta_2)]$
Orthogonal regression	$\beta_4 = \frac{1}{2}[(\beta_2 - \beta_1^{-1}) + \text{Sign}(S_{xy})\sqrt{4 + (\beta_2 - \beta_1^{-1})^2}]$	$\frac{\beta_2^2}{4\beta_1^2 + (\beta_1 \beta_2 - 1)^2} [\beta_1^{-2} \widehat{\text{Var}}(\beta_1) + 2 \widehat{\text{Cov}}(\beta_1, \beta_2) + \beta_1^2 \widehat{\text{Var}}(\beta_2)]$
Reduced major-axis	$\beta_5 = \text{Sign}(S_{xy})\beta_1 \beta_2^{1/2}$	$\frac{1}{4} \left[\frac{\beta_2}{\beta_1} \widehat{\text{Var}}(\beta_1) + 2 \widehat{\text{Cov}}(\beta_1, \beta_2) + \frac{\beta_1}{\beta_2} \widehat{\text{Var}}(\beta_2) \right]$

NOTE.—An estimate of covariance term is given by:

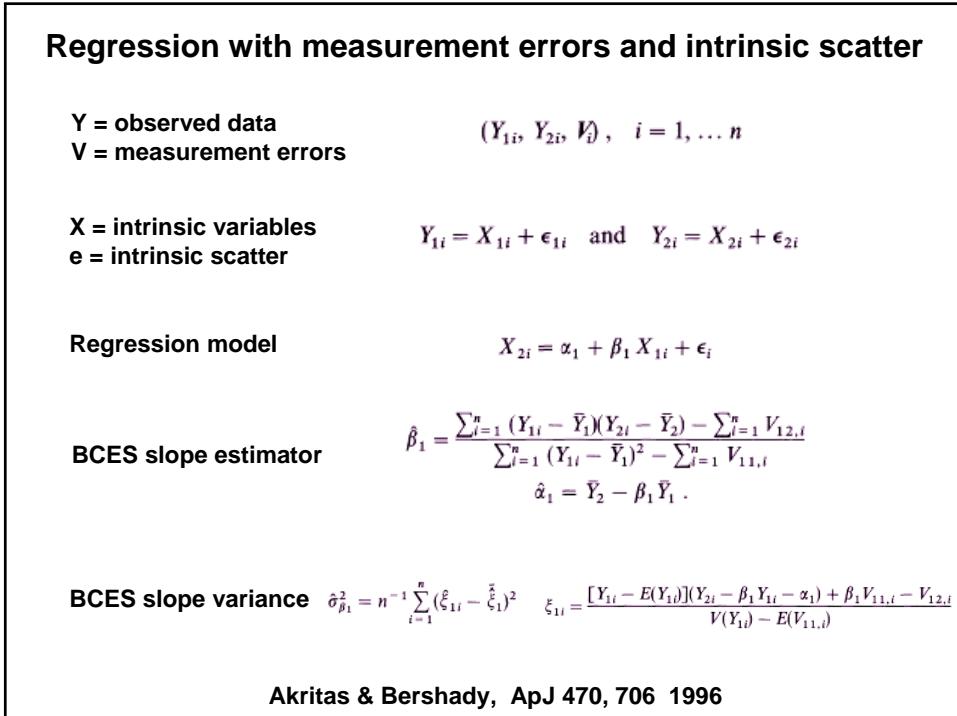
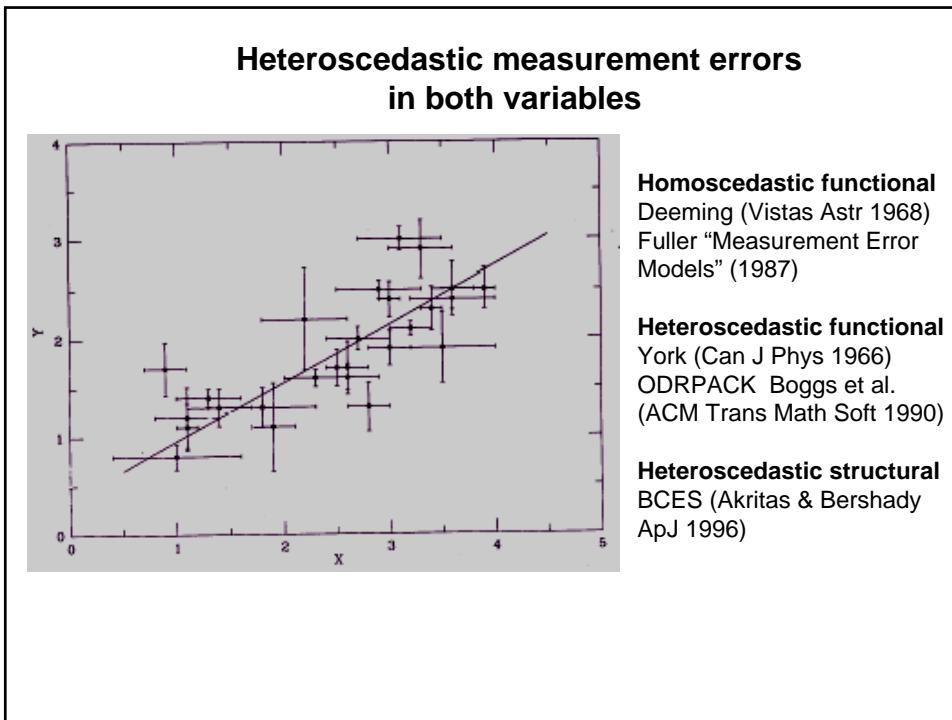
$$\widehat{\text{Cov}}(\beta_1, \beta_2) = (\beta_1 S_{xx}^2)^{-1} \left\{ \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})(y_i - \beta_1(x_i - \bar{x}))[(y_i - \bar{y} - \beta_2(x_i - \bar{x}))] \right\}.$$



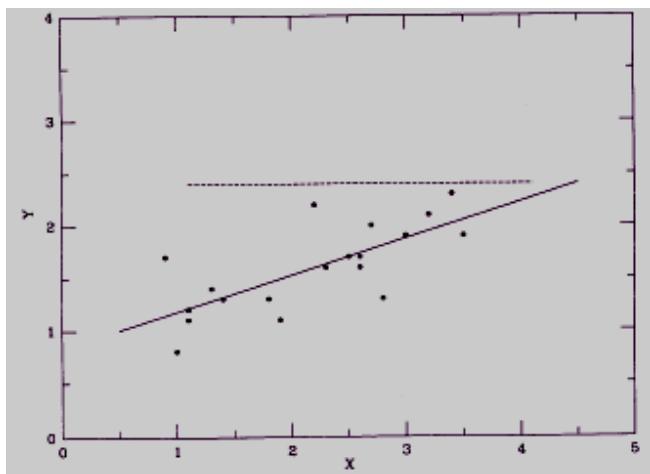
Example: Faber-Jackson relation
between diameter and stellar velocity
dispersion of elliptical galaxies

TABLE 4
REGRESSIONS FOR COMA AND VIRGO $\log D_v$ VERSUS $\log \sigma^*$

METHOD (1)	ASYMPTOTIC FORMULAE		BOOTSTRAP SLOPE (4)	JACKKNIFE SLOPE (5)
	Intercept (2)	Slope (3)		
23 Coma Ellipticals				
OLS($Y X$)	-1.595 ± 0.186	1.162 ± 0.082	1.186 ± 0.094	1.164 ± 0.111
OLS($X Y$)	-1.765 ± 0.216	1.238 ± 0.096	1.261 ± 0.104	1.239 ± 0.128
OLS bisector	-1.678 ± 0.200	1.199 ± 0.088	1.223 ± 0.099	1.201 ± 0.119
Orthogonal	-1.694 ± 0.209	1.206 ± 0.092	1.231 ± 0.102	1.208 ± 0.124
Reduced major axis	-1.679 ± 0.200	1.199 ± 0.088	1.223 ± 0.099	1.201 ± 0.119
OLS mean	-1.680 ± 0.200	1.200 ± 0.088	1.224 ± 0.099	1.201 ± 0.119
16 Virgo Ellipticals				
OLS($Y X$)	-0.790 ± 0.230	1.144 ± 0.101	1.143 ± 0.127	1.114 ± 0.118
OLS($X Y$)	-1.183 ± 0.180	1.316 ± 0.082	1.322 ± 0.132	1.316 ± 0.093
OLS bisector	-0.978 ± 0.190	1.227 ± 0.085	1.227 ± 0.107	1.226 ± 0.099
Orthogonal	-1.021 ± 0.198	1.245 ± 0.089	1.246 ± 0.121	1.245 ± 0.104
Reduced major axis	-0.979 ± 0.190	1.227 ± 0.085	1.228 ± 0.108	1.227 ± 0.099
OLS mean	-0.986 ± 0.188	1.230 ± 0.084	1.233 ± 0.110	1.230 ± 0.098



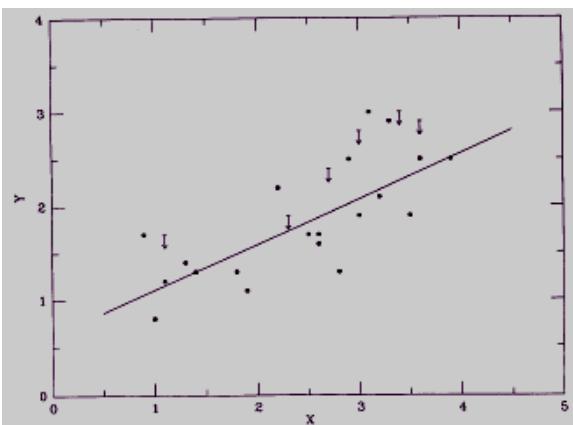
Truncation due to flux limits



Econometrics: Tobit & LIMDEP models (Amemiya, Advanced econometrics 1985; Maddala, Limited-dependent & Quantitative Variables in Econometrics 1983)

Astronomy: Malmquist bias in Hubble diagram (Deeming, Vistas Astr 1968, Segal, PNAS 1975)

Censoring due to non-detections



Correlation coefficients:
Generalized Kendall's τ (Brown, Hollander & Korwar 1974)

Linear regression with normal residuals:
EM Algorithm (Wolynetz Appl Stat 1979)

Linear regression with Kaplan-Meier residuals:
Buckley & James (Biometrika 1979) Schmitt (ApJ 1985)

Presented for astronomy by Isobe, Feigelson & Nelson (ApJ 1986)
Implemented in Astronomy Survival Analysis (ASURV) package

Bayesian treatment of measurement errors in linear regression
 (Brandon C. Kelly, ApJ 2007)

Errors-in-variable regression model (cf. monograph Fuller 1987):

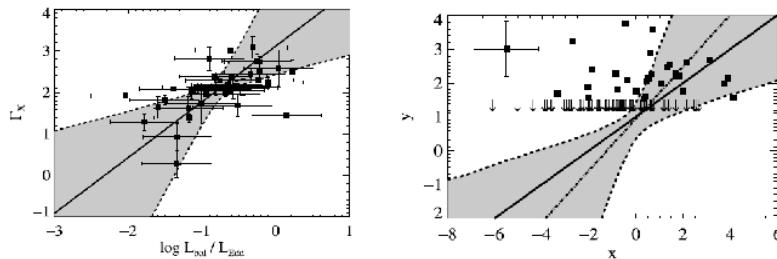
$$\begin{aligned}\eta_i &= \alpha + \beta \xi_i + \varepsilon_i && \text{(True relationship)} \\ x &= \xi_i + \varepsilon_{x,i} && \text{(True variables indirectly observed} \\ y &= \eta_i + \varepsilon_{y,i} && \text{with measurement error)}\end{aligned}$$

ξ_i is modeled as a mixture of normals $N(\mu, \tau)$. The MLE is found by maximizing the following likelihood using the EM Algorithm:

$$\begin{aligned}p(x, y | \theta, \psi) &= \prod_{i=1}^n \sum_{k=1}^K \frac{\pi_k}{2\pi |\mathbf{V}_{k,i}|^{1/2}} \\ &\quad \times \exp \left[-\frac{1}{2} (\mathbf{z}_i - \boldsymbol{\zeta}_k)^T \mathbf{V}_{k,i}^{-1} (\mathbf{z}_i - \boldsymbol{\zeta}_k) \right], \\ \boldsymbol{\zeta}_k &= (\alpha + \beta \mu_k, \mu_k), \\ \mathbf{V}_{k,i} &= \begin{pmatrix} \beta^2 \tau_k^2 + \sigma^2 + \sigma_{y,i}^2 & \beta \tau_k^2 + \sigma_{xy,i} \\ \beta \tau_k^2 + \sigma_{xy,i} & \tau_k^2 + \sigma_{x,i}^2 \end{pmatrix},\end{aligned}$$

Prior distributions are assigned to the parameters $(\alpha, \beta, \sigma, \tau, \mu)$, Bayes' Theorem is applied, and posterior distributions are computed with Markov chain Monte Carlo techniques.

The method can be applied to censored and truncated regression problems, as well as measurement error problems. Performance is demonstrably better than earlier de-biased least-squares solutions (BCES, FITEXY). IDL code is available.



Conclusions

Bivariate linear regression in astronomy can be surprisingly complex. Pay attention to precise question being asked, and details of situation. Several codes available through <http://astrostatistics.psu.edu/statcodes>.

- Functional vs. structural regression
- Symmetrical vs. dependent regression
- Weighting by measurement error
- Truncation & censoring due to flux limits

Other topics not considered here (some covered later in the Summer School):

- Robust & rank regression techniques to treat outliers
- Goodness-of-fit, model selection and parsimony
- Nonlinear regression
- Multivariate regression

Exploratory Data Analysis (EDA) and Regression

This tutorial demonstrates some of the capabilities of R for exploring relationships among two (or more) quantitative variables.

Bivariate exploratory data analysis

We begin by loading the Hipparcos dataset used in the [descriptive statistics](#) tutorial, found at http://astrostatistics.psu.edu/datasets/HIP_star.html. Type

```
hip <- read.table("http://astrostatistics.psu.edu/datasets/HIP_star.dat",
  header=T, fill=T)
names(hip)
attach(hip)
```

In the [descriptive statistics](#) tutorial, we considered boxplots, a one-dimensional plotting technique. We may perform a slightly more sophisticated analysis using boxplots to get a glimpse at some bivariate structure. Let us examine the values of Vmag, with objects broken into categories according to the B minus V variable:

```
boxplot(Vmag~cut(B.V,breaks=(-1:6)/2),
  notch=T, varwidth=T, las=1, tcl=.5,
  xlab=expression("B minus V"),
  ylab=expression("V magnitude"),
  main="Can you find the red giants?",
  cex=1, cex.lab=1.4, cex.axis=.8, cex.main=1)
axis(2, labels=F, at=0:12, tcl=-.25)
axis(4, at=3*(0:4))
```

The notches in the boxes, produced using "notch=T", can be used to test for differences in the medians (see [boxplot.stats](#) for details). With "varwidth=T", the box widths are proportional to the square roots of the sample sizes. The "cex" options all give scaling factors, relative to default: "cex" is for plotting text and symbols, "cex.axis" is for axis annotation, "cex.lab" is for the x and y labels, and "cex.main" is for main titles. The two [axis](#) commands are used to add an axis to the current plot. The first such command above adds smaller tick marks at all integers, whereas the second one adds the axis on the right.

Scatterplots

The [boxplots](#) in the plot above are telling us something about the bivariate relationship between the two variables. Yet it is probably easier to grasp this relationship by producing a [scatterplot](#).

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

The above plot looks too busy because of the default plotting character, set let us use a different one:

```
plot(Vmag,B.V,pch=".")
```

Let's now use exploratory scatterplots to locate the Hyades stars. This open cluster should be concentrated both in the sky coordinates RA and DE, and also in the proper motion variables pm_RA and pm_DE. We start by noticing a concentration of stars in the RA distribution:

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

See the cluster of stars with RA between 50 and 100 and with DE between 0 and 25?

```
rect(50,0,100,25,border=2)
```

Let's construct a logical (TRUE/FALSE) variable that will select only those stars in the appropriate rectangle:

```
filter1 <- (RA>50 & RA<100 & DE>0 & DE<25)
```

Next, we select in the proper motions. (As our cuts through the data are parallel to the axes, this variable-by-variable classification approach is sometimes called Classification and Regression Trees or CART, a very common multivariate classification procedure.)

```
plot(pmRA[filter1],pmDE[filter1],pch=20)
rect(0,-150,200,50,border=2)
```

Let's replot after zooming in on the rectangle shown in red.

```
plot(pmRA[filter1],pmDE[filter1],pch=20, xlim=c(0,200),ylim=c(-150,50))
rect(90,-60,130,-10,border=2)
filter2 <- (pmRA>90 & pmRA<130 & pmDE>-60 & pmDE< -10) # Space in 'pmDE< -10' is necessary!
filter <- filter1 & filter2
```

Let's have a final look at the stars we have identified using the [pairs](#) command to produce all bivariate plots for pairs of variables. We'll exclude the first and fifth columns (the HIP identifying number and the parallax, which is known to lie in a narrow band by construction).

```
pairs(hip[filter,-c(1,5)],pch=20)
```

Notice that indexing a matrix or vector using negative integers has the effect of *excluding* the corresponding entries.

We see that there is one outlying star in the e_Plx variable, indicating that its measurements are not reliable. We exclude this point:

```
filter <- filter & (e_Plx<5)
pairs(hip[filter,-c(1,5)],pch=20)
```

How many stars have we identified? The filter variable, a vector of TRUE and FALSE, may be summed to reveal the number of TRUE's (summation causes R to coerce the logical values to 0's and 1's).

```
sum(filter)
```

As a final look at these data, let's consider the HR plot of Vmag versus B.V but make the 92 Hyades stars we just identified look bigger (pch=20 instead of 46) and color them red (col=2 instead of 1). This shows the Zero Age Main Sequence, plus four red giants, with great precision.

```
plot(Vmag,B.V,pch=c(46,20)[1+filter], col=1+filter,
      xlim=range(Vmag[filter]), ylim=range(B.V[filter]))
```

Linear and polynomial regression

Here is how one may reproduce the output seen in the regression lecture, i.e., a linear regression relating BminusV to logL, where logL is the luminosity, defined to be $(15 - \text{Vmag} - 5 \log(\text{Plx})) / 2.5$. However, we'll use a different subset of the data than the one seen in the lecture, namely, the main-sequence Hyades:

```
mainseqhyades <- filter & (Vmags>4 | B.V<0.2)
logL <- (15 - Vmag - 5 * log10(Plx)) / 2.5
x <- logL[mainseqhyades]
y <- B.V[mainseqhyades]
plot(x, y)
regline <- lm(y~x)
abline(regline, lwd=2, col=2)
```

```
summary(regrline)
```

Note that the regression line passes exactly through the point (xbar, ybar):

```
points(mean(x), mean(y), col=3, pch=20, cex=3)
```

Here is a regression of y on $\exp(-x/4)$:

```
newx <- exp(-x/4)
regrline2 <- lm(y~newx)
xseq <- seq(min(x), max(x), len=250)
lines(xseq, regrline2$coef %*% rbind(1, exp(-xseq/4))), lwd=2, col=3)
```

For an implementation of the ridge regression technique mentioned in lecture, see the [lm.ridge](#) function in the MASS package. To use this function, you must first type library(MASS). For a package that implements LASSO (which uses an L1-penalty instead of the L2-penalty of ridge regression), check out, e.g., the lasso2 package on CRAN.

Let's consider a new dataset, one that 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 <- read.table("http://astrostatistics.psu.edu/datasets/GRB_afterglow.dat",
header=T, skip=1)
```

The skip=1 option in the previous statement tells R to ignore the first row in the data file. You will see why this is necessary if you look at the [file](#). Let's focus on the first two columns, which are times and X-ray fluxes:

```
plot(grb[,1:2], xlab="time", ylab="flux")
```

This plot is very hard to interpret because of the scales, so let's take the log of each variable:

```
x <- log(grb[,1])
y <- log(grb[,2])
plot(x,y, xlab="log time", ylab="log flux")
```

The relationship looks roughly linear, so let's try a linear model ([lm](#) in R):

```
model1 <- lm(y~x)
abline(model1, col=2, lwd=2)
```

The "response ~ predictor(s)" format seen above is used for model formulas in functions like [lm](#).

The model1 object just created is an object of [class "lm"](#). The class of an object in R can help to determine how it is treated by functions such as [print](#) and [summary](#).

```
model1 # same as print(model1)
summary(model1)
```

Notice the sigma-hat, the R-squared and adjusted R-squared, and the standard errors of the beta-hats in the output from the summary function.

There is a lot of information contained in model1 that is not displayed by [print](#) or [summary](#):

```
names(model1)
```

For instance, we will use the model1\$fitted.values and model1\$residuals information later when we look at some residuals plots.

Notice that the coefficient estimates are listed in a regression table, which is standard regression output for any software package. This table gives not only the estimates but their standard errors as well, which enables us to determine whether the estimates are very different from zero. It is possible to give individual

confidence intervals for both the intercept parameter and the slope parameter based on this information, but remember that a line really requires both a slope **and** an intercept. Since our goal is really to estimate a line here, maybe it would be better if we could somehow obtain a confidence "interval" for the lines themselves.

This may in fact be accomplished. By viewing a line as a single two-dimensional point in (intercept, slope) space, we set up a one-to-one correspondence between all (nonvertical) lines and all points in two-dimensional space. It is possible to obtain a two-dimensional confidence *ellipse* for the (intercept, slope) points, which may then be mapped back into the set of lines to see what it looks like.

Performing all the calculations necessary to do this is somewhat tedious, but fortunately, someone else has already done it and made it available to all R users through CRAN, the [Comprehensive R Archive Network](#). The necessary functions are part of the "car" (companion to applied regression) package, which may be installed onto the V: drive (we don't have write access to the default location where R packages are installed) as follows:

```
install.packages("car", lib="V:/") # lib=... is not always necessary!
```

You will have to choose a CRAN mirror as part of the installation process. Once the car package is installed, its contents can be loaded into the current R session using the [library](#) function:

```
library(car, lib.loc="V:/")
```

If all has gone well, there is now a new set of functions, along with relevant documentation. Here is a 95% confidence ellipse for the (intercept, slope) pairs:

```
confidence.ellipse(model1)
```

Remember that each point on the boundary or in the interior of this ellipse represents a line. If we were to plot all of these lines on the original scatterplot, the region they described would be a 95% confidence band for the true regression line. My advisor, Dave Hunter, and I wrote a simple function to draw the borders of this band on a scatterplot. You can see this function at www.stat.psu.edu/~dhunter/R/confidence.band.r; to read it into R, use the [source](#) function:

```
source("http://www.stat.psu.edu/~dhunter/R/confidence.band.r")
confidence.band(model1)
```

In this dataset, the confidence band is so narrow that it's hard to see. However, the borders of the band are not straight. You can see the curvature much better when there are fewer points or more variation, as in:

```
tmpx <- 1:10
tmpy <- 1:10+rnorm(10) # Add random Gaussian noise
confidence.band(lm(tmpy~tmpx))
```

Also note that increasing the sample size increases the precision of the estimated line, thus narrowing the confidence band. Compare the previous plot with the one obtained by replicating tmpx and tmpy 25 times each:

```
tmpx25 <- rep(tmpx, 25)
tmpy25 <- rep(tmpy, 25)
confidence.band(lm(tmpy25~tmpx25))
```

A related phenomenon is illustrated if we are given a value of the predictor and asked to predict the response. Two types of intervals are commonly reported in this case: A *prediction* interval for an individual observation with that predictor value, and a *confidence* interval for the mean of all individuals with that predictor value. The former is always wider than the latter because it accounts for not only the uncertainty in estimating the true line but also the individual variation around the true line. This phenomenon may be illustrated as follows. Again, we use a toy data set here because the effect is harder to observe on our astronomical dataset. As usual, 95% is the default confidence level.

```
confidence.band(lm(tmpy~tmpx))
```

```

predict(lm(tmpy~tmpx), data.frame(tmpx=7), interval="prediction")
text(c(7,7,7), .Last.value, "P", col=4)
predict(lm(tmpy~tmpx), data.frame(tmpx=7), interval="conf")
text(c(7,7,7), .Last.value, "C", col=5)

```

Polynomial curve-fitting: Still linear regression!

Because there appears to be a bit of a bend in the scatterplot, let's try fitting a quadratic curve instead of a linear curve. **Note: Fitting a quadratic curve is still considered linear regression.** This may seem strange, but the reason is that the quadratic regression model assumes that the response y is a *linear* combination of 1, x , and x^2 . Notice the special form of the [lm](#) command when we implement quadratic regression. The [I](#) function means "as is" and it resolves any ambiguity in the model formula:

```

model2 <- lm(y~x+I(x^2))
summary(model2)

```

Here is how to find the estimates of beta using the closed-form solution seen in lecture:

```

x <- cbind(1, x, x^2) # Create nx3 X matrix
solve(t(X) %*% X) %*% t(X) %*% y # Compare to the coefficients above

```

Plotting the quadratic curve is not a simple matter of using the [abline](#) function. To obtain the plot, we'll first create a sequence of x values, then apply the linear combination implied by the regression model using matrix multiplication:

```

xx <- seq(min(x), max(x), len=200)
yy <- model2$coef %*% rbind(1, xx, xx^2)
lines(xx, yy, lwd=2, col=3)

```

Diagnostic residual plots

Comparing the (red) linear fit with the (green) quadratic fit visually, it does appear that the latter looks slightly better. However, let's check some diagnostic residual plots for these two models. To do this, we'll use the [plot.lm](#) command, which is capable of producing six different types of diagnostic plots. We will only consider two of the six: A plot of residuals versus fitted values and a normal quantile-quantile (Q-Q) plot.

```
plot.lm(model1, which=1:2)
```

It is not actually necessary to type [plot.lm](#) in the previous command; [plot](#) would have worked just as well. This is because `model1` is an object of [class](#) "lm" -- a fact that can be verified by typing `"class(model1)"` -- and so R knows to apply the function [plot.lm](#) if we simply type `"plot(model1, which=1:2)"`.

Looking at the first plot, residuals vs. fitted, we immediately see a problem with model 1. A "nice" residual plot should have residuals both above and below the zero line, with the vertical spread around the line roughly of the same magnitude no matter what the value on the horizontal axis. Furthermore, there should be no obvious curvature pattern. The red line is a [lowess](#) smoother produced to help discern any patterns (more on lowess later), but this line is not necessary in the case of `model1` to see the clear pattern of negative residuals on the left, positive in the middle, and negative on the right. There is curvature here that the model missed!

Pressing the return key to see the second plot reveals a normal quantile-quantile plot. The idea behind this plot is that it will make a random sample from a normal distribution look like a straight line. To the extent that the normal Q-Q plot does not look like a straight line, the assumption of normality of the residuals is

suspicious. For model1, the clear S-shaped pattern indicates non-normality of the residuals.

How do the same plots look for the quadratic fit?

```
plot(model2, which=1:2)
```

These plots look much better. There is a little bit of waviness in the residuals vs. fitted plot, but the pattern is nowhere near as obvious as it was before. And there appear to be several outliers among the residuals on the normal Q-Q plot, but the normality assumption looks much less suspect here.

The residuals we have been using in the above plots are the ordinary residuals. However, it is important to keep in mind that even if all of the assumptions of the regression model are perfectly true (including the assumption that all errors have the same variance), the variances of the *residuals* are not equal. For this reason, it is better to use the studentized residuals. Unfortunately, R reports the ordinary residuals by default and it is necessary to call another function to obtain the studentized residuals. The good news is that in most datasets, residual plots using the studentized residuals are essentially indistinguishable in shape from residual plots using the ordinary residuals, which means that we would come to the same conclusions regardless of which set of residuals we use.

```
rstu <- rstudent(model2)
plot(model2$fit, rstu)
```

To see how similar the studentized residuals are to a scaled version of the ordinary residuals (called the standardized residuals), we can depict both on the same plot:

```
rsta <- rstandard(model2)
points(model2$fit, rsta, col=2, pch=3)
```

Collinearity and variance inflation factors

Let's check the variance inflation factors (VIFs) for the quadratic fit. The car package that we installed earlier contains a function called vif that does this automatically. Check its help page by typing "?vif" if you wish. Note that it does not make sense to look at variance inflation factors for model1, which has only one term (try it and see what happens). So we'll start by examining model2.

```
vif(model2)
```

The VIFs of more than 70 indicate a high degree of collinearity between the values of x and x² (the two predictors). This is not surprising, since x has a range from about 5 to 13. In fact, it is easy to visualize the collinearity in a plot:

```
plot(x,x^2) # Note highly linear-looking plot
```

To correct the collinearity, we'll replace x and x² by (x-m) and (x-m)², where m is the sample mean of x:

```
centered.x <- x-mean(x)
model2.2 <- lm(y ~ centered.x + I(centered.x^2))
```

This new model has much lower VIFs, which means that we have greatly reduced the collinearity. However, the fit is exactly the same: It is still the best-fitting quadratic curve. We may demonstrate this by plotting both fits on the same set of axes:

```
plot(x,y,xlab="log time",ylab="log flux")
yy2 <- model2.2$coef %*% rbind(1, xx-mean(x), (xx-mean(x))^2)
lines(xx, yy, lwd=2, col=2)
lines(xx, yy2, lwd=2, col=3, lty=2)
```

Model selection using AIC and BIC

Let's compare the AIC and BIC values for the linear and the quadratic fit. Without getting too deeply into details, the idea behind these criteria is that we know the model with more parameters (the quadratic model) should achieve a higher maximized log-likelihood than the model with fewer parameters (the linear model). However, it may be that the additional increase in the log-likelihood statistic achieved with more parameters is not worth adding the additional parameters. We may test whether it is worth adding the additional parameters by *penalizing* the log-likelihood by subtracting some positive multiple of the number of parameters. In practice, for technical reasons we take -2 times the log-likelihood, add a positive multiple of the number of parameters, and look for the smallest resulting value. For AIC, the positive multiple is 2; for BIC, it is the natural log of n, the number of observations. We can obtain both the AIC and BIC results using the [AIC](#) function. Remember that R is case-sensitive, so "AIC" must be all capital letters.

```
AIC(model1)
AIC(model2)
```

The value of AIC for model2 is smaller than that for model1, which indicates that model2 provides a better fit that is worth the additional parameters. However, AIC is known to tend to overfit sometimes, meaning that it sometimes favors models with more parameters than they should have. The BIC uses a larger penalty than AIC, and it often seems to do a slightly better job; however, in this case we see there is no difference in the conclusion:

```
n <- length(x)
AIC(model1, k=log(n))
AIC(model2, k=log(n))
```

It did not make any difference in the above output that we used model2 (with the uncentered x values) instead of model2.2 (with the centered values). However, if we had looked at the AIC or BIC values for a model containing ONLY the quadratic term but no linear term, then we would see a dramatic difference. Which one of the following would you expect to be higher (i.e., indicating a worse fit), and why?

```
AIC(lm(y~I(x^2)), k=log(n))
AIC(lm(y~I(centered.x^2)), k=log(n))
```

Other methods of curve-fitting

Let's try a nonparametric fit, given by [loess](#) or [lowess](#). First we plot the linear (red) and quadratic (green) fits, then we overlay the lowess fit in blue:

```
plot(x,y,xlab="log time",ylab="log flux")
abline(model1, lwd=2, col=2)
lines(xx, yy, lwd=3, col=3)
npmodel1 <- lowess(y~x)
lines(npmodel1, col=4, lwd=2)
```

It is hard to see the pattern of the lowess curve in the plot. Let's replot it with no other distractions. Notice that the "type=n" option to [plot](#) function causes the axes to be plotted but not the points.

```
plot(x,y,xlab="log time",ylab="log flux", type="n")
lines(npmodel1, col=4, lwd=2)
```

This appears to be a piecewise linear curve. An analysis that assumes a piecewise linear curve will be carried out on these data later in the week.

In the case of non-polynomial (but still parametric) curve-fitting, we can use [nls](#). If we replace the response y by the original (nonlogged) flux values, we might posit a parametric model of the form $\text{flux} = \exp(a+b*x)$, where $x=\log(\text{time})$ as before. Compare a nonlinear approach (in red) with a nonparametric approach (in green) for this case:

```

flux <- grb[,2]
nlsmodel1 <- nls(flux ~ exp(a+b*x), start=list(a=0,b=0))
npmodel2 <- lowess(flux~x)
plot(x, flux, xlab="log time", ylab="flux")
lines(xx, exp(9.4602-.9674*xx), col=2, lwd=2)
lines(npmodel2, col=3, lwd=2)

```

Interestingly, the coefficients of the nonlinear least squares fit are different than the coefficients of the original linear model fit on the logged data, even though these coefficients have exactly the same interpretation: If $\text{flux} = \exp(a + b*x)$, then shouldn't $\log(\text{flux}) = a + b*x$? The difference arises because these two fitting methods calculate (and subsequently minimize) the residuals on different scales. Try plotting $\exp(a + b*xx)$ on the scatterplot of x vs. flux for both (a,b) solutions to see what happens. Next, try plotting $a + b*xx$ on the scatterplot of x vs. $\log(\text{flux})$ to see what happens.

If outliers appear to have too large an influence over the least-squares solution, we can also try resistant regression, using the [lqs](#) function in the MASS package. The basic idea behind lqs is that the largest residuals (presumably corresponding to "bad" outliers) are ignored. The results for our $\log(\text{flux})$ vs. $\log(\text{time})$ example look terrible but are very revealing. Can you understand why the output from lqs looks so very different from the least-squares output?

```

library(MASS)
lqsmodel1 <- lqs(y~x, method="lts")
plot(x,y,xlab="log time",ylab="log flux")
abline(model1,col=2)
abline(lqsmodel1,col=3)

```

Finally, let's consider least absolute deviation regression, which may be considered a milder form of resistant regression than [lqs](#). In least absolute deviation regression, even large residuals have an influence on the regression line (unlike in [lqs](#)), but this influence is less than in least squares regression. To implement it, we'll use a function called rq (regression quantiles) in the "quantreg" package. Like the "car" package, this package is not part of the standard distribution of R, so we'll need to download it. Also, we will need to install the "SparseM" package as this is required for the "quantreg" package. In order to do this, we must tell R where to store the installed library using the [install.packages](#) function.

```

install.packages(c("SparseM","quantreg"),lib="V:/") # lib=... is not always necessary!
library(quantreg, lib.loc="V:/")

```

Assuming the quantreg package is loaded, we may now compare the least-squares fit (red) with the least absolute deviations fit (green). In this example, the two fits are nearly identical:

```

rqmodel1 <- rq(y~x)
plot(x,y,xlab="log time",ylab="log flux")
abline(model1,col=2)
abline(rqmodel1,col=3)

```

Summer School in Statistics for
Astronomers & Physicists, IV
June 9-14, 2008

Laws of Probability, Bayes' theorem, and
the Central Limit Theorem

June 10, 8:45 - 10:15 am

Mosuk Chow

Department of Statistics
Penn State University

- p. 1/4

Deterministic experiment: One whose outcome is determined entirely by the conditions under which the experiment is performed.

Mathematical models for familiar deterministic experiments:

$s = ut + \frac{1}{2}at^2$: The distance traveled in time t by an object with initial velocity u and constant acceleration a :

Kepler's laws for planetary motion

$F = ma$: Newton's Second Law,

$V = kT$: Charles' gas law for V , the volume of a gas sample with T the gas temperature on the absolute scale

- p. 2/4

A *random experiment* is one which

- (a) can be repeated indefinitely under essentially unchanged conditions, and
- (b) whose outcome cannot be predicted with *complete certainty*, although all its possible outcomes can be described completely.

Toss a coin four times and observe the number of heads obtained.

Draw four cards from a fair deck and count the number of queens obtained.

The number of particles emitted by a radioactive substance in a one-second time period.

The annual number of radio flares observed on the Algol-type systems β Persei and δ Librae.

– p. 3/4

Sample space, S : The set of all possible outcomes of our random experiment.

Event: A subset of the sample space.

If A and B are events then:

$A \cup B$ is the event which occurs iff A or B occur.

$A \cap B$ is the event which occurs iff A and B occur.

\bar{A} is the event which occurs iff A does not occur.

S : The event which always occurs

\emptyset : The event which never occurs

Mutually exclusive events: $A \cap B = \emptyset$

– p. 4/4

The Axioms of Probability

Let \mathcal{E} be a random experiment with sample space S . With each subset A of S , we associate a number $P(A)$, the probability of A , such that:

- (1) $0 \leq P(A) \leq 1$
- (2) $P(S) = 1$
- (3) If A, B are mutually exclusive then

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

- (4) If $A_1, A_2, \dots, A_k, \dots$ are pairwise mutually exclusive events then

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i).$$

- p. 5/4

Some simple properties of probability:

$$P(\emptyset) = 0$$

$$P(\bar{A}) = 1 - P(A)$$

If $A \subseteq B$ then $P(A) \leq P(B)$

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

The Inclusion-Exclusion Formula:

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

- p. 6/4

Finite sample space: S consists of a finite number of “elementary” outcomes, $S = \{a_1, \dots, a_m\}$

Equally likely outcomes: Each elementary outcome has the same probability of occurring

$$P(A) = \frac{\text{No. of ways in which } A \text{ can occur}}{\text{No. of possible outcomes}}$$

Toss a fair coin twice.

$$P(\text{Two heads}) = \frac{1}{4}$$

$$P(\text{At least one head}) = 1 - P(\text{Two tails}) = \frac{3}{4}$$

Reminder:

$${n \choose k} = \frac{n!}{k!(n-k)!}$$

is the number of ways in which k objects can be chosen from a set of n objects.

- p. 7/4

Conditional Probability

A and B : subsets of S , the sample space

$P(A|B)$: “The probability of A given B ”

The *conditional probability* of A , given that B has already occurred, is defined as

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \quad P(B) \neq 0$$

$P(A|B)$ is undefined if $P(B) = 0$

Usually, $P(A|B) \neq P(A)$ because of the additional information that B has already occurred.

Choose a day at random

$A = \{\text{It will snow on that day}\}$

$B = \{\text{That day's low temperature is } 80^{\circ}\text{F}\}$

$P(A) \neq 0$, however $P(A|B) = 0$

- p. 8/4

Example: An urn contains 20 blue and 80 red balls. Mary and John each choose a ball at random from the urn. Let $B = \{\text{Mary chooses a blue ball}\}$, $P(B) = \frac{20}{100}$
 $A = \{\text{John chooses a blue ball}\}$, $P(A|B) = \frac{19}{99}$
The additional information that B has *already* occurred decreases the probability of A

$$\begin{aligned} P(A \cap B) &= \frac{\text{No. of ways of choosing 2 blue balls}}{\text{No. of ways of choosing 2 balls}} \\ &= \frac{\binom{20}{2}}{\binom{100}{2}} = \frac{19}{495} \end{aligned}$$

Note that

$$\frac{P(A \cap B)}{P(B)} = \frac{19}{495} \div \frac{20}{100} = \frac{19}{99} = P(A|B)$$

- p. 9/4

$P(A|B)$ satisfies the axioms of probability: If $P(B) \neq 0$ then

- (1) $0 \leq P(A|B) \leq 1$
- (2) $P(S|B) = 1$
- (3) If A_1 and A_2 are mutually exclusive then

$$P(A_1 \cup A_2|B) = P(A_1|B) + P(A_2|B)$$

- (4) If $A_1, A_2, \dots, A_k, \dots$ are pairwise mutually exclusive events then

$$P\left(\bigcup_{i=1}^{\infty} A_i|B\right) = \sum_{i=1}^{\infty} P(A_i|B).$$

Also,

$$P(\emptyset|B) = 0$$

$$P(\bar{A}|B) = 1 - P(A|B)$$

If $A_1 \subseteq A_2$ then $P(A_1|B) \leq P(A_2|B)$

$$P(A_1 \cup A_2|B)$$

$$= P(A_1|B) + P(A_2|B) - P(A_1 \cap A_2|B)$$

- p. 11/4

The Multiplication Theorem: If $P(B) \neq 0$ then

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

Proof:

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

Repeat the argument: If $P(A_1 \cap A_2 \cap A_3) \neq 0$ then

$$P(A_1 \cap A_2 \cap A_3) = P(A_1)P(A_2|A_1)P(A_3|A_1 \cap A_2)$$

- p. 12/4

Events B_1, \dots, B_k form a *partition* of the sample space S if

- (1) $B_i \cap B_j = \emptyset$ for all $i \neq j$ (pairwise mutually exclusive)
- (2) $B_1 \cup B_2 \cup \dots \cup B_k = S$, the full sample space
- (3) $P(B_i) \neq 0$ for all $i = 1, \dots, k$

The Law of Total Probability: Let A be any event and B_1, \dots, B_k be a partition. Then

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

By means of a simple Venn diagram,

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

By the Multiplication Theorem,

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

- p. 13/4

Example: An urn contains 20 blue and 80 red balls. Two balls are chosen at random and without replacement.

Calculate $P(A)$ where

$$A = \{\text{The second ball chosen is blue}\}$$

Let $B = \{\text{The first ball chosen is blue}\}$

Then $\bar{B} = \{\text{The first ball chosen is red}\}$

B, \bar{B} are a *partition* of the sample space

$$B \cap \bar{B} = \emptyset, B \cup \bar{B} = S, P(B) = \frac{20}{100}, P(\bar{B}) = \frac{80}{100}$$

By the Law of Total Probability,

$$\begin{aligned} P(A) &= P(A|B) \cdot P(B) + P(A|\bar{B}) \cdot P(\bar{B}) \\ &= \frac{19}{99} \cdot \frac{20}{100} + \frac{20}{99} \cdot \frac{80}{100} = \frac{20}{100} \end{aligned}$$

Notice that $P(A) = P(B) = \frac{20}{100}$

- p. 14/4

Cabinets I and II each have 2 drawers. I contains 2 silver coins in each drawer. II contains 2 silver coins in one drawer and 1 gold coin in the other drawer. Chris rolls a fair die and chooses I if the die rolls 1, 2, 3, or 4; otherwise, he chooses II . Having chosen a cabinet, he chooses a drawer at random and opens it.

Find the probability that the opened drawer contains 2 silver coins:

$$A = \{\text{The opened drawer contains 2 silver coins}\}$$

$$B = \{\text{Chris chooses cabinet } I\}$$

$$\bar{B} = \{\text{Chris chooses cabinet } II\}$$

$\{B, \bar{B}\}$ form a partition of the sample space

$$B \cap \bar{B} = \emptyset, B \cup \bar{B} = S, P(B) = \frac{4}{6}, P(\bar{B}) = \frac{2}{6}$$

- p. 15/4

By the Law of Total Probability,

$$\begin{aligned} P(A) &= P(A|B) \cdot P(B) + P(A|\bar{B}) \cdot P(\bar{B}) \\ &= 1 \cdot \frac{4}{6} + \frac{1}{2} \cdot \frac{2}{6} \\ &= \frac{5}{6} \end{aligned}$$

A more difficult problem

Given that the opened drawer contains 2 silver coins, find $P(\bar{B}|A)$, the probability that cabinet II was chosen.

- p. 16/4

1. B_1, \dots, B_k is a partition, and
2. A is an event with $P(A) \neq 0$.

Then

$$P(B_1|A) = \frac{P(A|B_1)P(B_1)}{P(A|B_1)P(B_1) + \dots + P(A|B_k)P(B_k)}$$

1. Definition of conditional probability:

$$P(B_1|A) = \frac{P(B_1 \cap A)}{P(A)} = \frac{P(A \cap B_1)}{P(A)}$$

2. The multiplication theorem:

$$P(A \cap B_1) = P(A|B_1)P(B_1)$$

3. The Law of Total Probability:

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

Return to the drawer problem

$$P(A|B) = 1, P(B) = \frac{4}{6}$$

$$P(A|\bar{B}) = \frac{1}{2}, P(\bar{B}) = \frac{2}{6}$$

By Bayes' theorem,

$$\begin{aligned} P(\bar{B}|A) &= \frac{P(A|\bar{B}) \cdot P(\bar{B})}{P(A|B) \cdot P(B) + P(A|\bar{B}) \cdot P(\bar{B})} \\ &= \frac{\frac{1}{2} \cdot \frac{2}{6}}{1 \cdot \frac{4}{6} + \frac{1}{2} \cdot \frac{2}{6}} \\ &= \frac{1}{5} \end{aligned}$$

Homework Assignment

Joseph Bertrand, *Calcul des Probabilités*, 1889

Bertrand's Box Paradox: A box contains three drawers. In one drawer there are two gold coins; in another drawer there are two silver coins; and in the third drawer there is one silver coin and one gold coin. A drawer is selected at random, and then a coin is selected at random from that drawer. Given that the selected coin is gold, what is the probability that the other coin is gold?

Three suspected burglars, Curly, Larry, and Moe, are held incommunicado in a distant country by a ruthless jailer. The three are told that one of them has been chosen at random to be imprisoned and that the other two will be freed, but they are not told whom. Curly asks the jailer who will be imprisoned, but the jailer declines to answer. Instead the jailer tells him that Larry will be freed. Given that Larry is to be freed, what is the probability that Curly will be imprisoned?

Hint: Search the Internet.

— p. 19/4

Independence: Events A and B are *independent* if

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

Motivation: $P(A|B) = P(A)$, $P(B|A) = P(B)$

Theorem: If $\{A, B\}$ are independent then so are $\{A, \bar{B}\}$, $\{\bar{A}, B\}$, and $\{\bar{A}, \bar{B}\}$

Example: Roll two fair dice. Let

$A = \{\text{The first die shows an even number}\}$

$B = \{\text{The second die shows an odd number}\}$

$C = \{\text{The total rolled is an even number}\}$

$\{A, B\}$, $\{B, C\}$, $\{A, C\}$ are independent pairs

$$P(A \cap B \cap C) = 0; P(A)P(B)P(C) = \frac{1}{8}$$

— p. 20/4

Events A, B, C are *mutually independent* if:

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

$$P(A \cap C) = P(A)P(C),$$

$$P(B \cap C) = P(B)P(C),$$

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

Pairwise independence does not imply mutual independence

- p. 21/4

Random variable: A numerical value calculated from the outcome of a random experiment

A function from S , the sample space, to \mathbb{R} , the real line

Discrete random variable: One whose possible values are a discrete set

The number of collisions daily between the ISS and orbital debris

Continuous random variable: One whose values form an interval
The length of time a “shooting star” is visible in the sky

- p. 22/4

Let X be the number of heads obtained among 2 tosses of a (fair) coin

The *possible values* of X are: 0, 1, 2

The *probability distribution* of X is a listing of its possible values and the corresponding probabilities

Toss a coin twice:

$$P(X = 0) = \frac{1}{4}, P(X = 1) = \frac{1}{2}, P(X = 2) = \frac{1}{4}$$

Notice that $\sum_{x=0}^2 P(X = x) = 1$

Billingsley, Probability and Measure: The infinite coin toss

- p. 23/4

Bernoulli trial: A random experiment with only two possible outcomes, “success” or “failure”

$$p \equiv P(\text{success}), q \equiv 1 - p = P(\text{failure})$$

Perform n independent repetitions of a Bernoulli trial

X : No. of successes among all n repetitions

Possible values of X : 0, 1, 2, ..., n

Probability distribution of X :

$$P(X = k) = \binom{n}{k} p^k q^{n-k}, \quad k = 0, 1, 2, \dots, n$$

The binomial distribution: $X \sim B(n, p)$

Every probability distribution satisfies:

$$1. 0 \leq P(X = x) \leq 1$$

$$2. \sum_{x=0}^n P(X = x) = 1$$

- p. 24/4

A. Mészáros, “On the role of Bernoulli distribution in cosmology,” *Astron. Astrophys.*, 328, 1-4 (1997).

n uniformly distributed points in a region of volume $V = 1$ unit

X : No. of points in a fixed region of volume p

X has a binomial distribution, $X \sim B(n, p)$

– p. 25/4

M. L. Fudge, T. D. Maclay, “Poisson validity for orbital debris ...” *Proc. SPIE*, 3116 (1997) 202-209, Small Spacecraft, Space Environments, and Instrumentation Technologies ISS ... at risk from orbital debris and micrometeorite impact fundamental assumption underlying risk modeling: orbital collision problem can be modeled using a Poisson distribution

assumption found to be appropriate based upon the Poisson ... as an approximation for the binomial distribution and ... that is it proper to physically model exposure to the orbital debris flux environment using the binomial.

– p. 26/4

The geometric distribution:

Perform independent repetitions of a Bernoulli trial

X : No. of trials needed to obtain one success

Possible values of X : 1, 2, 3, ...

$$P(X = k) = q^{k-1}p, \quad k = 1, 2, 3, \dots$$

$$\text{Geometric series: } \sum_{k=1}^{\infty} q^{k-1} = \frac{1}{1-q} = \frac{1}{p}$$

- p. 27/4

The negative binomial distribution:

Perform independent repetitions of a Bernoulli trial

X : No. of trials needed to obtain r successes

Possible values of X : $r, r + 1, r + 2, \dots$

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

Neyman, Scott, and Shane (1953, ApJ): Counts of galaxies in clusters

ν : The number of galaxies in a randomly chosen cluster

Basic assumption: ν follows a negative binomial distribution

- p. 28/4

$$P(Y = k) = \frac{e^{-\lambda} \lambda^k}{k!}, \quad k = 0, 1, 2, \dots$$

X : binomial random variable

$$P(X = k) = \binom{n}{k} p^k q^{n-k}, \quad k = 0, 1, 2, \dots, n$$

If $n \rightarrow \infty$, $p \rightarrow 0$, and $np \rightarrow \lambda$ then

$$\lim P(X = k) = \frac{e^{-\lambda} \lambda^k}{k!}, \quad k = 0, 1, 2, \dots$$

Example: X is the number of typographical errors on a newspaper page. X has a binomial distribution, $X \sim B(n, p)$. When n is very large, p is very small, the Poisson distribution is a good approximation to the distribution of X

- p. 29/4

Expected value of X : Average value of X based on many repetitions of the experiment,

$$\mu \equiv E(X) = \sum_{\text{all } k} k \cdot P(X = k)$$

If $X \sim B(n, p)$ then $E(X) = np$

If X is geometric then $E(X) = 1/p$

Standard deviation: A measure of average fluctuation of X around μ

Variance of X :

$$\text{Var}(X) = E(X - \mu)^2 = \sum_{\text{all } k} (k - \mu)^2 \cdot P(X = k)$$

$$S.D.(X) = \sqrt{\text{Var}(X)}$$

If $X \sim B(n, p)$ then $\text{Var}(X) = npq$

- p. 30/4

Every nice continuous random variable X has a probability density function f .

The three important properties of f :

$$f(x) \geq 0 \text{ for all } x$$

$$\int_{-\infty}^{\infty} f(x) dx = 1$$

$$P(X \leq t) = \int_{-\infty}^t f(x) dx \text{ for all } t$$

Similar to discrete random variables,

$$\mu \equiv E(X) = \int_{-\infty}^{\infty} x f(x) dx$$

$$\text{Var}(X) = E(X - \mu)^2 = \int_{-\infty}^{\infty} (x - \mu)^2 f(x) dx$$

- p. 31/4

The function

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, \quad -\infty < x < \infty$$

is a probability density function:

1. $\phi(x) \geq 0$ for all x ;
2. $\int_{-\infty}^{\infty} \phi(x) dx = 1$

Standard normal distribution: A continuous random variable Z having probability density function $\phi(x)$, i.e., for all t ,

$$P(Z \leq t) = \int_{-\infty}^t \phi(x) dx$$

Homework: Show that $E(Z) = 0$, $\text{Var}(Z) = 1$

- p. 32/4

Normal Approximation to the Binomial Distn.:

DeMoivre (1733), $p = 1/2$

Laplace (1812), general p

If $X \sim B(n, p)$ then

$$\lim_{n \rightarrow \infty} P\left(\frac{X - np}{\sqrt{npq}} \leq t\right) = P(Z \leq t)$$

Proof: 1. Compute the Fourier transform of the density function of $(X - np)/\sqrt{npq}$

2. Find the limit of this F.t. as $n \rightarrow \infty$

3. Check that the limiting F.t. is the F.t. of the density function of Z

Intuition: If $X \sim B(n, p)$ and n is large then

$$\frac{X - np}{\sqrt{npq}} \approx Z$$

- p. 33/4

Micrometeorite: piece of rock less than 1 mm in diameter (similar to a fine grain of sand); moving at up to 80 km/sec. (48 miles/sec.); difficult to detect; no practical way to avoid them

Claim: "The probability that the ISS will receive a micrometeorite impact during its (30-year) mission is nearly 100%"

n : No. of days in 30 years, $30 \times 365 = 10,950$

p : Probability that a micrometeorite hits the ISS on a randomly chosen day

X : No. of micrometeorite impacts on the ISS over its designed lifetime of 30 years

$X \sim B(n, p)$: a model for the distribution of the no. of impacts

- p. 34/4

Calculate $P(X \geq 1)$

Poisson approximation: n is large, p is small

$$X \approx \text{Poisson}(\lambda), \lambda = np$$

$$P(X \geq 1) = 1 - P(X = 0) \simeq 1 - e^{-\lambda}$$

$$n = 10,950, p = 10^{-2}, \lambda = 109.5$$

Normal approximation:

$$\begin{aligned} P(X \geq 1) &= P\left(\frac{X - np}{\sqrt{npq}} \geq \frac{1 - np}{\sqrt{npq}}\right) \\ &\simeq P\left(Z \geq \frac{1 - 109.5}{\sqrt{108.405}}\right) \\ &= P(Z \geq -10.42) \\ &= 1 \end{aligned}$$

- p. 35/4

Important continuous random variables

Normal, chi-square, t -, and F -distributions

Z : standard normal distribution

Z^2 : χ_1^2 (chi-square with 1 degree of freedom)

t -distribution: $\frac{N(0,1)}{\sqrt{\chi_p^2/p}}$

F -distribution: $\frac{\chi_p^2/p}{\chi_q^2/q}$

- p. 36/4

Las Vegas or Monte Carlo: $n = 10^6$ people each play a game (craps, trente-et-quarante, roulette, ...)

Each of the 10^6 repetitions is a Bernoulli trial

Casinos fear of too many (or too few) winners

X : the number of winners among all n people

p : probability of success on each repetition

Probability distribution: $X \sim B(n, p)$

Markov's Inequality: For any random variable X and $t > 0$,

$$P(X \geq t) \leq \frac{E(X)}{t}$$

We can compute the maximum probability of having too many winners

– p. 37/4

Craps game: $p = .492929292\dots = \frac{244}{495}$

$E(X) = np = 492,929.29$

Markov's inequality: $P(X \geq t) \leq \frac{492,929.29}{t}$

$t = 500,000$: $P(X \geq 500,000) \leq 0.98585\dots$

Chebyshev's Inequality: For any $t > 0$,

$$P\left(\frac{|X - \mu|}{S.D.(X)} < t\right) \geq 1 - \frac{1}{t^2}$$

Craps game: $S.D.(X) = \sqrt{npq} = 499.95$

$$P\left(\frac{|X - 492,929.29|}{499.95} < t\right) \geq 1 - \frac{1}{t^2}$$

$t = 8$: $P(488929 < X < 496929) \geq .984375$

– p. 38/4

The (weak) Law of Large Numbers

X_1, X_2, \dots : independent, identically distributed random variables

μ : The mean of each X_i

$\bar{X} = \frac{1}{n}(X_1 + \dots + X_n)$: The sample mean of the first n observations

For $t > 0$,

$$P(|\bar{X} - \mu| \geq t) \rightarrow 0 \quad \text{as } n \rightarrow \infty$$

- p. 39/4

X_1, \dots, X_n : independent, identically distributed random variables

μ : The mean (expected value) of each X_i

σ^2 : The variance of each X_i

$\bar{X} = \frac{1}{n}(X_1 + \dots + X_n)$: The average of the X_i 's

The Central Limit Theorem: If n is large then

$$\frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \approx Z$$

where $Z \sim N(0, 1)$.

Proof: 1. Compute the Fourier transform of the density function of $(X_1 + \dots + X_n - n\mu)/\sigma\sqrt{n}$

2. Find the limit of this F.t. as $n \rightarrow \infty$

3. Check that the limiting F.t. is the F.t. of the density function of Z

- p. 40/4

Historical note

The Central Limit Theorem was first stated and proved by Laplace (Pierre Simon, the Marquis de Laplace).

Francis Galton, *Natural Inheritance*, 1889

"I know of scarcely anything so apt as to impress the imagination as the wonderful form of cosmic order expressed by the 'Law of Frequency of Error.' The Law would have been personified by the Greeks and deified, if they had known of it. It reigns with serenity and in complete self-effacement amidst the wildest confusion. The huger the mob and the greater the apparent anarchy, the more perfect is its sway. It is the supreme law of unreason."

Website to demonstrate Central Limit Theorem:
http://www.ruf.rice.edu/~lane/stat_sim/sampling_dist

- p. 41/4

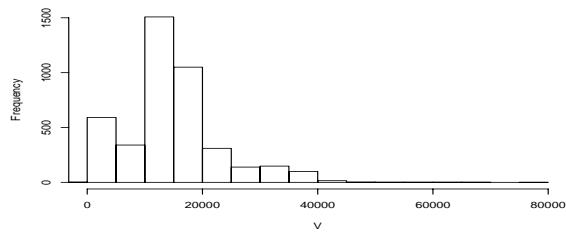
Next, we will show the histogram of V: Velocities of galaxies in the direction of the Shapley superclusters. The dataset is described at

http://astrostatistics.psu.edu/datasets/Shapley_galaxy.html
(Notice that the data appear to be non-Gaussian)

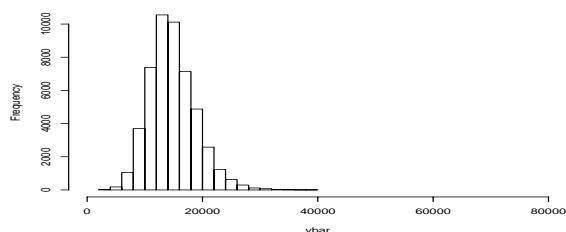
In this example, we treat this data set as the population and hence we can compute the exact values of the population mean and population standard deviation. We will then examine the behavior of \bar{V} base on samples of sizes $n = 4$ and 36 selected randomly with replacement from the population. The sampling distributions based on 50000 simulations are given below the histogram of the population.

- p. 42/4

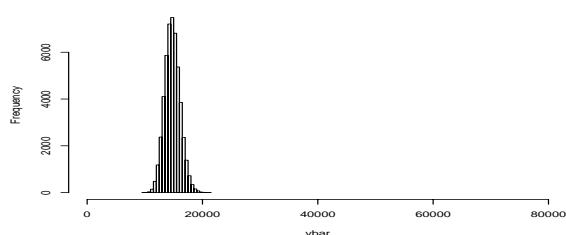
Histogram of velocities for 4215 galaxies, mean = 14789.24, st.dev. = 8043.123



Sampling distribution for vbar, n=4, mean = 14782.83, st.dev. = 4012.514



Sampling distribution for vbar, n=36, mean = 14785.41, st.dev. = 1335.290



Summer School in Statistics for
Astronomers IV
June 10, 2008

Estimation, Confidence Intervals,
and Tests of Hypotheses

James L Rosenberger

(Notes from Donald Richards and William Harkness)
Department of Statistics
Center for Astrostatistics
Penn State University

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

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

$M_0 \equiv \mu$: Mean visual absolute magnitude

σ : Std. deviation of visual absolute magnitude

Magnitudes are log variables (a log-normal distribution)

Statistical Problems:

1. On the basis of collected data, estimate the *parameters* μ and σ . Also, derive a plausible range of values for each parameter; etc.
2. V-d-B, etc., conclude that the LF is “adequately described” by a normal distribution. How can we quantify the plausibility of their conclusion?

Here is a diagram from van den Bergh (1985), providing complete data for the Milky Way (notice that the data *appear* to be non-Gaussian)

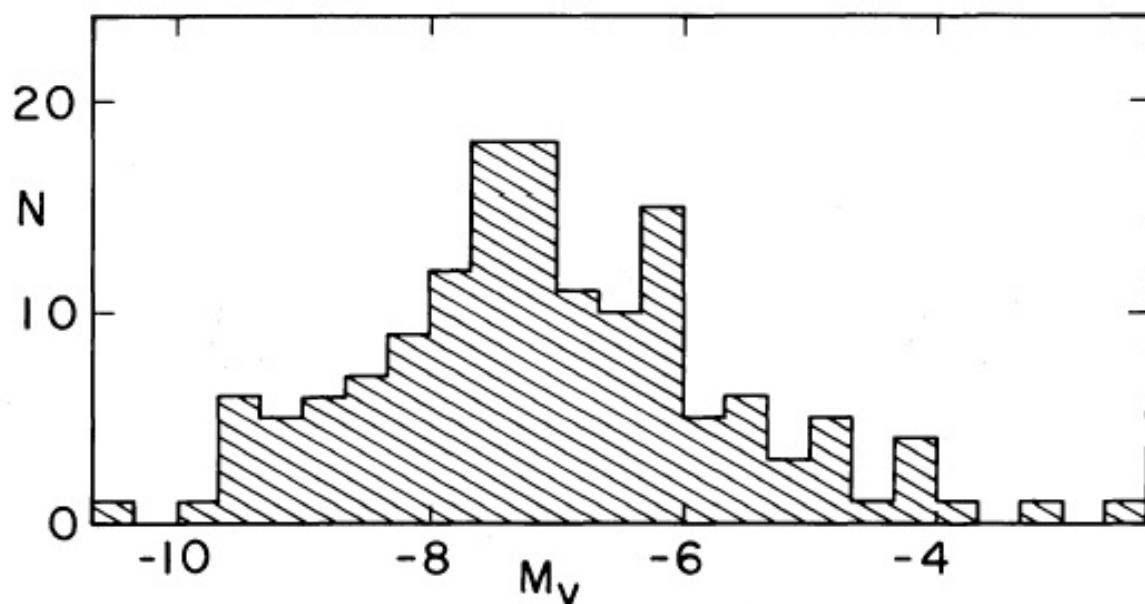


FIG. 2.—Luminosity function of Galactic globular clusters (one object at $M_V = -1.7$ is not plotted). Note that the luminosity function is asymmetrical with a long tail extending to faint magnitudes.

A second diagram from van den Bergh (1985); truncated dataset for M31 (Andromeda galaxy)

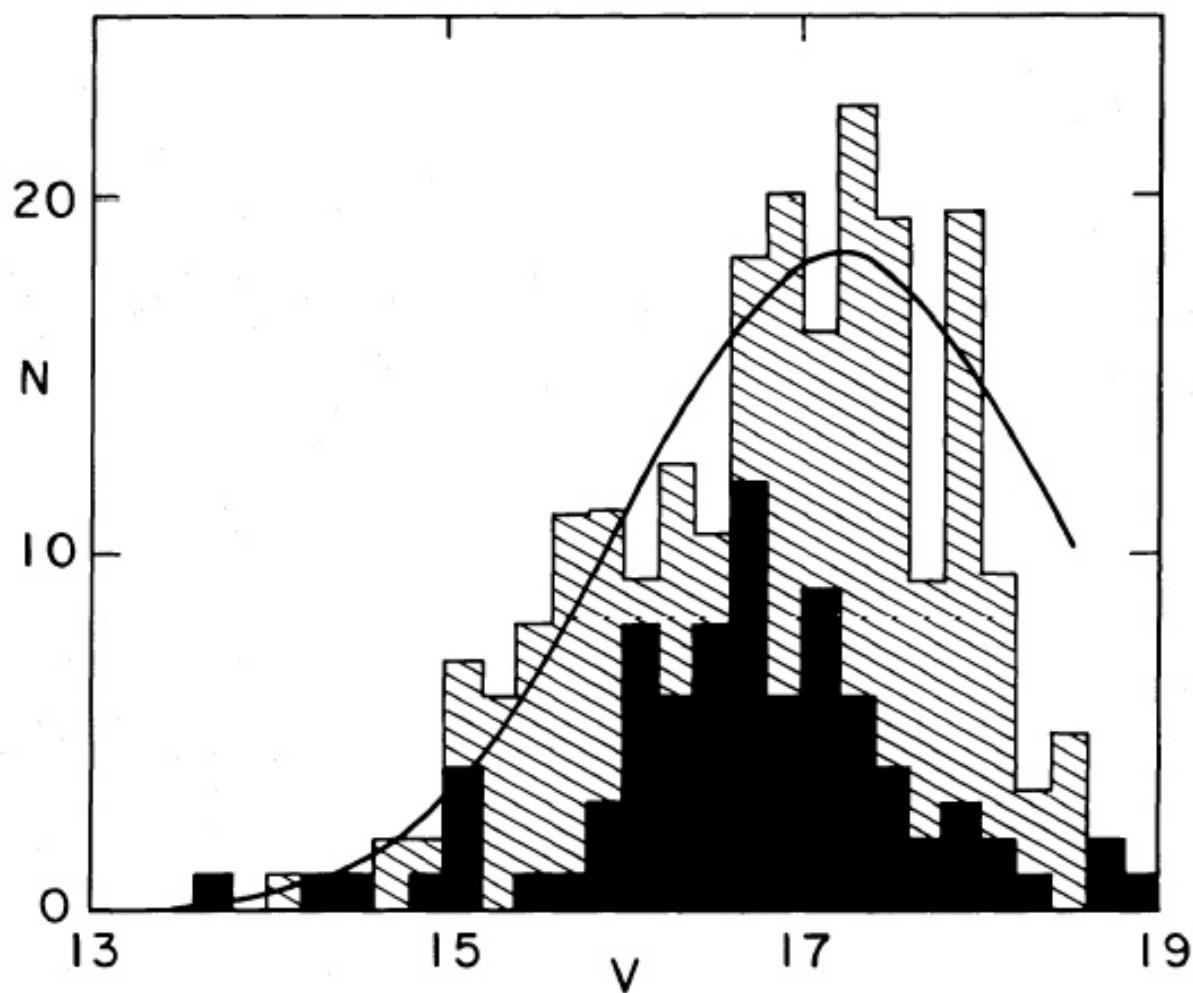


FIG. 1.—Luminosity function of clusters with $0.70 \leq B - V < 1.00$ in M31. Smooth curve is a Gaussian with $V(\max) = 17.2$ and $\sigma = 1.2$ mag. Lower histogram shows the luminosity function of the halo of M31 derived by Racine and Shara (1979).

X : A random variable

Population: The collection of all values of X

$f(x)$: The prob. density function (p.d.f.) of X

Statistical model: A choice of p.d.f. for X

We choose a model which “adequately describes” data collected on X

Parameter: A number which describes a property of the population

μ and σ are parameters for the p.d.f. of the LF for Galactic globulars

Values of the chosen p.d.f. depend on X and on the parameters: $f(x; \mu, \sigma)$

Parameter space: The set of permissible values of the parameters

$$\Omega = \{(\mu, \sigma) : -\infty < \mu < \infty, \sigma > 0\}$$

Random sample

In practice: Data values x_1, \dots, x_n which are *fully representative* of the population

In theory: Mutually independent random variables X_1, \dots, X_n which all have the same distribution as X

Parameter: A number computable only from the entire population

Statistic: A number computed from the random sample X_1, \dots, X_n

Sample mean: $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$

Sample variance: $S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$

In general, a statistic is a function $Y = U(X_1, \dots, X_n)$ of the observations.

Sampling distribution: The probability distribution of a statistic

X : LF for globular clusters

Model: $N(\mu, \sigma^2)$, the normal distribution with mean μ and variance σ^2

Problem: Given a random sample x_1, \dots, x_n , estimate μ

\bar{x} is a very good estimate of μ

\hat{x} , the sample median, is a good plausible estimate of μ

$x_{(n)}$, the largest observed value in the LF, is obviously a poor estimate of μ , since it almost certainly is much larger than μ .

Statistics, like the sample mean \bar{x} and the sample median \hat{x} are called *point estimators* of μ

Roman letters are used to denote Data and Greek letters to denote parameters.

Let θ be a ‘generic’ parameter (for example, μ or σ)

$Y = u(X_1, \dots, X_n)$, a function of the data;

Y is

- (i) a point estimator of θ ,
- (ii) a random variable and so
- (iii) has a probability distribution called *the sampling distribution of the statistic Y* .

Conceptually, we can calculate the moments of Y , including the mean $E(Y)$.

If $E(Y) = \theta$, the Y is said to be an unbiased estimator of θ , for example \bar{x} is an unbiased estimator of the population mean μ .

Intuitively, Y is unbiased if its long-term average value is equal to θ

Example: LF for globular clusters

The sample mean, \bar{X} , is unbiased:

$$E(\bar{X}) = \frac{1}{n} \sum_{i=1}^n E(X_i) = \frac{1}{n} \sum_{i=1}^n \mu = \mu$$

Similarly, if the LF has a normal distribution then \hat{x} , the sample median, is an unbiased estimate of μ also.

$X_{(n)}$: the largest observed LF

$X_{(n)}$ is not unbiased: $E(X_{(n)}) > \mu$

We also want statistics which are “close” to θ

For all statistics Y , calculate $E[(Y - \theta)^2]$, the mean square error (MSE)

Choose as our point estimator the statistic for which the MSE is smallest

A statistic Y which minimizes $E[(Y - \theta)^2]$ is said to have *minimum mean square error*

If Y is also unbiased then $MSE = \text{Var}(Y)$, and Y is a *minimum variance unbiased estimator* ($MVUE$)

Reminder: If R_1, R_2 are random variables and a, b are constants then

$$E(aR_1 + bR_2) = aE(R_1) + bE(R_2).$$

If R_1 and R_2 are also independent then

$$\text{Var}(aR_1 + bR_2) = a^2 \text{Var}(R_1) + b^2 \text{Var}(R_2).$$

Example: LF for globular clusters

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

Random sample of size $n = 3$: X_1, X_2, X_3

Two point estimators of μ :

Sample mean: $\bar{X} = \frac{1}{3}(X_1 + X_2 + X_3)$

Place more weight on the last observation

A weighted average: $Y = \frac{1}{6}(X_1 + 2X_2 + 3X_3)$

Both estimators are unbiased: $E(\bar{X}) = \mu$, and

$$\begin{aligned} E(Y) &= \frac{1}{6}E(X_1 + 2X_2 + 3X_3) \\ &= \frac{1}{6}(\mu + 2\mu + 3\mu) = \mu \end{aligned}$$

However,

$$\text{Var}(\bar{X}) = \frac{1}{3^2}(\sigma^2 + \sigma^2 + \sigma^2) = \frac{1}{3}\sigma^2,$$

while

$$\begin{aligned}\text{Var}(Y) &= \frac{1}{6^2}\text{Var}(X_1 + 2X_2 + 3X_3) \\ &= \frac{1}{36}(\sigma^2 + 2^2\sigma^2 + 3^2\sigma^2) = \frac{7}{18}\sigma^2\end{aligned}$$

\bar{X} and Y are unbiased but $\text{Var}(\bar{X}) < \text{Var}(Y)$

The distribution of \bar{X} is more concentrated around μ than the distribution of Y

\bar{X} is a better estimator than Y

Note: For any sample size n , $\text{Var}(\bar{X}) = \frac{\sigma^2}{n}$

Random sample: X_1, \dots, X_n

$Y = u(X_1, \dots, X_n)$: An estimator of θ

Bear in mind that Y depends on n

It would be good if Y “converges” to θ as $n \rightarrow \infty$

Y is *consistent* if, for any $t > 0$,

$$P(|Y - \theta| \geq t) \rightarrow 0$$

as $n \rightarrow \infty$

The Law of Large Numbers: If X_1, \dots, X_n is a random sample from X then for any $t > 0$,

$$P(|\bar{X} - \mu| \geq t) \rightarrow 0$$

as $n \rightarrow \infty$

Very Important Conclusion: For any population, \bar{X} is a consistent estimator of μ .

How do we construct good estimators?

Judicious guessing

The method of maximum likelihood

The method of moments

Bayesian methods

Decision-theoretic methods

Unbiased estimator

Consistent estimator

A consequence of Chebyshev's inequality: If Y is an unbiased estimator of θ and $\text{Var}(Y) \rightarrow 0$ as $n \rightarrow \infty$ then Y is consistent.

The Method of Moments

X : Random variable with p.d.f. $f(x; \theta_1, \theta_2)$

Parameters to be estimated: θ_1, θ_2

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 θ_1 and θ_2

3. Solve for θ_1, θ_2 the simultaneous equations

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

The solutions are the *method-of-moments estimators* of θ_1, θ_2

Example: LF for globular clusters

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

Random sample: X_1, \dots, X_n

1. The first two sample moments:

$$m_1 = \frac{1}{n} \sum_{i=1}^n X_i = \bar{X}$$

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

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

Hanes-Whittaker (1987), “Globular clusters as extragalactic distance indicators ...,” AJ 94, p. 906

M_l : The absolute magnitude limit of the study

T : A parameter identifying the size of a cluster

Truncated normal distribution:

$$f(x; \mu, \sigma^2, T) \propto \begin{cases} \frac{T}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], & x \leq M_l \\ 0, & x > M_l \end{cases}$$

Method of moments: Calculate

1. The first three sample moments,

$$m_k = \frac{1}{n} \sum_{i=1}^n X_i^k, \quad k = 1, 2, 3$$

2. The first three population moments

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

3. Solve the equations $m_k = E(X^k)$, $k = 1, 2, 3$

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, 1978

Measured abundances compared with propagation calculations using distributions of path lengths; data suggest an exponential distribution truncated at short path lengths

Protheroe, et al. "Interpretation of cosmic ray composition - The path length distribution," ApJ., 247 1981

X : Length of paths

Parameters: $\theta_1, \theta_2 > 0$

Model:

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

LF for globular clusters in the Galaxy,

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

Random sample: X_1, \dots, X_n

\bar{X} is an unbiased estimator of μ

\bar{X} has minimum variance among all estimators which are linear combinations of X_1, \dots, X_n

S^2 is an unbiased estimator of σ^2

Given an actual data set, we calculate \bar{x} and s^2 to obtain point estimates of μ and σ^2

Point estimates are not perfect

We wish to quantify their accuracy

Confidence Intervals

LF for globular clusters in the Galaxy

X is $N(\mu, \sigma^2)$

Random sample: X_1, \dots, X_n

\bar{X} is an unbiased estimator of μ : $E(\bar{X}) = \mu$

What is the probability distribution of \bar{X} ?

Let Y be a linear combination of independent normal random variables. Then Y also has a normal distribution.

Conclusion: \bar{X} has a normal distribution

$E(\bar{X}) = \mu$, $\text{Var}(\bar{X}) = \frac{\sigma^2}{n}$, so $\bar{X} \sim N(\mu, \frac{\sigma^2}{n})$,

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

Consult the tables of the $N(0, 1)$ distribution:

$$P(-1.96 < Z < 1.96) = 0.95$$

For LF data,

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

Assume that σ is known, $\sigma = 1.2$ mag for Galactic globulars (van den Bergh , 1985)

Solve for μ the inequalities

$$-1.96 < \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} < 1.96$$

The solution is

$$\bar{X} - 1.96 \frac{\sigma}{\sqrt{n}} < \mu < \bar{X} + 1.96 \frac{\sigma}{\sqrt{n}}$$

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

The probability that the interval

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

“captures” μ is 0.95.

This interval is called a *95% confidence interval for μ*

It is a plausible range of values for μ together with a quantifiable measure of its plausibility

Notes:

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” ± 1.96 symmetrically around 0 to minimize the length of the confidence interval.

“Cutoff limits” are called “percentage points”

Example (devised from van den Bergh, 1985):

$n = 148$ Galactic globular clusters

$\bar{x} = -7.1$ mag

We assume that $\sigma = 1.2$ mag

M_0 : The population mean visual absolute magnitude

A 95% confidence interval for M_0 is

$$\begin{aligned} & \left(\bar{x} - 1.96 \frac{\sigma}{\sqrt{n}}, \bar{x} + 1.96 \frac{\sigma}{\sqrt{n}} \right) \\ &= \left(-7.1 - 1.96 \frac{1.2}{\sqrt{148}}, -7.1 + 1.96 \frac{1.2}{\sqrt{148}} \right) \\ &= (-7.1 \mp 0.193) \end{aligned}$$

This is a plausible range of values for M_0 .

The Warning: Don't bet your life that your 95% confidence interval has captured μ (but the odds are in your favor -19 to 1)

Intervals with higher levels of confidence, 90%, 98%, 99%, 99.9%, can be obtained similarly

Intervals with confidence levels $100(1 - \alpha)\%$ are obtained by replacing the multiplier 1.96 in a 95% confidence by $Z_{\alpha/2}$, where $Z_{\alpha/2}$ is determined by

$$P(-Z_{\alpha/2} < Z < Z_{\alpha/2}) = 1 - \alpha;$$

a 95% confidence has $\alpha = 0.05$.

90%, 98%, 99%, 99.9% confidence intervals correspond to $\alpha = .10, .02, .01$, and $.001$, respectively; the corresponding values of $Z_{\alpha/2}$ are 1.645, 2.33, 2.58, and 3.09, respectively.

If σ is unknown then the previous confidence intervals are not useful

A basic principle in statistics: Replace any unknown parameter with a good estimator

LF data problem; a random sample X_1, \dots, X_n drawn from $N(\mu, \sigma^2)$

We are tempted to construct confidence intervals for μ using the statistic $\frac{\bar{X} - \mu}{S/\sqrt{n}}$

What is 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 on $n-1$ degrees of freedom

We construct confidence intervals as before

Suppose that $n = 16$, then see the tables of the t -distribution on 15 degrees of freedom:

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

Therefore

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

Solve for μ in the inequalities

$$-2.131 < \frac{\bar{X} - \mu}{S/\sqrt{n}} < 2.131$$

A 95% confidence interval for μ is

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

Example: $n = 16$, $\bar{x} = -7.1$ mag, $s = 1.1$ mag.

A 95% confidence interval for μ is -7.1 ∓ 0.586

Normal population $N(\mu, \sigma^2)$

We want to obtain confidence intervals for σ

Random sample: X_1, \dots, X_n

S^2 is an unbiased and consistent estimator of σ^2

What is the sampling distribution of S^2 ?

The *chi-squared* (χ^2) distribution: $(n-1)S^2/\sigma^2$ has a chi-squared distribution on $n-1$ degrees of freedom.

We now construct confidence intervals as before

Consult the tables of the χ^2 distribution

Find the percentage points, and solve the various inequalities for σ^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 χ^2 distribution

Solve for σ^2 the inequalities: $a < \frac{(n-1)S^2}{\sigma^2} < b$

A 95% confidence interval for σ^2 is

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

Example: $n = 16$, $s = 1.2$ mag

Percentage points from the χ^2 tables (with 15 degrees of freedom): 6.262 and 27.49

Note: The percentage points are not symmetric about 0

A 95% confidence interval for σ^2 is

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

All other things remaining constant:

The greater the level of confidence, the longer the confidence interval

The larger the sample size, the shorter the confidence interval

How do we choose n ?

In our 95% confidence intervals for μ , 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 then we choose n so that

$$\frac{1.96\sigma}{\sqrt{n}} = 0.01$$

Solve this equation for n :

$$n = \left(\frac{1.96\sigma}{0.01} \right)^2$$

Confidence intervals with large sample sizes

Papers on LF for globular clusters

Sample sizes are large: 68, 148, 300, 1000, ...

A modified Central Limit Theorem

X_1, \dots, X_n : a random sample

μ : The population mean

\bar{X} and S : The sample mean and std. deviation

The modified CLT: If n is large then

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

The conclusion does not depend on the population probability distribution

The resulting confidence intervals for μ also do not depend on the population probability distribution

Tests of Hypotheses

Alternatives to confidence intervals

A LF researcher believes that $M_0 = -7.7$ mag for the M31 globular clusters. The researcher collects a *random sample* of data from M31

A natural question: “Are the data strongly in support of the claim that $M_0 = -7.7$ mag?”

Statistical hypothesis: A statement about the parameters of a population.

Statistical test of significance: A procedure for comparing observed data with a hypothesis whose plausibility is to be assessed.

Null hypothesis: The statement being tested.

Alternative hypothesis: A competing statement.

In general, the alternative hypothesis is chosen as the statement for which we hope to find supporting evidence.

In the case of our M31 LF researcher, the null hypothesis is $H_0: M_0 = -7.7$

An alternative hypothesis is $H_a: M_0 \neq -7.7$

Two-sided alternative hypotheses

One-sided alternatives, e.g., $H_a: M_0 < -7.7$

To test H_0 vs. H_a , we need:

(a) A *test statistic*: This statistic will be calculated from the observed data, and will measure the compatibility of H_0 with the observed data. It will have a sampling distribution free of unknown parameters.

(b) A *rejection rule* which specifies the values of the test statistic for which we reject H_0 .

Example: A random sample of 64 measurements has mean $\bar{x} = 5.2$ and std. dev. $s = 1.1$. 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:

$$H_0 : \mu = 4.9, \quad H_a : \mu \neq 4.9$$

2. The test statistic:

$$T = \frac{\bar{X} - 4.9}{S/\sqrt{n}}$$

3. The distribution of the test statistic under the assumption that H_0 is valid: $T \approx N(0, 1)$

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 is also called a *critical value*.

This choice of critical value results in a 5% *level of significance* of the test of hypotheses.

5. Calculate the value of the test statistic:

The calculated value of the test statistic 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 data are *significant* and that there is a *statistically significant* difference between the population mean and the hypothesized value of 4.9

7. The P -value of the test:

The smallest significance level at which the data are significant.

Summer School in Statistics for Astronomers IV

June 9-14, 2008

Maximum Likelihood Estimation
Cramer-Rao Inequality
Bayesian Information Criterion

Don Richards
Department of Statistics
Penn State University

Presented by Tom Hettmansperger
Department of Statistics
Penn State University

The Method of Maximum Likelihood

R. A. Fisher (1912), “On an absolute criterion for fitting frequency curves,” *Messenger of Math.* **41**, 155–160

Fisher’s first mathematical paper, written while a final-year undergraduate in mathematics and mathematical physics at 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, least-squares and the method of moments.

X : a random variable

θ is a parameter

$f(x; \theta)$: A statistical model for X

X_1, \dots, X_n : A random sample from X

We want to construct good estimators for θ

Protheroe, et al. "Interpretation of cosmic ray composition - The path length distribution," ApJ., 247 1981

X : Length of paths

Parameter: $\theta > 0$

Model: The exponential distribution,

$$f(x; \theta) = \theta^{-1} \exp(-x/\theta), \quad x > 0$$

Under this model,

$$E(X) = \int_0^\infty x f(x; \theta) dx = \theta$$

Intuition suggests using \bar{X} to estimate θ

\bar{X} is unbiased and consistent

LF for globular clusters in the Milky Way; van den Bergh's normal model,

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

μ : Mean visual absolute magnitude

σ : Std. deviation of visual absolute magnitude

\bar{X} is a good estimator for μ

S^2 is a good estimator for σ^2

We seek a method which produces good estimators automatically

Fisher's brilliant idea: The method of maximum likelihood

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 ,

$$P(X = c) = 0$$

Suppose $X \sim N(\mu = -6.9, \sigma^2 = 1.21)$

X has probability density function

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

$P(X = -7.1) = 0$, but

$$\begin{aligned} f(-7.1) &= \frac{1}{1.1\sqrt{2\pi}} \exp\left[-\frac{(-7.1 + 6.9)^2}{2(1.1)^2}\right] \\ &= 0.37 \end{aligned}$$

Interpretation: In one simulation of the random variable X , the “likelihood” of observing the number -7.1 is 0.37

$$f(-7.2) = 0.28$$

In one simulation of X , the value $x = -7.1$ is 32% more likely to be observed than the value $x = -7.2$

$x = -6.9$ is the value which has the greatest (or maximum) likelihood, for it is where the probability density function is at its maximum

Return to a general model $f(x; \theta)$

Random sample: X_1, \dots, X_n

Recall that the X_i are independent random variables

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 θ is fixed

Fisher's brilliant idea: Reverse the roles of the x 's and θ

Regard the X 's as fixed and θ as the variable

The *likelihood function* is

$$L(\theta; X_1, \dots, X_n) = f(X_1; \theta)f(X_2; \theta) \cdots f(X_n; \theta)$$

Simpler notation: $L(\theta)$

$\hat{\theta}$, the *maximum likelihood estimator* of θ , is the value of θ where L is maximized

$\hat{\theta}$ is a function of the X 's

Caution: The MLE is not always unique.

Example: "... cosmic ray composition - The path length distribution ..."

X : Length of paths

Parameter: $\theta > 0$

Model: The exponential distribution,

$$f(x; \theta) = \theta^{-1} \exp(-x/\theta), \quad x > 0$$

Random sample: X_1, \dots, X_n

Likelihood function:

$$\begin{aligned} L(\theta) &= f(X_1; \theta) f(X_2; \theta) \cdots f(X_n; \theta) \\ &= \theta^{-n} \exp(-(X_1 + \cdots + X_n)/\theta) \\ &= \theta^{-n} \exp(-n\bar{X}/\theta) \end{aligned}$$

To maximize L , we use calculus

It is also equivalent to maximize $\ln L$:

$$\begin{aligned}\ln L(\theta) &= -n \ln(\theta) - n\bar{X}\theta^{-1} \\ \frac{d}{d\theta} \ln L(\theta) &= -n\theta^{-1} + n\bar{X}\theta^{-2} \\ \frac{d^2}{d\theta^2} \ln L(\theta) &= n\theta^{-2} - 2n\bar{X}\theta^{-3}\end{aligned}$$

Solve the equation $d \ln L(\theta)/d\theta = 0$:

$$\theta = \bar{X}$$

Check that $d^2 \ln L(\theta)/d\theta^2 < 0$ at $\theta = \bar{X}$

$\ln L(\theta)$ is maximized at $\theta = \bar{X}$

Conclusion: The MLE of θ is $\hat{\theta} = \bar{X}$

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

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

Assume that σ is known (1.1 mag, say)

Random sample: X_1, \dots, X_n

Likelihood function:

$$\begin{aligned} L(\mu) &= f(X_1; \mu) f(X_2; \mu) \cdots f(X_n; \mu) \\ &= (2\pi)^{-n/2} \sigma^{-n} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^n (X_i - \mu)^2\right] \end{aligned}$$

Maximize $\ln L$ using calculus: $\hat{\mu} = \bar{X}$

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

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

Both μ and σ are unknown

A likelihood function of two variables,

$$\begin{aligned} L(\mu, \sigma^2) &= f(X_1; \mu, \sigma^2) \cdots f(X_n; \mu, \sigma^2) \\ &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp \left[-\frac{1}{2\sigma^2} \sum_{i=1}^n (X_i - \mu)^2 \right] \end{aligned}$$

$$\ln L = -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (X_i - \mu)^2$$

$$\begin{aligned} \frac{\partial}{\partial \mu} \ln L &= \frac{1}{\sigma^2} \sum_{i=1}^n (X_i - \mu) \\ \frac{\partial}{\partial (\sigma^2)} \ln L &= -\frac{n}{2\sigma^2} + \frac{1}{2(\sigma^2)^2} \sum_{i=1}^n (X_i - \mu)^2 \end{aligned}$$

Solve for μ and σ^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

Example: "... cosmic ray composition ..."

Parameter: $\theta > 0$

$$\text{Model: } f(x; \theta) = \begin{cases} \exp(-(x - \theta)), & x \geq \theta \\ 0, & x < \theta \end{cases}$$

Random sample: X_1, \dots, X_n

$$\begin{aligned} L(\theta) &= f(X_1; \theta) \cdots f(X_n; \theta) \\ &= \begin{cases} \exp(-\sum_{i=1}^n (X_i - \theta)), & \text{all } X_i \geq \theta \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

$X_{(1)}$: The smallest observation in the sample

"all $X_i \geq \theta$ " is equivalent to " $X_{(1)} \geq \theta$ "

$$L(\theta) = \begin{cases} \exp(-n(\bar{X} - \theta)), & \theta \leq X_{(1)} \\ 0, & \text{otherwise} \end{cases}$$

Conclusion: $\hat{\theta} = X_{(1)}$

General Properties of the MLE $\hat{\theta}$

- (a) $\hat{\theta}$ may not be unbiased. 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 θ 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 θ and variance $1/B$ where

$$B = nE \left[\frac{\partial}{\partial \theta} \ln f(X; \theta) \right]^2$$

The asymptotic property can be used to develop confidence intervals for θ

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.

The ML Method for Linear Regression Analysis

Scatterplot data: $(x_1, y_1), \dots, (x_n, y_n)$

Basic assumption: The x_i 's are non-random measurements; the y_i are observations on Y , a random variable

Statistical model:

$$Y_i = \alpha + \beta x_i + \epsilon_i, \quad i = 1, \dots, n$$

Errors $\epsilon_1, \dots, \epsilon_n$: a random sample from $N(0, \sigma^2)$

Parameters: α, β, σ^2

$Y_i \sim N(\alpha + \beta x_i, \sigma^2)$: The Y_i 's are independent

The Y_i are not identically distributed, because they have differing means

The likelihood function is the joint density function of the observed data, Y_1, \dots, Y_n

$$\begin{aligned} L(\alpha, \beta, \sigma^2) &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[-\frac{(Y_i - \alpha - \beta x_i)^2}{2\sigma^2} \right] \\ &= (2\pi\sigma^2)^{-n/2} \exp \left[-\frac{\sum_{i=1}^n (Y_i - \alpha - \beta x_i)^2}{2\sigma^2} \right] \end{aligned}$$

Use partial derivatives to maximize L over all α, β and $\sigma^2 > 0$ (Wise advice: Maximize $\ln L$)

The ML estimators are:

$$\hat{\beta} = \frac{\sum_{i=1}^n (x_i - \bar{x})(Y_i - \bar{Y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad \hat{\alpha} = \bar{Y} - \hat{\beta}\bar{x}$$

and

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{\alpha} - \hat{\beta}x_i)^2$$

The ML Method for Testing Hypotheses

$X \sim N(\mu, \sigma^2)$; parameters μ and σ^2

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

Random sample: X_1, \dots, X_n

We wish to test $H_0 : \mu = 3$ vs. $H_a : \mu \neq 3$

Parameter space: The space of all permissible values of the parameters

$$\Omega = \{(\mu, \sigma) : -\infty < \mu < \infty, \sigma > 0\}$$

H_0 and H_a represent restrictions on the parameters, so we are led to parameter subspaces

$$\omega_0 = \{(\mu, \sigma) : \mu = 3, \sigma > 0\}$$

$$\omega_a = \{(\mu, \sigma) : \mu \neq 3, \sigma > 0\}$$

$$\begin{aligned}
 L(\mu, \sigma^2) &= f(X_1; \mu, \sigma^2) \cdots f(X_n; \mu, \sigma^2) \\
 &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp \left[-\frac{1}{2\sigma^2} \sum_{i=1}^n (X_i - \mu)^2 \right]
 \end{aligned}$$

Maximize $L(\mu, \sigma^2)$ over ω_0 and ω_a

The *likelihood ratio test statistic* 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: $0 \leq \lambda \leq 1$

$L(3, \sigma^2)$ is maximized over ω_0 at

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (X_i - 3)^2$$

$$\begin{aligned}\max_{\omega_0} L(3, \sigma^2) &= L\left(3, \frac{1}{n} \sum_{i=1}^n (X_i - 3)^2\right) \\ &= \left[\frac{n}{2\pi e \sum_{i=1}^n (X_i - 3)^2} \right]^{n/2}\end{aligned}$$

$L(\mu, \sigma^2)$ is maximized over ω_a at

$$\mu = \bar{X}, \quad \sigma^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$$

$$\begin{aligned}\max_{\omega_a} L(\mu, \sigma^2) &= L\left(\bar{X}, \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2\right) \\ &= \left[\frac{n}{2\pi e \sum_{i=1}^n (X_i - \bar{X})^2} \right]^{n/2}\end{aligned}$$

The likelihood ratio test statistic:

$$\begin{aligned}\lambda &= \left[\frac{n}{2\pi e \sum_{i=1}^n (X_i - 3)^2} \right]^{n/2} \div \left[\frac{n}{2\pi e \sum_{i=1}^n (X_i - \bar{X})^2} \right]^{n/2} \\ &= \left[\sum_{i=1}^n (X_i - \bar{X})^2 \div \sum_{i=1}^n (X_i - 3)^2 \right]^{n/2}\end{aligned}$$

λ is close to 1 iff \bar{X} is close to 3

λ is close to 0 iff \bar{X} is far from 3

This particular LRT statistic λ is equivalent to the t -statistic seen earlier

In this case, the ML method discovers the obvious test statistic

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

Parameter: θ

X : Random variable with model $f(x; \theta)$

The “support” of f is the region where $f > 0$

We assume that the “support” of f does not depend on θ

Random sample: X_1, \dots, X_n

Y : An unbiased estimator of θ

The Cramér-Rao Inequality: The smallest possible value that $\text{Var}(Y)$ can attain is $1/B$ where

$$B = nE \left[\frac{\partial}{\partial \theta} \ln f(X; \theta) \right]^2 = -nE \left[\frac{\partial^2}{\partial \theta^2} \ln f(X; \theta) \right]$$

Example: "... cosmic ray composition - The path length distribution ..."

X : Length of paths

Parameter: $\theta > 0$

Model: $f(x; \theta) = \theta^{-1} \exp(-x/\theta)$, $x > 0$

$$\ln f(X; \theta) = -\ln \theta - \theta^{-1} X$$

$$\frac{\partial^2}{\partial \theta^2} \ln f(X; \theta) = \theta^{-2} - 2\theta^{-3} X$$

$$\begin{aligned} E \left[\frac{\partial^2}{\partial \theta^2} \ln f(X; \theta) \right] &= E(\theta^{-2} - 2\theta^{-3} X) \\ &= \theta^{-2} - 2\theta^{-3} E(X) \\ &= \theta^{-2} - 2\theta^{-3} \theta \\ &= -\theta^{-2} \end{aligned}$$

The smallest possible value of $\text{Var}(Y)$ is θ^2/n

This is attained by \bar{X} . For this problem, \bar{X} is the best unbiased estimator of θ

Y : An unbiased estimator of a parameter θ

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 result 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 states that if Y is any unbiased estimator of θ then

$$\text{Var}(Y) \geq \frac{1}{nE\left[\frac{\partial}{\partial\theta} \ln f(X; \theta)\right]^2}$$

The Heisenberg uncertainty principle is known to be a consequence of the Cramér-Rao inequality.

Dembo, Cover, and Thomas (1991) 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. This remarkable paper demonstrates that there is a basic oneness among these various fields.

Reference

Dembo, Cover, and Thomas (1991), “Information-theoretic inequalities,” IEEE Trans. Information Theory 37, 1501–1518.

The Bayesian Information Criterion

Suppose that we have two competing statistical models

We can fit these models using residual sums of squares, the method of moments, the method of maximum likelihood, ...

The choice of model cannot be assessed entirely by these methods

By increasing the number of parameters, we can always reduce the residual sums of squares

Polynomial regression: By increasing the number of terms, we can reduce the residual sum of squares

More complicated models generally will have lower residual errors

A standard approach to hypothesis testing for *large* data sets is to use the Bayesian information criterion (BIC).

The BIC penalizes models with greater numbers of free parameters

Two competing models:

$$f_1(x; \theta_1, \dots, \theta_{m_1}) \text{ and } f_2(x; \phi_1, \dots, \phi_{m_2})$$

Random sample: X_1, \dots, X_n

Likelihood functions:

$$L_1(\theta_1, \dots, \theta_{m_1}) \text{ and } L_2(\phi_1, \dots, \phi_{m_2})$$

Bayesian Information Criterion:

$$\text{BIC} = 2 \ln \frac{L_1(\theta_1, \dots, \theta_{m_1})}{L_2(\phi_1, \dots, \phi_{m_2})} - (m_1 - m_2)n$$

The BIC balances any improvement in the likelihood with the number of model parameters used to achieve that improvement

Calculate all MLEs $\hat{\theta}_i$ and $\hat{\phi}_i$

Compute the estimated BIC:

$$\widehat{\text{BIC}} = 2 \ln \frac{L_1(\hat{\theta}_1, \dots, \hat{\theta}_{m_1})}{L_2(\hat{\phi}_1, \dots, \hat{\phi}_{m_2})} - (m_1 - m_2)n$$

General rules:

$\widehat{\text{BIC}} < 2$: Weak evidence that Model 1 is superior to Model 2

$2 \leq \widehat{\text{BIC}} \leq 6$: Moderate evidence that Model 1 is superior to Model 2

$6 < \widehat{\text{BIC}} \leq 10$: Strong evidence that Model 1 is superior to Model 2

$\widehat{\text{BIC}} > 10$: Very strong evidence that Model 1 is superior to Model 2

Exercise: Two competing models for globular cluster LF in the Galaxy

1. A Gaussian model (van den Bergh, 1985)

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

2. A t -distn. model (Secker 1992, AJ 104)

$$g(x; \mu, \sigma, \delta) = \frac{\Gamma(\frac{\delta+1}{2})}{\sqrt{\pi\delta}\sigma\Gamma(\frac{\delta}{2})} \left[1 + \frac{(x - \mu)^2}{\delta\sigma^2}\right]^{-\frac{\delta+1}{2}}$$

$$-\infty < \mu < \infty, \sigma > 0, \delta > 0$$

In each model, μ is the mean and σ^2 is the variance. In Model 2, δ is a shape parameter.

Maximum likelihood calculations suggest that Model 1 is inferior to Model 2.

Question: Is the increase in likelihood due to larger number of parameters?

This question can be studied using the BIC.

We use the data of Secker (1992), Table 1

We assume that the data constitute a *random sample*

TABLE 1. Milky Way sample.

M_V	R_{GC}	E_{B-V}	Name	M_V	R_{GC}	E_{B-V}	Name
-1.75	26.74	0.06	AM4	-7.37	3.50	0.42	N6712
-3.28	26.28	0.02	Pal13	-7.38	5.62	0.10	N6652
-3.86	8.62	0.30	E3	-7.42	4.01	0.08	N6809
-3.88	2.76	0.31	E452SC	-7.43	4.97	1.05	N6553
-4.08	14.93	0.02	Pal12	-7.45	35.39	0.05	N7006
-4.30	2.81	0.19	N6496	-7.48	24.49	0.10	N5694
-4.54	7.19	0.34	Pal11	-7.52	5.35	0.05	N6752
-4.91	16.03	0.03	Pal5	-7.55	13.91	0.27	IC4499
-5.20	6.74	0.26	N6838	-7.57	17.11	0.00	N1261
-5.24	5.14	0.30	Pal8	-7.64	6.97	0.45	N4372
-5.28	20.89	0.11	Arp2	-7.64	7.01	0.02	N7099
-5.52	23.35	0.01	N7492	-7.65	5.26	0.62	N6760
-5.57	35.64	0.40	Pal15	-7.65	20.78	0.05	N5834
-6.87	10.31	0.02	N4147	-7.70	6.27	0.27	N6284
-5.89	2.99	0.37	N6642	-7.77	2.17	0.37	N6293
-5.89	3.89	0.33	N6535	-7.77	9.84	0.03	N4590
-6.04	4.79	0.65	N6366	-7.79	2.65	0.74	N6139
-6.07	2.42	0.80	N6256	-7.85	2.86	0.22	N6093
-6.14	15.12	0.15	N2298	-7.85	18.59	0.01	N1904
-6.16	5.55	0.73	N6544	-7.86	3.20	0.38	N6273
-6.20	3.62	0.25	N6352	-7.86	9.18	0.04	N362
-6.24	2.04	0.62	N6528	-7.88	2.50	0.03	N6723
-6.32	12.19	0.37	N6426	-7.97	28.16	0.01	N6229
-6.41	22.21	0.10	Rp106	-7.98	2.05	0.32	N6333
-6.45	2.11	0.86	N6325	-7.99	4.74	0.56	N5946
-6.48	6.85	0.32	N4833	-8.02	2.57	0.37	N6626
-6.49	11.33	0.03	N288	-8.04	9.40	0.02	N6341
-6.53	5.04	0.08	N6362	-8.14	11.71	0.16	N6864
-6.54	2.01	0.40	N6342	-8.19	4.38	0.21	N6218
-6.66	2.32	0.20	N6717	-8.20	7.27	0.24	N5286
-6.68	4.58	0.43	N5927	-8.23	15.97	0.02	N1851
-6.76	3.50	0.33	N6171	-8.24	4.76	0.25	N5986
-6.80	2.78	0.61	N6453	-8.27	2.27	0.12	N6541
-6.91	16.66	0.00	N5466	-8.29	18.51	0.14	N5824
-6.93	11.40	0.04	N6101	-8.35	5.06	0.35	N6656
-6.95	3.03	0.35	N6144	-8.40	8.22	0.03	N6205
-6.95	12.39	0.03	N6981	-8.59	7.53	0.30	N6356
-6.96	6.61	0.10	N5897	-8.60	3.40	0.87	N6539
-6.97	2.79	0.52	N6304	-8.65	11.68	0.00	N5272
-7.03	2.25	0.38	N6235	-8.70	19.13	0.00	N5024
-7.04	6.05	0.18	N6397	-8.73	6.12	0.03	N5904
-7.08	16.56	0.02	N5053	-8.80	4.32	0.48	N6316
-7.17	8.82	0.21	N3201	-8.82	10.41	0.02	N7089
-7.18	3.14	1.09	N6517	-9.04	10.17	0.10	N7078
-7.19	9.29	0.21	N6779	-9.08	4.18	0.58	N6402
-7.26	11.67	0.11	N6934	-9.24	7.34	0.04	N104
-7.27	6.13	0.36	N6121	-9.25	10.85	0.22	N2808
-7.31	4.00	0.10	N6584	-9.33	13.13	0.14	N6715
-7.34	4.58	0.25	N6254	-9.34	3.38	0.35	N6388
-7.37	3.46	0.05	N6681	-10.28	6.34	0.11	N5139

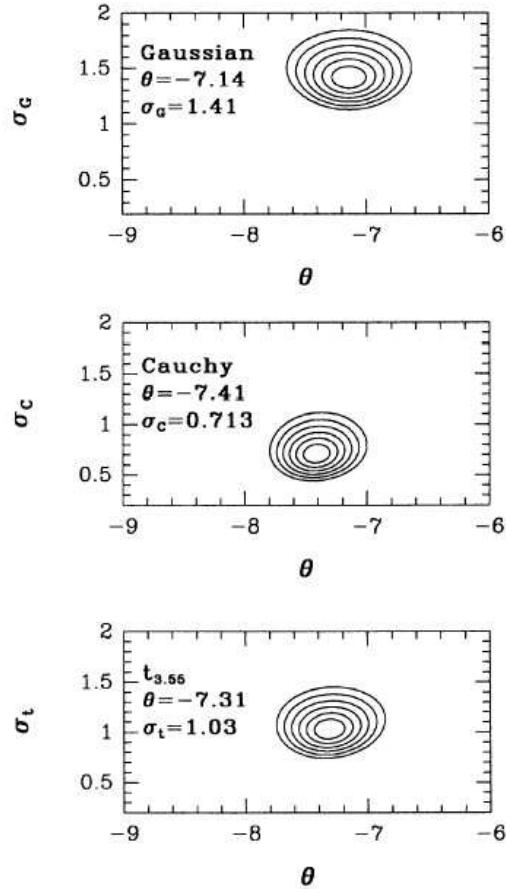


FIG. 1. Maximum-likelihood estimates for the Galactic GCLF expressed as contour plots in a two-dimensional parameter space, for the three distribution functions being considered. The most probable values for the parameters are given in the top left corner of the plot. The contours represent, from inner to outer, the 0.5, 1.0, 1.5, 2.0, 2.5, and 3.0 standard deviation probability limits on the maximum-likelihood parameter estimates.

Model 1: Write down the likelihood function,

$$\begin{aligned} L_1(\mu, \sigma) &= f(X_1; \mu, \sigma) \cdots f(X_n; \mu, \sigma) \\ &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp \left[-\frac{1}{2\sigma^2} \sum_{i=1}^n (X_i - \mu)^2 \right] \end{aligned}$$

Estimate μ with \bar{X} , the ML estimator. Also, estimate σ^2 with S^2 , a constant multiple of the ML estimator of σ^2 .

Note that

$$\begin{aligned} L_1(\bar{X}, S) &= \frac{1}{(2\pi S^2)^{n/2}} \exp \left[-\frac{1}{2S^2} \sum_{i=1}^n (X_i - \bar{X})^2 \right] \\ &= (2\pi S^2)^{-n/2} \exp(-(n-1)/2) \end{aligned}$$

Calculate \bar{x} and s^2 , the sample mean and variance of the Milky Way data. Use these values to calculate $L_1(\bar{x}, s)$

Secker (1992, p. 1476): $\ln L_1(\bar{x}, s) = -176.4$

Model 2: Write down the likelihood function,

$$\begin{aligned} L_2(\mu, \sigma, \delta) &= g(X_1; \mu, \sigma) \cdots g(X_n; \mu, \sigma) \\ &= \prod_{i=1}^n \frac{\Gamma(\frac{\delta+1}{2})}{\sqrt{\pi\delta}\sigma\Gamma(\frac{\delta}{2})} \left[1 + \frac{(X_i - \mu)^2}{\delta\sigma^2}\right]^{-\frac{\delta+1}{2}} \end{aligned}$$

Are the MLEs of μ, σ^2, δ unique?

No explicit formulas for the MLEs are known; we must evaluate them numerically

Substitute the Milky Way data for the X_i 's in the formula for L , and maximize L numerically.

Secker (1992): $\hat{\mu} = -7.31$, $\hat{\sigma} = 1.03$, $\hat{\delta} = 3.55$

Calculate $L_2(-7.31, 1.03, 3.55)$

Secker (1992, p. 1476):

$$\ln L_2(-7.31, 1.03, 3.55) = -173.0$$

Finally, calculate the estimated BIC:

$$\widehat{\text{BIC}} = 2 \ln \frac{L_1(\bar{x}, s)}{L_2(-7.31, 1.03, 3.55)} - (m_1 - m_2)n$$

where $m_1 = 2$, $m_2 = 3$, $n = 100$

$$\begin{aligned}\widehat{\text{BIC}} &= 2[\ln L_1(\bar{x}, s) - \ln L_2(-7.31, 1.03, 3.55)] \\ &\quad + 100 \\ &= 2[-176.4 - (-173.0)] + 100 \\ &= 93.2\end{aligned}$$

Apply the General Rules on p. 25 to assess the strength of the evidence that Model 1 may be superior to Model 2

Since $\widehat{\text{BIC}} > 10$, we have very strong evidence that Model 1 (the Gaussian model) is superior to Model 2 (the t -distribution model).

Concluding general remarks on the BIC

The BIC procedure is consistent: If Model 1 is the true model then, as $n \rightarrow \infty$, the BIC will determine that it is.

Not all information criteria are consistent.

The BIC is not a panacea; some authors recommend that it be used in conjunction with other information criteria.

There are also difficulties with the BIC

Findley (1991, Ann. Inst. Statist. Math.) studied the performance of the BIC for comparing two models with different numbers of parameters: “Suppose that the log-likelihood-ratio sequence of two models with different numbers of estimated parameters is bounded in probability. Then the BIC will, with asymptotic probability 1, select the model having fewer parameters.”

Summer School in Statistics for Astronomers IV

June 9-14, 2008

Mixture Models and The EM Algorithm

Tom Hettmansperger
Department of Statistics
Penn State University

Mixtures of Normal Distributions

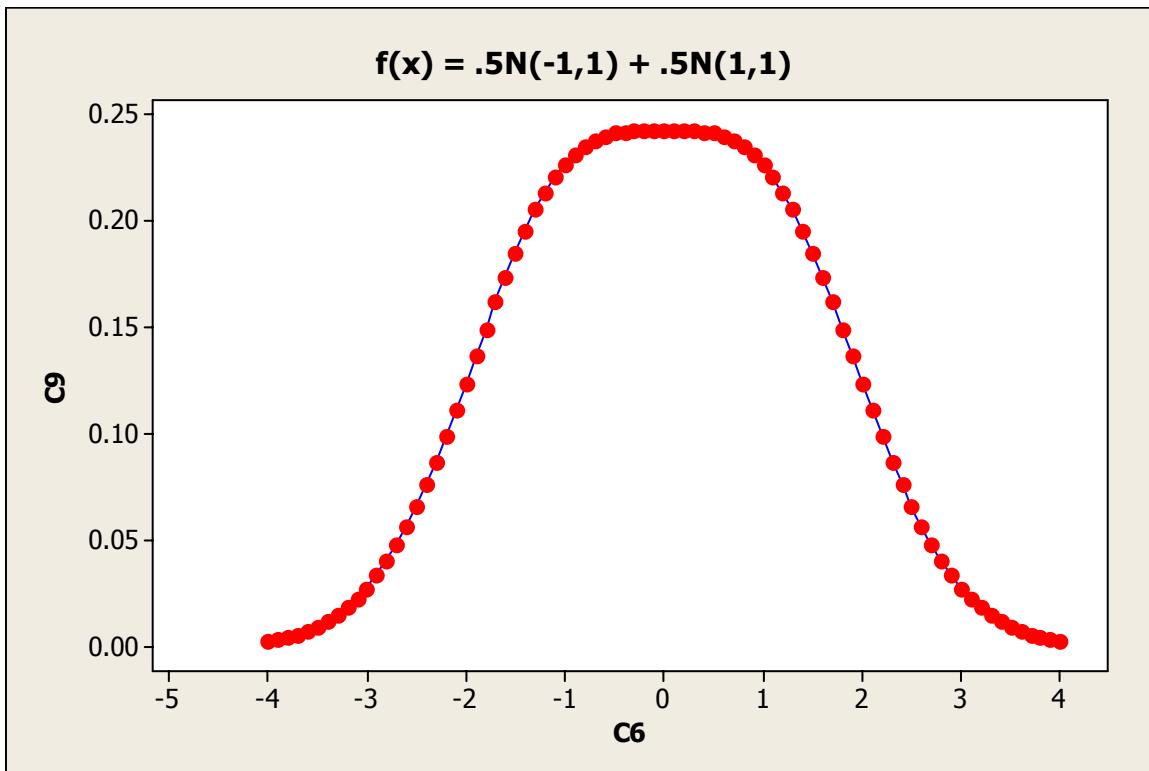
- Consider the problem of analyzing data that comes from a mixture of two or more normal populations.
- We **do not** have labels that indicate which population a particular data point comes from.
- If there are two populations then we have 5 parameters: a mixing proportion, 2 means, and 2 variances.
- We wish to estimate these 5 parameters and provide standard errors to assess their precision.

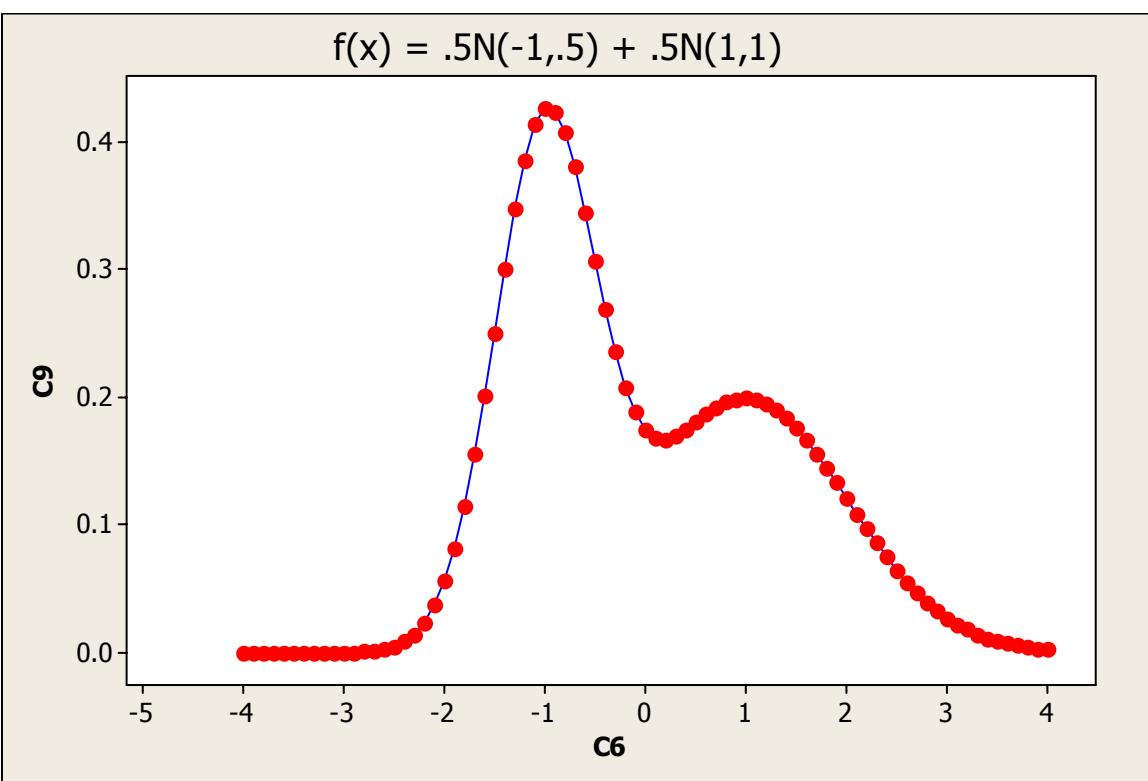
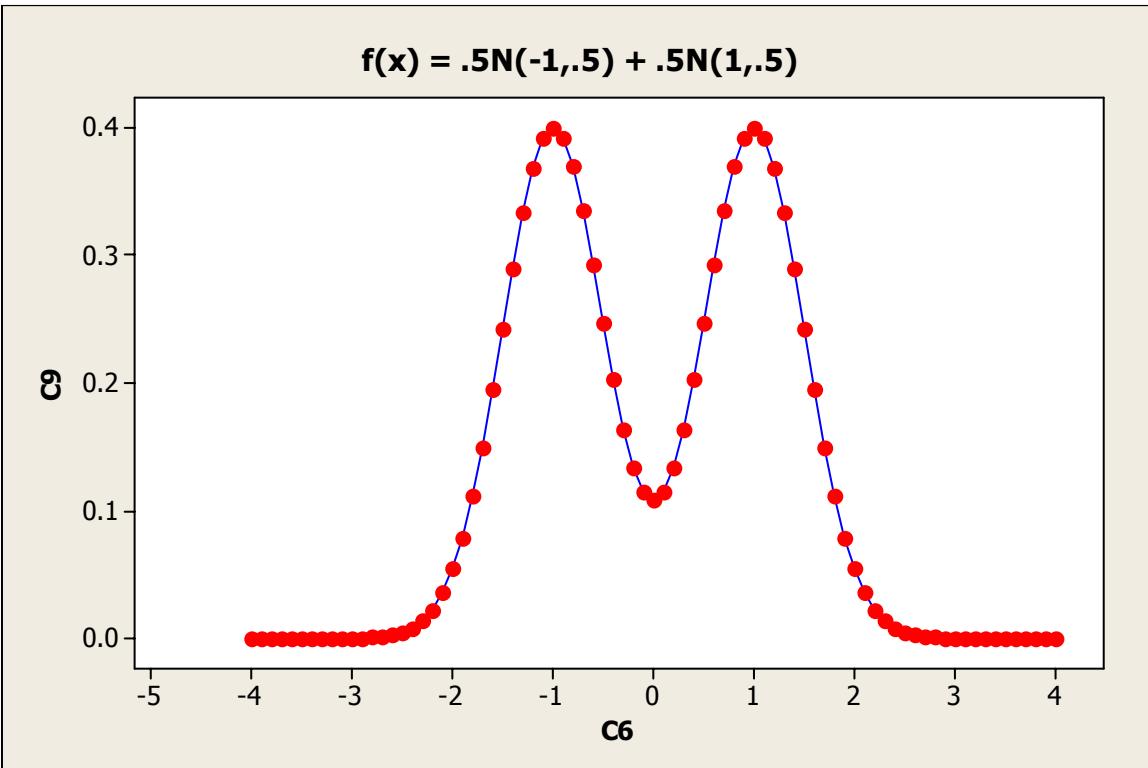
We write the model as:

$$f(x) = \lambda f_1(x) + (1 - \lambda) f_2(x)$$

Where $f_1(x)$ and $f_2(x)$ are the two normal pdfs.

Some examples of possible shapes
determined by mixing normal distributions.





We will use **maximum likelihood** to estimate the parameters. Given a sample x_1, \dots, x_n compute:

$$L(\Psi) = \prod_{i=1}^n \left\{ \lambda \frac{1}{\sqrt{2\pi} \sigma_1} \exp\left(-\frac{1}{2\sigma_1^2}(x_i - \mu_1)^2 + (1 - \lambda) \frac{1}{\sqrt{2\pi} \sigma_2} \exp\left(-\frac{1}{2\sigma_2^2}(x_i - \mu_2)^2\right) \right\}$$

where $\Psi^T = (\lambda, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$ is the parameter vector.

This likelihood function is quite difficult to maximize directly. Hence, we will introduce a new calculation device called the **EM algorithm**.

Suppose, for the moment, that we know which subpopulation the various data points come from.

Let $z_i = 1$ if x_i is from the first population and 0 otherwise. Then $P(z_i = 1) = \lambda$.

The **complete data** is given by $(x_1, z_1), \dots, (x_n, z_n)$.

The joint distribution of (X_i, Z_i) is given by

$$g(x, z) = f_1^z(x)f_2^{1-z}(x)\lambda^z(1 - \lambda)^{1-z}$$

where f_1 and f_2 are the two normal pdfs.

The complete data likelihood is given by:

$$L_c(\Psi) = \prod_{i=1}^n \left[\lambda \frac{1}{\sqrt{2\pi} \sigma_1} \exp\left(-\frac{1}{2\sigma_1^2} (x_i - \mu_1)^2\right) \right]^{z_i} \times$$

$$\left[(1 - \lambda) \frac{1}{\sqrt{2\pi} \sigma_2} \exp\left(-\frac{1}{2\sigma_2^2} (x_i - \mu_2)^2\right) \right]^{1-z_i}$$

$$\log L_c(\Psi) =$$

$$-\sum_{i=1}^n z_i \frac{1}{2\sigma_1^2} (x_i - \mu_1)^2 - \sum_{i=1}^n (1 - z_i) \frac{1}{2\sigma_2^2} (x_i - \mu_2)^2$$

$$+ \log \lambda \sum_{i=1}^n z_i + \log(1 - \lambda) \sum_{i=1}^n (1 - z_i)$$

$$- \log \sigma_1 \sum_{i=1}^n z_i - \log \sigma_2 \sum_{i=1}^n (1 - z_i) + K$$

$$\frac{\partial \log L_c}{\partial \mu_1} \Rightarrow \sum_{i=1}^n z_i(x_i - \mu_1) = 0 \text{ and } \hat{\mu}_1 = \frac{\sum z_i x_i}{\sum z_i}$$

$$\frac{\partial \log L_c}{\partial \sigma_1^2} \Rightarrow \hat{\sigma}_1^2 = \frac{\sum z_i(x_i - \hat{\mu}_1)^2}{\sum z_i}$$

$$\frac{\partial \log L_c}{\partial \lambda} \Rightarrow \hat{\lambda} = \frac{\sum z_i}{n}$$

Likewise for $\hat{\mu}_2$ and $\hat{\sigma}_2^2$.

Call this set of 5 estimates Eqns (1).

Easy to compute.

The problem: we don't know z_i !

The solution:

replace $\log L_c(\Psi)$ by $\log E(L_c(\Psi) \mid data)$

This only requires $E(Z_i \mid x_i) = P(Z_i = 1 \mid x_i)$

Recall Bayes formula:

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

$$E(Z_i \mid x_i) = P(Z_i = 1 \mid x_i) = \frac{\lambda f_1(x_i)}{\lambda f_1(x_i) + (1-\lambda)f_2(x_i)}$$

where $f_1(x_i) = f(x_i; \mu_1, \sigma_1^2)$ and
 $f_2(x_i) = f(x_i; \mu_2, \sigma_2^2)$ are the 2 normal pdfs.

Call this Eqn (2)

How it works:

Begin with a set of initial values:

$$\lambda^0, \mu_1^0, \mu_2^0, \sigma_1^0, \text{ and } \sigma_2^0$$

E-step: use *Eqn* (2) to compute: z_i^0

M-step: use z_i^0 and *Eqns* (1) to compute
 $\lambda^1, \mu_1^1, \mu_2^1, \sigma_1^1, \text{ and } \sigma_2^1$

Iterate between the E and M steps until convergence.

$$\widehat{\Psi} = (\widehat{\lambda}, \widehat{\mu}_1, \widehat{\mu}_2, \widehat{\sigma}_1, \widehat{\sigma}_2)$$

Did we **solve the wrong problem?**

We have maximized $L_c(\Psi)$ not the regular likelihood $L(\Psi)$.

Dempster, Laird, and Rubin (J. Royal Statist Soc B, 1977, p1-38) show that the solution to the complete data problem never decreases the regular likelihood.

Hence, we can use the EM algorithm to find the regular maximum likelihood estimates.

Dave Hunter will discuss the pitfalls.

Standard errors of the maximum likelihood estimates are estimated by estimating the information matrix.

The square roots of the diagonal elements of the inverse of the estimate of the **information matrix** divided by square root of n are the estimates of the standard errors.

The **bootstrap** can also be used to approximate the standard errors. This requires iterations (for the EM algorithm) inside the bootstrap iterations and can be computationally expensive.

Example: "We give here two small portions of the spectrum of a bright quasar described in the following study:

HIGH-RESOLUTION STIS/HUBBLE
SPACE TELESCOPE AND HIRES/KECK
SPECTRA OF THREE WEAK Mg ii
ABSORBERS TOWARD PG 1634+7061

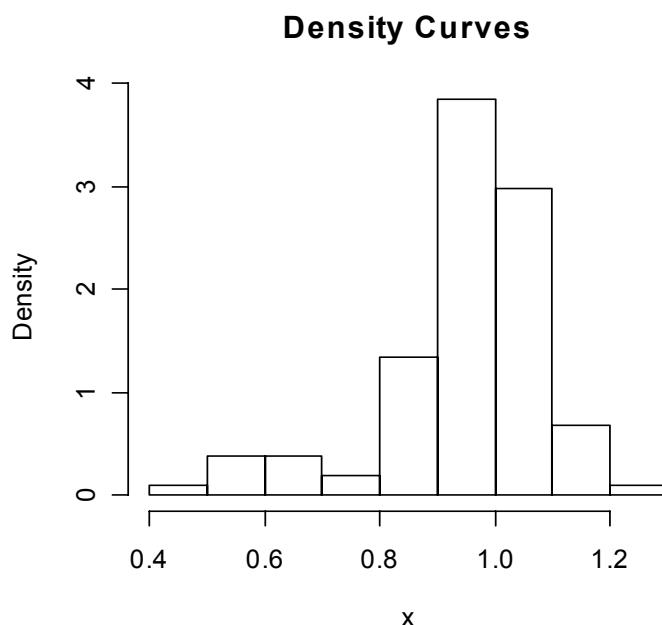
Jane C. Charlton, Jie Ding, Stephanie G. Zonak, Christopher W. Churchill, Nicholas A. Bond, and Jane R. Rigby

We give regions around the 3-times-ionized silicon line Si IV 1394 and the 3-times-ionized carbon line C IV 1551 for the $z=0.653411$ absorption system..."

[http://astrostatistics.psu.edu/datasets/
QSO_absorb.html](http://astrostatistics.psu.edu/datasets/QSO_absorb.html)

Data are the normalized intensities of the quasar light.

$n = 104$



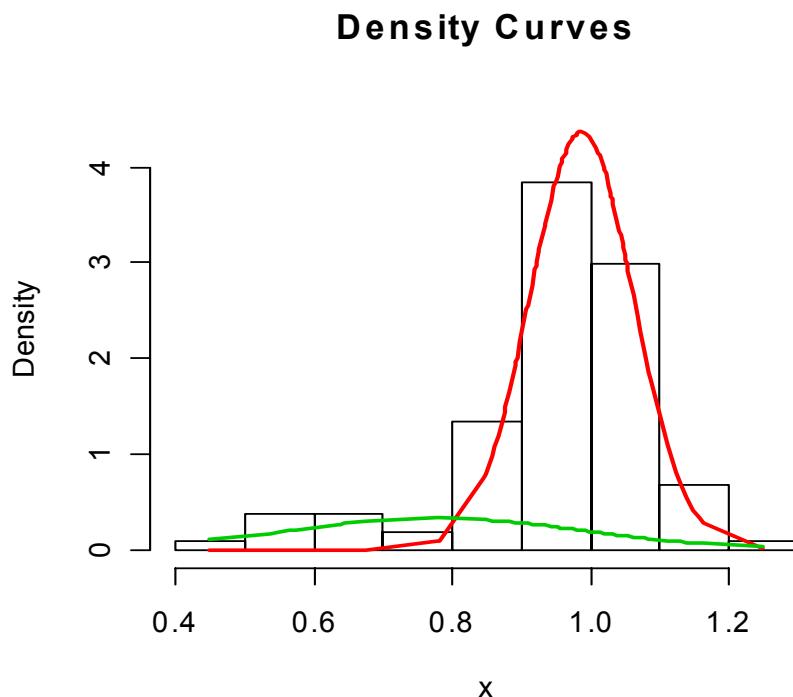
$$\text{loglik} = 57.4898$$

$$\text{BIC} = 2\text{loglik} - k \times \ln(n) = 105.69$$

$$\text{mean} = .95, \text{ Stdev} = .14$$

We will consider fitting normal mixture models with 2, 3, and 4 components.

First consider a **2 component mixture**:



Mean: 0.99 0.78

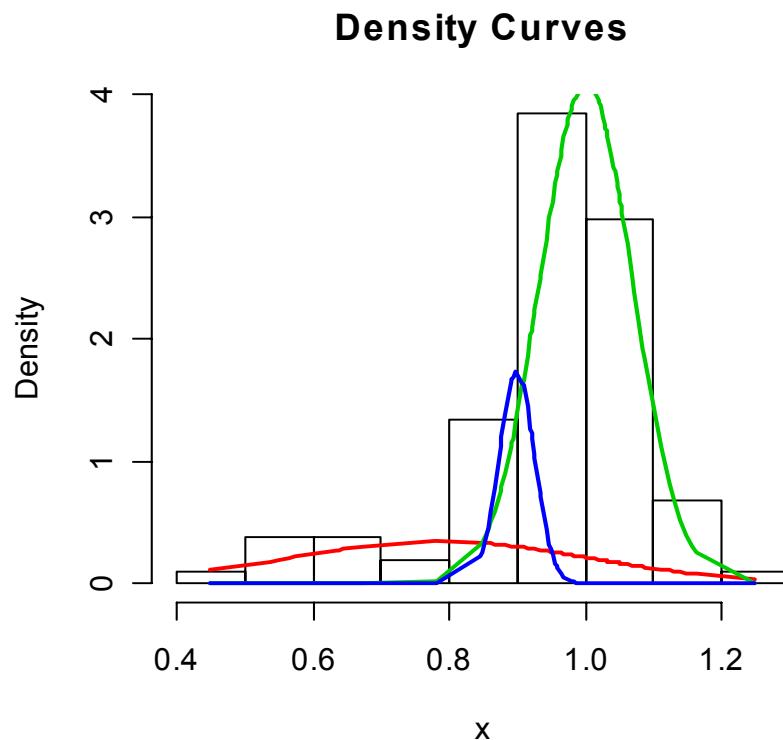
StDev: 0.075 0.22

lambda: 0.82 0.18

loglik = 78.91

BIC = $2\text{loglik} - k \times \ln(n) = 134.6$

Consider next a **3 component mixture**:



Mean: 0.786 1.000 0.899

StDev: 0.218 0.069 0.026

lambda: 0.19 0.70 0.11

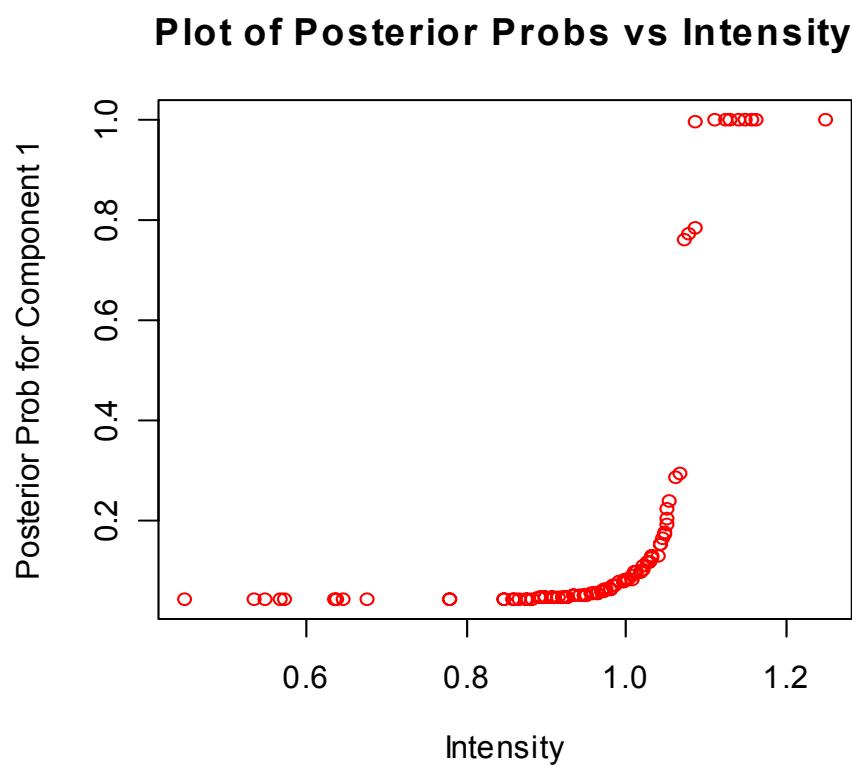
loglik = 80

BIC = 122.84

Recall for the 2 component mixture:

- Mean: 0.99 0.78
- StDev: 0.075 0.22
- lambda: 0.82 0.18

The EM algorithm also provides posterior probabilities that each data point comes from the first component.



Some references:

Finite Mixture Models by McLachlan and Peel

(Excellent book published in 2000 and covers a wide variety of topics in parametric mixture models with a lot of examples.)

Mixtools, a library of R functions developed by Derek Young.

(Excellent set of functions that covers a wide variety of applications including univariate parametric models, repeated measures, and regression. Model fitting and graphics.)

Likelihood Computations and Random Numbers

This tutorial focuses on likelihood computations and random number generation. We also discuss EM algorithms and mixture models.

Maximum likelihood estimation in a simple case

Let's reconsider the flux measurements in the gamma ray burst dataset.

```
grb <- read.table("http://astrostatistics.psu.edu/datasets/GRB_afterglow.dat",
  header=T, skip=1)
flux <- grb[,2]
hist(flux)
```

The histogram suggests that the univariate distribution has roughly the shape of an exponential distribution (we'll speak more about what this means later). Let us replot these data in a particular (and particularly common) manner besides the histogram that is also suggestive of an exponential distribution.

As a first step, let us calculate something akin to the (x,y) coordinates of the empirical distribution function -- the function that has a jump of size $1/n$ at every one of the sorted data points.

```
n <- length(flux)
xx <- sort(flux)
yy <- (1:n)/n
```

We could now obtain the empirical cdf by connecting the (xx,yy) points using a stair-step pattern. However, we'll look at these points slightly differently.

The exponential distribution has a distribution function given by $F(x) = 1 - \exp(-x/\mu)$ for positive x , where $\mu > 0$ is a scalar parameter equal to the mean of the distribution. This implies among other things that $\log(1-F(x)) = -x/\mu$ is a linear function of x in which the slope is the negative reciprocal of the mean. Let us then look for the characteristic linear pattern if we plot $\log(1-F(x))$ against x using the empirical distribution function for F :

```
plot(xx, log(1-yy+1/n), xlab="flux",
  ylab="log(1-F(flux)))")
```

You may recall from the [EDA and regression](#) tutorial that this plot looks like the plot of time vs. flux that we produced as part of a regression analysis. This is only a coincidence; *the two plots are fundamentally different*. The plot seen here is a univariate plot, whereas the time vs. flux plot was a bivariate plot. We are ignoring the time variable here.

The plot certainly looks linear, so let us proceed on the assumption that the flux data are a sample from an exponential distribution with unknown parameter μ .

The overriding question of this section is this: How shall we estimate μ ?

As mentioned above, μ is equal to the mean of this population. For a quick refresher on some probability theory, let us recall why this is so: The first step in going from the distribution function $F(x) = 1 - \exp(-x/\mu)$ to the mean, or expectation, is to obtain the density function by differentiating: $f(x) = \exp(-x/\mu)/\mu$. Notice that we typically use $F(x)$ to denote the distribution function and $f(x)$ to denote the density function. Next, we integrate $x*f(x)$ over the interval 0 to infinity, which gives the mean, μ .

Since μ is the population mean, it is intuitively appealing to simply estimate μ using the sample mean. This method, in which we match the population moments to the sample moments and then solve for the parameter estimators, is called the method of moments. Though it is a well-known procedure, we focus instead on a much more widely used method (for good reason) called maximum likelihood estimation.

The first step in maximum likelihood estimation is to write down the likelihood function, which is nothing but the joint density of the dataset viewed as a function of the parameters. Next, we typically take the log, giving what is commonly called the log likelihood function. Remember that all logs are natural logs unless specified otherwise.

The log likelihood function in this case is (with apologies for the awkward notation)
 $l(\mu) = -n \log(\mu) - x_1/\mu - \dots - x_n/\mu$

A bit of calculus reveals that $l(\mu)$ is therefore maximized at the sample mean. Thus, the sample mean is not only the method of moments estimator in this case but the maximum likelihood estimate as well.

In practice, however, it is sometimes the case that the linear-looking plot produced earlier is used to estimate μ . As we remarked, the negative reciprocal of the slope should give μ , so there is a temptation to fit a straight line using, say, least-squares regression, then use the resulting slope to estimate μ .

```
mean (flux) # This is the MLE
m1 <- lm(log(1-yy+1/n) ~ xx)
m1
-1/m1$coef[2] # An alternative estimator
```

There is a possible third method that I am told is sometimes used for some kinds of distributions. We start with a histogram, which may be viewed as a rough approximation of the density:

```
h <- hist(flux)
```

All of the information used to produce the histogram is now stored in the `h` object, including the midpoints of the bins and their heights on a density scale (i.e., a scale such that the total area of the histogram equals one).

To see how to use this information, note that the logarithm of the density function is $\log f(x) = -\log(\mu) - x/\mu$, which is a linear function of x . Thus, plotting the logarithm of the density against x might be expected to give a line.

```
counts <- h$counts
dens <- h$density[counts>0]
midpts <- h$mids[counts>0]
plot(midpts, log(dens))
```

When using linear regression to estimate the slope of the linear pattern just produced, I am told that it is standard to weight each point by the number of observations it represents, which is proportional to the reciprocal of the variance of the estimated proportion of the number of points in that bin. We can obtain both the weighted and unweighted versions here. We can then obtain an estimate of μ using either the intercept, which is $-\log(\mu)$, or the slope, which is $-1/\mu$:

```
m1 <- lm(log(dens) ~ midpts,
m2 <- lm(log(dens) ~ midpts,
weights=counts[counts>0])
exp(-m1$coef[1]) # This is one estimate
-1/m1$coef[2] # This is another
exp(-m2$coef[1]) # Yet another
-1/m2$coef[2] # And another
```

We have thus produced no fewer than six different estimators of μ (actually seven, except that the MLE and the method of moments estimator are the same in this case). How should we choose one of them?

There are a couple of ways to answer this question. One is to appeal to statistical theory. The method of maximum likelihood estimation is backed by a vast statistical literature that shows it has certain properties that may be considered optimal. The method of moments is also a well-established method, but arguably with less general theory behind it than the method of maximum likelihood. The regression-based methods, on the other hand, are all essentially ad hoc.

A second way to choose among estimators is to run a simulation study in which we repeatedly simulate datasets (whose parameters are then known to us) and test the estimators to see which seems to perform best. In order to do this, we will need to be able to generate random numbers, which is the next topic in this tutorial.

Standard error of the MLE

Based on asymptotic theory (i.e., the mathematics of statistical estimators as the sample size tends to infinity), we know that for many problems, the sampling distribution of the maximum likelihood estimator (for theta) is roughly normally distributed with mean theta and variance equal to the inverse of the Fisher information of the dataset. For an i.i.d. sample of size n, the Fisher information is defined to be n times the mean of the square of the first derivative of the log-density function. Equivalently, it is -n times the mean of the second derivative of the log-density function. *In many cases, it is easier to use the second derivative than the square of the first derivative.*

In the previous example, the log-density function is $-\log(\mu) - x/\mu$. The second derivative with respect to the parameter is $1/\mu^2 - 2x/\mu^3$. To find the Fisher information, consider x to be a random variable; we know its expectation is mu, so the expectation of the second derivative equals $-1/\mu^2$. We conclude that the Fisher information equals n/μ^2 . Intuitively, this means that we get more "information" about the true value of mu when this value is close to zero than when it is far from zero. For the exponential distribution with mean mu, this makes sense.

Earlier we calculated the MLE. Let's give it a name:

```
mu.hat <- mean (flux) # The MLE found earlier
```

The standard error of mu.hat, based on the Fisher information, is the square root of the inverse of the Fisher information evaluated at mu.hat:

```
sqrt(mu.hat/n) # SE based on (expected) information
mu.hat + 1.96 * c(-1,1) * sqrt(mu.hat/n) # approx. 95% CI
```

The Fisher information calculated above is sometimes called the *expected* information because it involves an expectation. As an alternative, we can use what is called the *observed* information, which is the negative second derivative of the log-likelihood function. In this example, the log likelihood function evaluated at the estimate mu.hat is equal to

```
-n * log(mu.hat) - sum(flux) / mu.hat
```

and the negative second derivative of this function, evaluated at mu.hat, is

```
-n / mu.hat^2 + 2*sum(flux) / mu.hat^3 # observed information at MLE
```

Notice in this case (though not in every model) that the observed information evaluated at the MLE is equal to the expected information evaluated at the MLE:

```
n / mu.hat^2 # expected information at MLE
```

Do you see why?

Generating random numbers in R

First, some semantics: "Random numbers" does not refer solely to uniform numbers between 0 and 1, though this is what "random numbers" means in some contexts. We are mostly interested in generating non-uniform random numbers here.

R handles many common distributions easily. To see a list, type

```
help.search("distribution", package="stats")
```

Let's consider the well-known normal distribution as an example:

```
?Normal
```

The four functions '[rnorm](#)', '[dnorm](#)', '[pnorm](#)', and '[qnorm](#)' give random normals, the normal density (sometimes called the differential distribution function), the normal cumulative distribution function (CDF), and the inverse of the normal CDF (also called the quantile function), respectively. Almost all of the other distributions have similar sets of four functions. The 'r' versions are [rbeta](#), [rbinom](#), [rcauchy](#), [rchisq](#), [rexp](#), [rf](#), [rgamma](#), [rgeom](#), [rhyper](#), [rllogis](#), [rlnorm](#), [rmultinom](#), [rnbino](#), [rnorm](#), [rpois](#), [rsignrank](#), [rt](#), [runif](#), [rweibull](#), and [rwilcox](#) (there is no [rtukey](#) because generally only [ptukey](#) and [qtukey](#) are needed). Additional distributions are available in other packages.

As an example, suppose we wish to simulate a vector of 10 independent, standard (i.e., mean 0 and standard deviation 1) normal random variables. We use the [rnorm](#) function for this purpose, and its defaults are mean=0 and standard deviation=1. Thus, we may simply type

```
rnorm(10)
```

Suppose we wish to simulate a large number of normal random variables with mean 10 and standard deviation 3, then check a histogram against two normal density functions, one based on the true parameters and one based on estimates, to see how it looks. We'll use 'col=2, lty=2, lwd=3' to make the curve based on the true parameters red (color=2), dashed (line type=2), and wider than normal (line width=3). Also note that we are requesting 100 bins in the histogram (nclass=100) and putting it on the same vertical scale as the density functions (freq=FALSE).

```
z <- rnorm(200000, mean=10, sd=3)
hist(z, freq=FALSE, nclass=100)
x <- seq(min(z), max(z), len=200)
lines(x, dnorm(x, mean=10, sd=3), col=2, lty=2, lwd=3)
lines(x, dnorm(x, mean=mean(z), sd=sqrt(var(z))))
```

We can find out what proportion of the deviates lie outside 3 standard deviations from the true mean, a common cutoff used by physical scientists. We can also see the true theoretical proportion:

```
sum(abs((z-10)/3)>3)/length(z)
2*pnorm(-3)
```

In the first line above, we are using [sum](#) to count the number of TRUE's in the logical vector (abs((z-10)/3)>3). This works because logical values are coerced to 0's and 1's when necessary.

The function [dnorm](#) has a closed form: With mean=0 and sd=1, dnorm(x) equals $\exp(-x^2/2)/\sqrt{2\pi}$. By contrast, the CDF, given by [pnorm](#), has no closed form and must be numerically approximated. By definition, pnorm(x) equals the integral of dnorm(t) as t ranges from minus infinity to x. To find a p-value (i.e., the probability of observing a statistic more extreme than the one actually observed), we use pnorm; to construct a confidence interval (i.e., a range of reasonable values for the true parameter), we use the inverse, [qnorm](#).

```
pnorm(1:3)-pnorm(-(1:3))
qnorm(c(.05,.95))
```

The first line above summarizes the well-known 68, 95, 99.7 rule for normal distributions (these are the approximate proportions lying within 1, 2, and 3 standard deviations from the mean). The second line gives the critical values used to construct a 90% confidence interval for a parameter when its estimator is approximately normally distributed.

Let us now briefly consider an example of a *discrete* distribution, which means a distribution on a finite or countably infinite set (as opposed to a *continuous* distribution like the normal). The [Poisson](#) distribution, which has a single real-valued parameter lambda, puts all of its probability mass on the nonnegative

integers. A Poisson distribution, often used to model data consisting of counts, has mean and variance both equal to lambda.

```
k <- 0:10
dpois(k,lambda=2.5) # or equivalently,
exp(-2.5)*2.5^k/factorial(k)
```

Next, simulate some Poisson variables:

```
x <- rpois(10000,lambda=2.5)
table(x)
mean(x)
var(x)
```

We can also illustrate the CDF for these Poisson variables:

```
hist(x,freq=F)
x.un <- sort(unique(x))
x.cdf=ppois(x.un,lambda=2.5)
x.sim=cumsum(table(x)/10000)
cbind(x.sim,x.cdf)
```

The power-law or Pareto distribution

A commonly used distribution in astrophysics is the power-law distribution, more commonly known in the statistics literature as the Pareto distribution. The R package 'actuar' actually defines this distribution, but because it is an extremely simple distribution, we will simply write the necessary functions. (For the purposes of this tutorial, do not load the 'actuar' package.)

The density function for the Pareto is $f(x)=ab^a/x^{(a+1)}$ for $x>b$. Here, a and b are fixed positive parameters, where b is the minimum possible value. (Note: The 'actuar' package shifts x by b , so $f(x)=ab^a/(x+b)^{(a+1)}$ for $x>0$.) As an example, consider the $\log N = -1.5 * \log S$ relationship, where S is the apparent brightness and N is the number of standard candles randomly located in transparent space. Thus, a Pareto distribution with $(a+1) = 1.5$ is a reasonable, if simplistic, model for the brightness of observed standard candles in space. The b parameter merely reflects the choice of units of measurement.

As another example, consider the Salpeter function, the simple but widely known expression of the initial mass function (IMF), in which the mass of a randomly selected newly formed star has a Pareto distribution with parameter $a=1.35$.

It turns out that a Pareto random variable is simply $b*\exp(X)$, where X is an [exponential](#) random variable with $\text{rate}=a$ (i.e., with $\text{mean}=1/a$). However, rather than exploiting this simple relationship, we wish to build functions for the Pareto distribution from scratch. Our default values, which may be changed by the user, will be $a=0.5$ and $b=1$.

```
dpareto <- function(x, a=0.5, b=1) a*b^a/x^(a+1)
```

Next, we integrate the density function to obtain the distribution function, which is $F(x)=1-(b/x)^a$ for $x>=b$ (and naturally $F(x)=0$ for $x < b$):

```
ppareto <- function(x, a=0.5, b=1) (x > b)*(1-(b/x)^a)
```

Note that $(x > b)$ in the above function is coerced to numeric, either 0 or 1.

Inverting the distribution function gives the quantile function. The following simplistic function is wrong unless $0 < u < 1$, so a better-designed function should do some error-checking.

```
qpareto <- function(u, a=0.5, b=1) b/(1-u)^(1/a)
```

Finally, to simulate random Pareto random variables, we use the fact that whenever the quantile function is applied to a uniform random variable, the result is a random variable with the desired distribution:

```
rpareto <- function(n, a=0.5, b=1) qpareto(runif(n),a,b)
```

Creating functions in R, as illustrated above, is a common procedure. Note that each of the arguments of a function may be given a default value, which is used whenever the user calls the function without specifying the value of this parameter. Also note that each of the above functions consists of only a single line; however, longer functions may be created by enclosing them inside curly braces { }.

A few simple plots

The commands below create plots related to the four functions just created.

```
par(mfrow=c(2,2))
x <- seq(1,50,len=200)
plot(x,dpareto(x),type="l")
plot(x,ppareto(x),type="l",lty=2)
u <- seq(.005,.9,len=200)
plot(u,qpareto(u),type="l",col=3)
z <- rpareto(200)
dotchart(log10(z), main="200 random logged Pareto deviates",
         cex.main=.7)
par(mfrow=c(1,1))
```

The above commands illustrate some of the many plotting capabilities of R. The [par](#) function sets many graphical parameters, for instance, 'mfrow=c(2,2)', which divides the plotting window into a matrix of plots, set here to two rows and two columns. In the [plot](#) commands, 'type' is set here to "l" for a line plot; other common options are "p" for points (the default), "b" for connected dots, and "n" for nothing (to create axes only). Other options used: 'lty' sets the line type (1=solid, 2=dashed, etc.), 'col' sets color (1=black, 2=red, 3=green, etc.), 'main' puts a string into the plot title, and 'cex.main' sets the text magnification.

Type '[?par](#)' to see a list of the many plotting parameters and options.

A simulation study

Let us posit that the random variable X has a Pareto distribution with parameters $a=1.35$ and $b=1$. We will simulate multiple datasets with this property and then apply several different estimation methods to each. To simplify matters, we will assume that the value of b is known, so that the goal is estimation of a.

To evaluate the estimators, we will look at their mean squared error (MSE), which is just what it sounds like: The average of the squared distances from the estimates to the true parameter value of $a=1.35$.

To illustrate the estimators we'll evaluate, let's start by simulating a single dataset of size 100:

```
d <- rpareto(100, a=1.35)
```

Here are the estimators we'll consider:

1. The **maximum likelihood estimator**. Since the density with $b=1$ is given by $f(x) = a/x^{a+1}$, the log likelihood function is $l(a) = n \log(a) - (a+1)(\log x_1 + \log x_2 + \dots + \log x_n)$. The maximizer may be found using calculus to equal $n/(\log x_1 + \dots + \log x_n)$. For our dataset, this may be found as follows:

```
1/mean(log(d))
```

We used the sum of logarithms above where we could have used the equivalent mathematical expression given by the log of the product. Sometimes the former method gives more numerically stable answers for very large samples, though in this case "100/log(prod(d))" gives exactly the same answer.

2. The **method of moments estimator**. By integrating, we find that the mean of the Pareto distribution with $b=1$ is equal to $a/(a-1)$. (This fact requires that a be greater than 1.) Setting $a/(a-1)$ equal to the sample mean and solving for a gives $1/(1-1/\text{samplemean})$ as the estimator.

```
1 / (1 - 1 / mean(d))
```

3. What we'll call the **EDF (empirical distribution function) estimator**. Since $\log(1-F(x))$ equals $-a \log(x)$ when $b=1$, by plotting the sorted values of $\log(d)$ against $\log(n/n), \log((n-1)/n), \dots, \log(1/n)$, we should observe roughly a straight line. We may then use least-squares regression to find the slope of the line, which is our estimate of $-a$:

```
lsd <- log(sort(d))
lseq <- log((100:1)/100)
plot(lsd, lseq)
tmp <- lm(lseq ~ lsd)
abline(tmp, col=2)
-tmp$coef[2]
```

4. What we'll call the **unweighted histogram estimator**. Since $\log f(x)$ equals $\log(a) - (a+1) \log(x)$ when $b=1$, if we plot the values of $\log(d)$ against histogram-based estimates of the log-density function, we should observe roughly a straight line with slope $-(a+1)$ and intercept $\log(a)$. Let's use only the slope, since that is the feature that is most often the focus of a plot that is supposed to illustrate a power-law relationship.

```
hd <- hist(d, nclass=20, plot=F)
counts <- hd$counts
ldens <- log(hd$density[counts>0])
lmidpts <- log(hd$midpoints[counts>0])
plot(lmidpts, ldens)
tmp <- lm(ldens ~ lmidpts)
abline(tmp, col=2)
-1-as.numeric(tmp$coef[2])
```

5. What we'll call the **weighted histogram estimator**. Exactly the same as the unweighted histogram estimator, but we'll estimate the slope using weighted least squares instead of ordinary least squares. The weights should be proportional to the bin counts.

```
plot(lmidpts, ldens)
tmp <- lm(ldens ~ lmidpts,
          weights=counts[counts>0])
abline(tmp, col=2)
-1-as.numeric(tmp$coef[2])
```

Now let's write a single function that will take a vector of data as its argument, then return all five of these estimators.

```
five <- function(d) {
  lsd <- log(sort(d))
  n <- length(d)
  lseq <- log((n:1)/n)
  m1 <- lm(lseq ~ lsd)$coef
  hd <- hist(d, nclass=n/5, plot=F)
  counts <- hd$counts
  ldens <- log(hd$density[counts>0])
```

```

lmidpts <- log(hd$mids[counts>0])
m2 <- lm(ldens~lmidpts)$coef
m3 <- lm(ldens~lmidpts,
  weights=counts[counts>0])$coef
out <- c(max.lik=1/mean(log(d)),
  meth.mom=1/(1-1/mean(d)),
  EDF=-as.numeric(m1[2]),
  unwt.hist=-1-as.numeric(m2[2]),
  wt.hist=-1-as.numeric(m3[2]))
return(out)
}

```

The very last line of the function, "return(out)", is the value that will be returned. (We could also have simply written "out".) Let's test this function on our dataset:

```
five(d)
```

There is no good way to compare these estimators based on a single sample like this. We now need to simulate multiple samples. Let's begin by taking n=100.

```

n.100 <- NULL
for(i in 1:250) {
  dd <- rpareto(100, a=1.35)
  n.100 <- rbind(n.100, five(dd))
}

```

Now we can get estimates of the biases of the estimators (their expectations minus the true parameter) and their variances. Note that we'll use the biased formula for the variance (i.e., the one that uses n instead of n-1 in the denominator) for a technical reason explained below.

```

bias.100 <- apply(n.100, 2, mean) - 1.35
var.100 <- apply(n.100, 2, var) * (249/250)

```

It is a mathematical identity that the mean squared error (MSE) equals the square of the bias plus the variance, as we may check numerically for (say) the first column of n.100. However, the identity only works if we use the biased formula for the variance, which is why we used the multiplier (249/250) above.

```

mean((n.100[,1]-1.35)^2)
bias.100[1]^2 + var.100[1]

```

Thus, we can construct the MSEs and view the results as follows:

```

mse.100 <- bias.100^2 + var.100
rbind(bias.100, var.100, mse.100)

```

Finally, let's repeat the whole experiment using samples of size 200.

```

n.200 <- NULL
for(i in 1:250) {
  dd <- rpareto(200, a=1.35)
  n.200 <- rbind(n.200, five(dd))
}
bias.200 <- apply(n.200, 2, mean) - 1.35
var.200 <- apply(n.200, 2, var) * (249/250)
mse.200 <- bias.200^2 + var.200
rbind(bias.200, var.200, mse.200)

```

EM algorithms

The class of algorithms called EM algorithms is enormously important in statistics. There are many, many

different specific algorithms that can be called EM algorithms, but they have this in common: They seek to iteratively maximize a likelihood function in a situation in which the data may be thought of as incompletely observed.

The name "EM algorithm" has its genesis in a seminal 1977 paper by Dempster, Laird, and Rubin in the Journal of the Royal Statistical Society, Series B. Many distinct algorithms published prior to 1977 were examples of EM, including the Lucy-Richardson algorithm for image deconvolution that is apparently quite well known in astronomy. The major contribution of Dempster et al was to unify these algorithms and prove certain facts about them. Interesting historical note: They even "proved" an untrue fact that was refuted six years (!) later (even thirty years ago, publications in statistics churned through the pipeline at a snail's pace).

We'll derive a simple EM algorithm on a toy example: We'll pretend that some of the gamma ray burst flux measurements were right-censored, as follows:

```
cflux <- flux
cflux[flux>=60] <- 60
n <- length(cflux)
yy <- (1:n)/n
plot(sort(cflux), log(1-yy+1/n))
```

The situation may be viewed like this: The complete dataset is a set of n observations from an exponential distribution with unknown mean μ , say, X_1, \dots, X_n . What we observe, however, is Z_1, \dots, Z_n , where Z_i is defined as $\min(X_i, 60)$. The log likelihood for the observed data is as follows:

```
m <- sum(flux>=60)
s <- sum(cflux)
loglik <- function(mu)
  -(n-m)*log(mu)-s/mu
```

As it turns out, this log likelihood function can be maximized explicitly:

```
mle <- s/(n-m)
```

However, we will construct an EM algorithm anyway for two reasons: First, it is instructive to see how the EM operates. Second, not all censoring problems admit a closed-form solution like this one does!

We start by writing down the complete-data log likelihood for the sample. This is straightforward because the complete data are simply a random sample from an exponential distribution with mean μ . Next, we pick a starting value of μ , say μ_0 . Then comes the tricky part: We take the conditional expectation of the complete data log likelihood, conditional on the observed data and assuming that μ_0 is the correct parameter. (This will have to happen on the chalkboard!) The result is a function of **both** μ and μ_0 , and construction of this function is called the E (expectation) step. Next, we maximize this function over μ . The result of the maximization becomes our next iterate, μ_1 , and the process repeats.

Let's start with $\mu_0=20$. Carrying out the calculations described above yields the following iterative scheme:

```
mu <- 20
loglik(mu)
mu <- s/n + m*mu/n; loglik(mu)
mu <- s/n + m*mu/n; loglik(mu)
# repeat the last line a few times
```

Notice that the value of the (observed data) log likelihood increases at each iteration. This is the fundamental property of any EM algorithm! In fact, it is very helpful when debugging computer code, since there must be a bug somewhere whenever the log likelihood is ever observed to decrease. Notice also that the value of μ has converged to the true maximum likelihood estimator after a few iterations.

An EM algorithm for a mixture: A simple case

Let's try a two-component mixture of normals model on the quasar absorption line dataset that Tom showed in his lecture:

```
qso <- scan("http://www.astrostatistics.psu.edu/datasets/QSO_absorb.txt",
skip=1, nlines=104)[2*(1:104)]
```

As a Ph.D. student, I developed the R package 'mixtools', which consists of various functions for analyzing mixture models. The functions in the package originally dealt with analyzing mixtures of regressions, but has since grown to include a wide array of other mixture procedures. First, let us load the 'mixtools' package and look at the corresponding help file:

```
install.packages("mixtools", lib="V:/") # lib=... is not always
necessary!
library(mixtools, lib.loc="V:/")
help(package="mixtools")
```

Let us also look at the function to implement an EM algorithm for a mixture of normals:

```
?normalmixEM
```

Notice that if you specify 'arbmean=FALSE' or 'arbvar=FALSE', you can get a scale mixture of normals or a location mixture of normals, respectively (a scale mixture is when the component means are all the same and a location mixture is when the component variances are all the same). In addition, you can specify starting values as well as the number of components you are interested in fitting (by using the option 'k').

Let's look at a histogram of the quasar absorption line dataset:

```
hist(qso, nclass=20)
```

It appears that a location mixture of normals may be appropriate. In fact, we can perform a formal chi-square test where the null hypothesis is the component variances are the same versus the alternative that they are different. This can be accomplished by:

```
test.equality(qso, lambda=c(.9,.1), mu=c(1,.6), sigma=.001, arbvar=FALSE)
```

Thus, we proceed with fitting a scale mixture of normals:

```
out <- normalmixEM(qso, lambda=c(.9,.1), mu=c(1,.6), sigma=.001, arbvar=FALSE)
out[2:5]
hist(qso, nclass=20)
plot(out, density=TRUE)
```

This seems to converge to a sensible solution. However, try some other starting values (or let the algorithm generate them for you) and see what happens. If you find a different solution (i.e., a different local maximum), how does its log likelihood value compare to that of the first solution?

Nonparametrics.Zip (a compressed version of nonparamtrics)

Tom Hettmansperger
Department of Statistics, Penn State University

References:

1. Higgins (2004) *Intro to Modern Nonpar Stat*
2. Hollander and Wolfe (1999) *Nonpar Stat Methods*
3. Arnold Notes
4. Johnson, Morrell, and Schick (1992) Two-Sample Nonparametric Estimation and Confidence Intervals Under Truncation, *Biometrics*, **48**, 1043-1056.
5. Website: <http://www.stat.wmich.edu/slab/RGLM/>

Single Sample Methods

- Robust Data Summaries
- Graphical Displays
- Inference: Confidence Intervals and Hypothesis Tests

Location, Spread, Shape

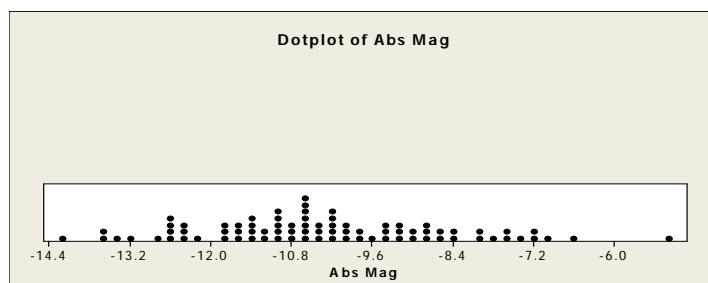
CI-Boxplots (notched boxplots)

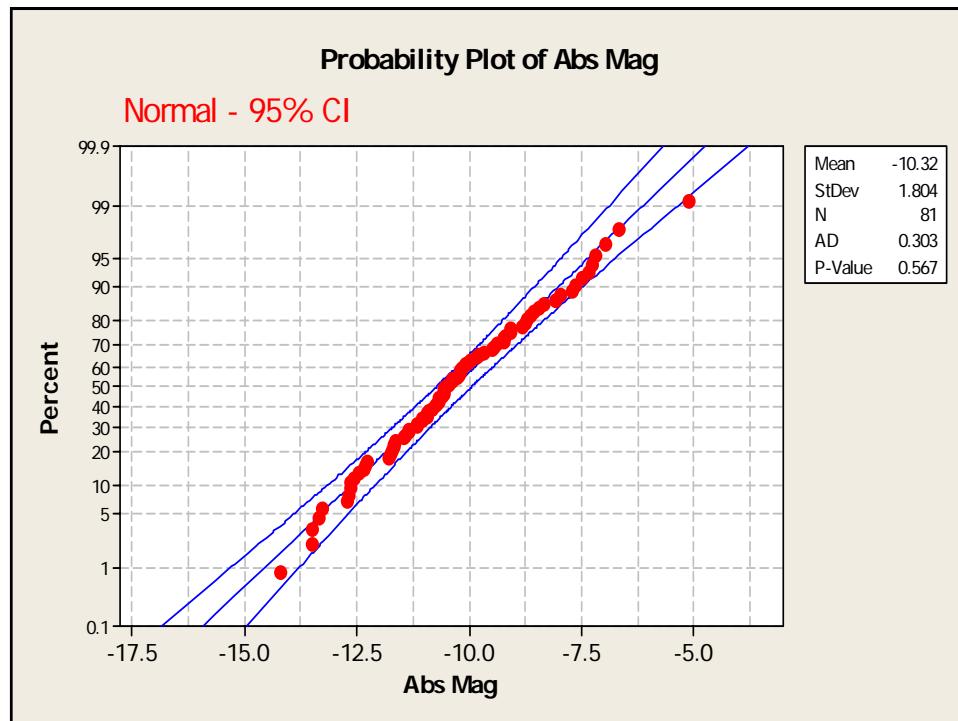
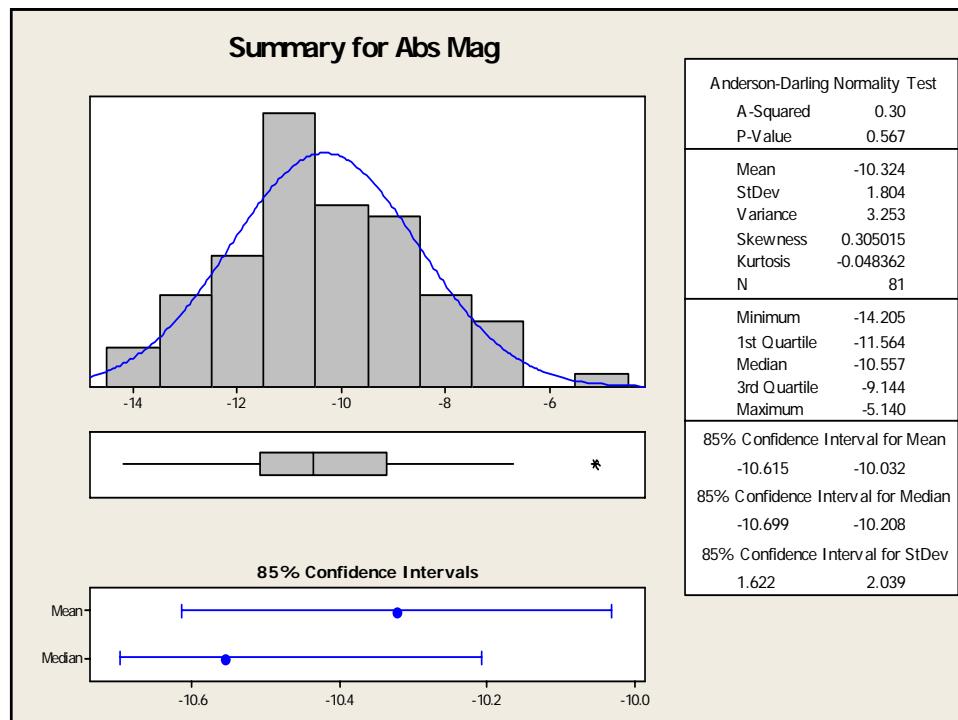
Histograms, dotplots, kernel density estimates.

Absolute Magnitude Planetary Nebulae Milky Way

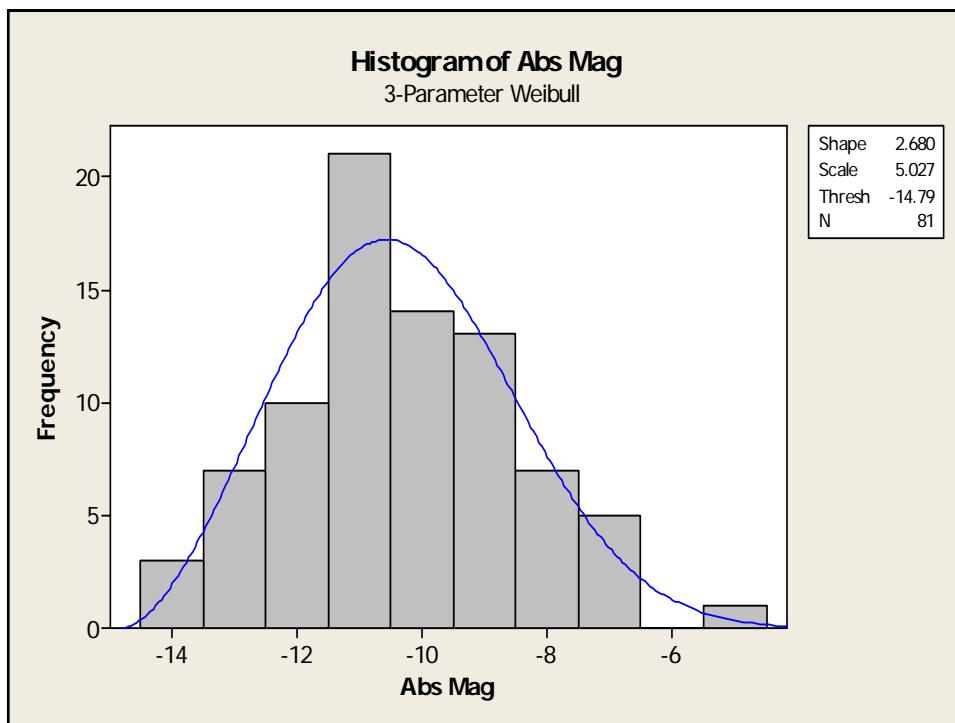
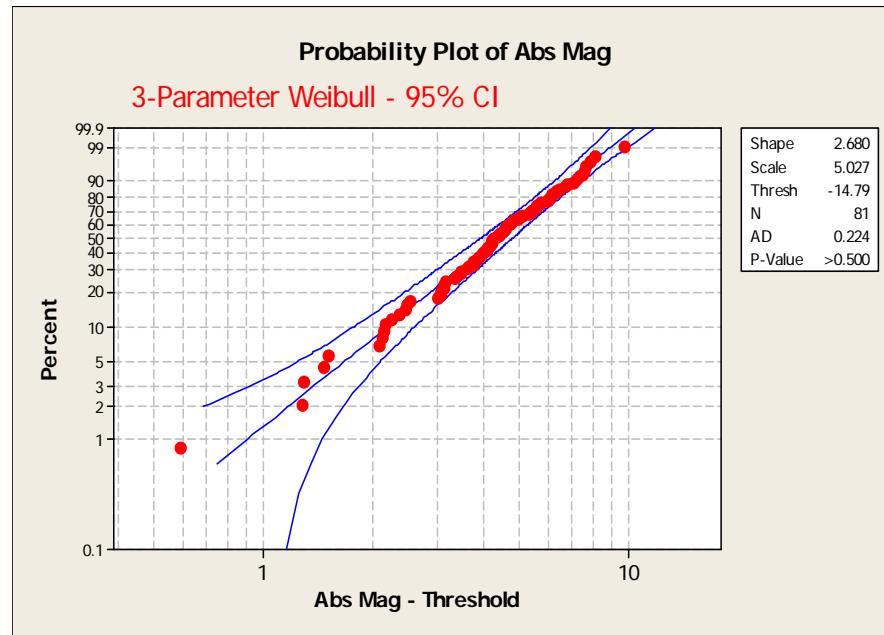
Abs Mag (n = 81)

17.537 15.845 15.449 12.710 15.499 16.450 14.695 14.878
15.350 12.909 12.873 13.278 15.591 14.550 16.078 15.438
14.741 ...





But don't be too quick to "accept" normality:



Weibull Distribution:

$$f(x) = \frac{c(x-t)^{c-1}}{b^c} \exp\left\{-\left(\frac{x-t}{b}\right)^c\right\} \text{ for } x > t \text{ and } 0 \text{ otherwise}$$

t = threshold

b = scale

c = shape

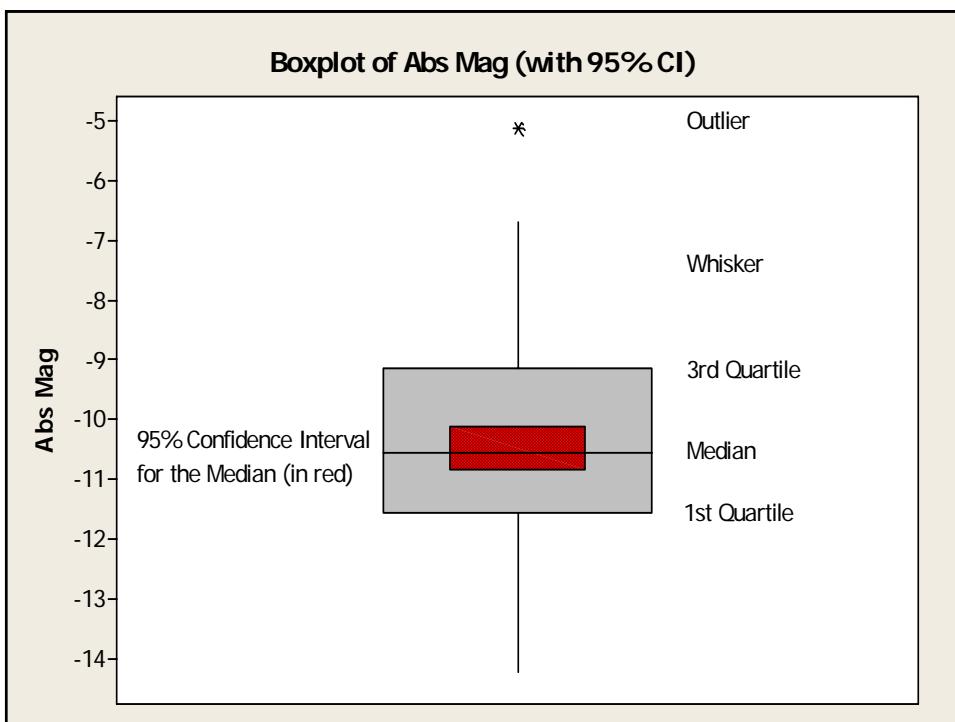
Null Hyp: Pop distribution, $F(x)$ is normal

The Kolmogorov-Smirnov Statistic

$$D = \max |F_n(x) - F(x)|$$

The Anderson-Darling Statistic

$$AD = n \int (F_n(x) - F(x))^2 [F(x)(1 - F(x))]^{-1} dF(x)$$



Anatomy of a 95% CI-Boxplot

- Box formed by quartiles and median
- IQR (interquartile range) $Q_3 - Q_1$
- Whiskers extend from the end of the box to the farthest point within $1.5 \times IQR$.

For a **normal benchmark** distribution, $IQR = 1.348\text{Stdev}$ and $1.5 \times IQR = 2\text{Stdev}$.

Outliers beyond the whiskers are more than 2.7 stdevs from the median. For a normal distribution this should happen about .7% of the time.

$$\text{Pseudo Stdev} = .75 \times IQR$$

The confidence interval and hypothesis test

A population is located at d_0 if the population median is d_0 .

Sample X_1, \dots, X_n from the population.

Say X_1, \dots, X_n is located at d if $X_1 - d, \dots, X_n - d$ is located at 0.

$S(d) = S(X_1 - d, \dots, X_n - d)$ a statistic useful for location analysis if

$E_{d_0}(S(d_0)) = 0$ when pop is located at d_0

Sign Statistic :

$$\begin{aligned} S(d) &= \sum \text{sgn}(X_i - d) = \# X_i > d - \# X_i < d \\ &= S^+(d) - S^-(d) = 2S^+(d) - n \end{aligned}$$

Estimate d_0 from data, note : $E_{d_0} S(d_0) = 0$

Find $\hat{d} \ni S(\hat{d}) = 0$ [or $S^+(\hat{d}) = n/2$]

Solution : $\hat{d} = \text{median}(X_i)$

HYPOTHESIS TEST of $H_0 : d = d_0$ vs. $H_A : d \neq d_0$

Rule: reject H_0 if $|S(d_0)| = |2S^+(d_0) - n| \geq c$

where $P_{d_0}(|2S^+(d_0) - n| \geq c) = \alpha$.

$$S^+(d_0) \leq \frac{n-c}{2} = k \text{ or } S^+(d_0) \geq \frac{n+c}{2} = n-k$$

Under $H_0 : d = d_0$,

$S^+(d_0)$ distributed Binomial($n, \frac{1}{2}$)

Distribution Free

CONFIDENCE INTERVAL

d is population location

$$P_d(k < S^+(d) < n - k) = 1 - \alpha$$

Find smallest d $\exists (\# X_i > d) < n - k$

$$d = X_{(k)} : (\# X_i > X_{(k)}) = n - k$$

$$d_{\min} = X_{(k+1)} : (\# X_i > X_{(k+1)}) = n - k - 1$$

Likewise $d_{\max} = X_{(n-k)}$

Then $[X_{(k+1)}, X_{(n-k)}]$ is $(1 - \alpha)100\%$ Conf. Int.

Distribution Free

SUMMARY :

X_1, \dots, X_n a sample from a population located at d_0 .

SIGN STATISTIC : $S(d) = S^+(d) - S^-(d) = \#X_i > d - \#X_i < d$

ESTIMATE : $\hat{d} \ni S(\hat{d}) = 0 \Rightarrow \hat{d} = \text{median}(X_i)$

TEST of $H_0: d = d_0$ vs. $H_A: d \neq d_0$ \Leftrightarrow

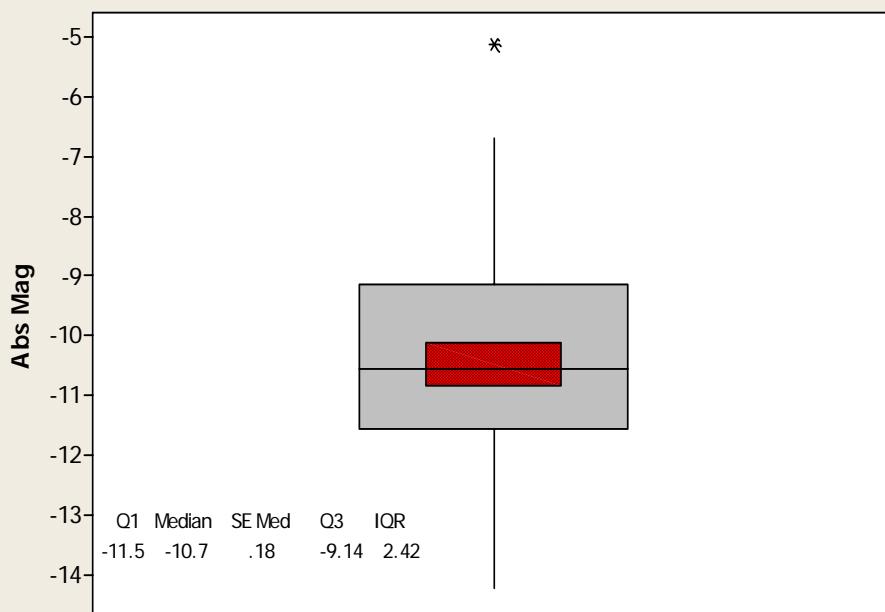
$H_0: P(X > d_0) = \frac{1}{2}$ vs. $H_A: P(X > d_0) \neq \frac{1}{2}$

reject H_0 if $S^+(d_0) \leq k$ or $\geq n - k$

where $P_{d_0}(S^+(d_0) \leq k) = \alpha/2$ and $S^+(d_0)$ binomial $(n, 1/2)$

CONFIDENCE INTERVAL : if $P_d(S^+(d) \leq k) = \alpha/2$ then
 $[X_{(k+1)}, X_{(n-k)}]$ has confidence coefficient $(1-\alpha)100\%$

Boxplot of Abs Mag (with 95% CI)



Additional Remarks:

The median is a robust measure of location. It is not affected by outliers. It is efficient when the population has heavier tails than a normal population.

The sign test is also robust and insensitive to outliers. It is efficient when the tails are heavier than those of a normal population.

Similarly for the confidence interval.

In addition, the test and the confidence interval are distribution free and do not depend on the shape of the underlying population to determine critical values or confidence coefficients.

They are only 64% efficient relative to the mean and t-test when the population is normal.

If the population is symmetric then the Wilcoxon Signed Rank statistic can be used, and it is robust against outliers and 95% efficient relative to the t-test.

Two-Sample Methods

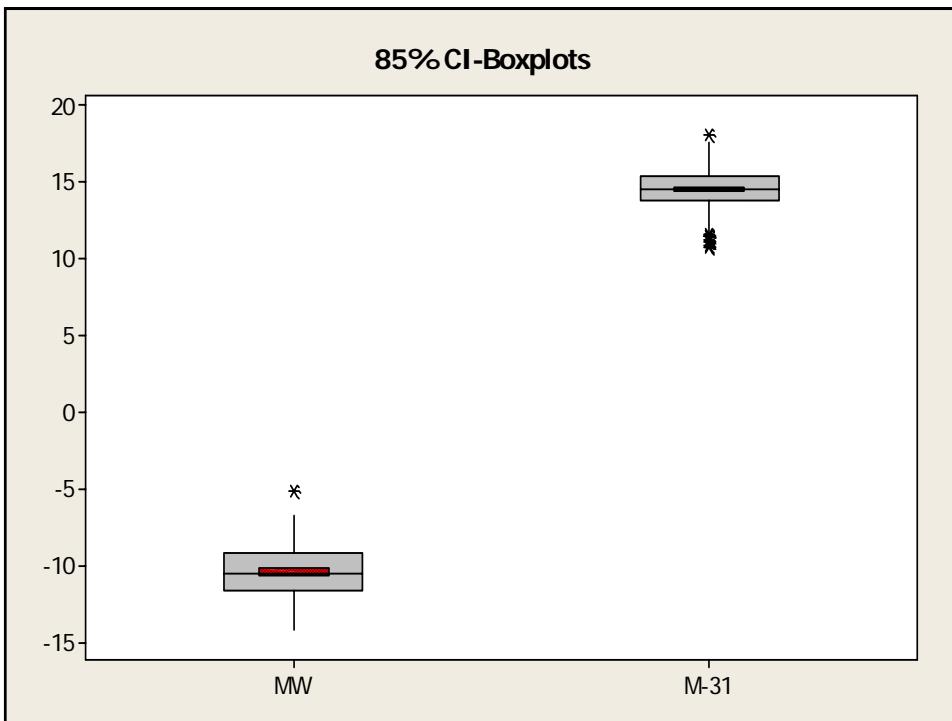
Two-Sample Comparisons

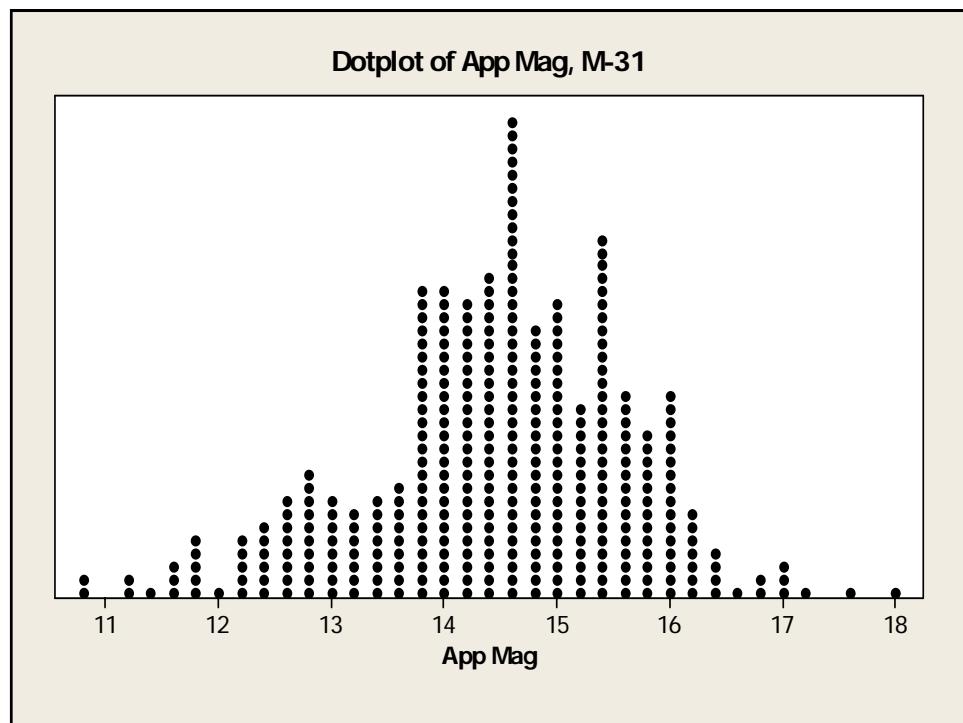
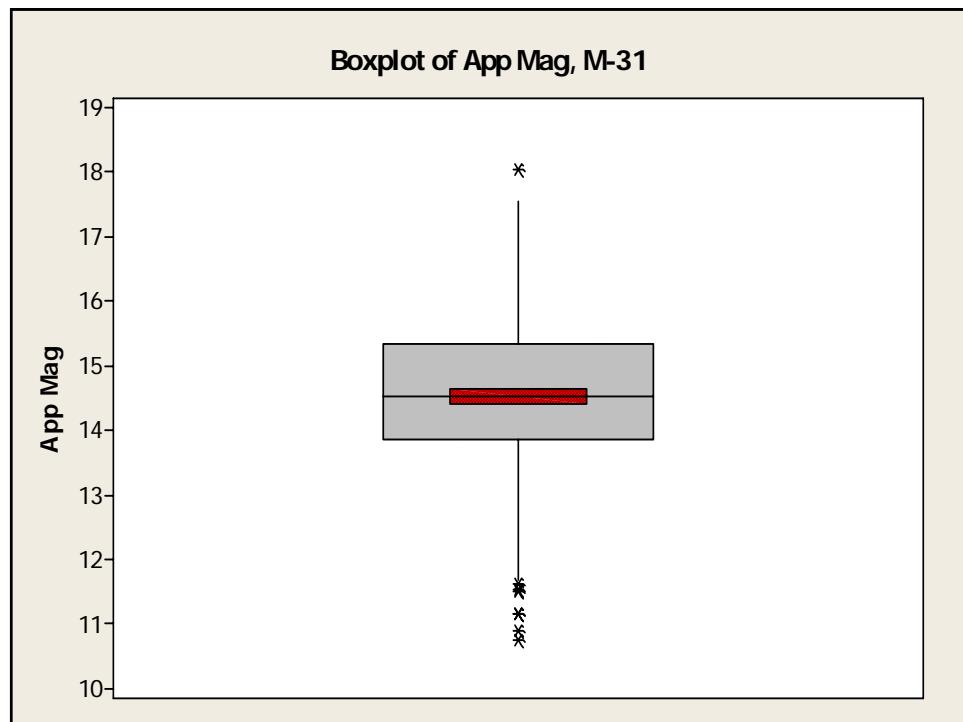
85% CI-Boxplots

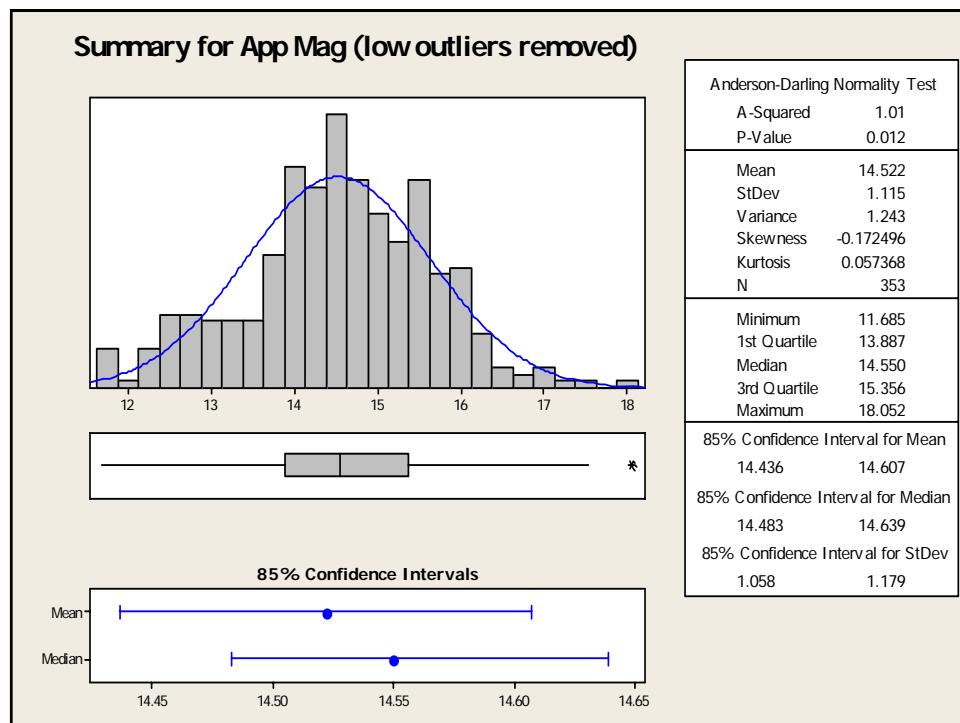
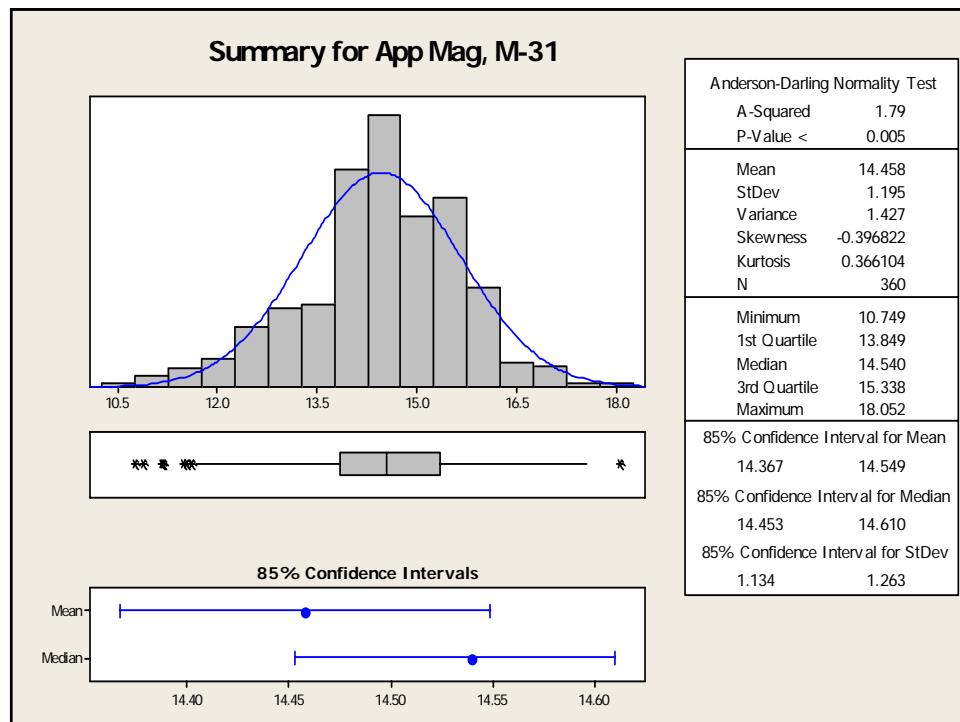
Mann-Whitney-Wilcoxon Rank Sum Statistic

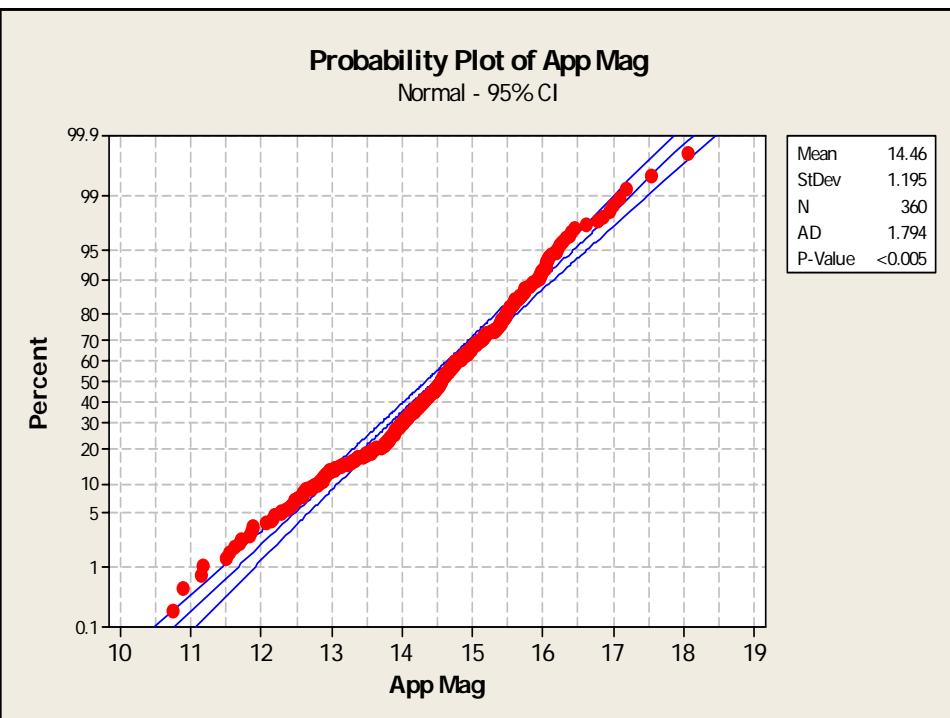
- Estimate of difference in locations
- Test of difference in locations
- Confidence Interval for difference in locations

Levene's Rank Statistic for differences in scale or variance.









Why 85% Confidence Intervals?

We have the following test of

$$H_0 : d = d_1 - d_2 = 0 \text{ vs. } H_A : d = d_1 - d_2 \neq 0$$

Rule: reject the null hyp if the 85% confidence intervals do not overlap.

The significance level is close to 5% provided the ratio of sample sizes is less than 3.

Mann-Whitney-Wilcoxon Statistic: The sign statistic on the pairwise differences.

X_1, \dots, X_m and Y_1, \dots, Y_n with X from pop F and Y from pop G with $d = d_Y - d_X$.

$$\begin{aligned} U(d) &= \sum \sum \text{sgn}(Y_i - d - X_j) = U^+(d) - U^-(d) \\ &= (\# Y_i - X_j > d) - (\# Y_i - X_j < d) \end{aligned}$$

Unlike the sign test (64% efficiency for normal population, the MWW test has 95.5% efficiency for a normal population. And it is robust against outliers in either sample.

SUMMARY :

MWW STATISTIC : $U(d) = U^+(d) - U^-(d) = \#Y_j - X_i > d - \#Y_j - X_i < d$

ESTIMATE : $\hat{d} \ni U(\hat{d}) = 0 \Rightarrow \hat{d} = \text{median}_{i,j}(Y_j - X_i)$

TEST of $H_0: d = 0$ vs. $H_A: d \neq 0 \Leftrightarrow$

$H_0: P(Y > X) = \frac{1}{2}$ vs. $H_A: P(Y > X) \neq \frac{1}{2}$

reject H_0 if $U^+(0) \leq k$ or $\geq n - k$

where $P_{d_0}(U^+(0) \leq k) = \alpha/2$ and $U^+(d_0)$ at a tabled distribution.

CONFIDENCE INTERVAL : if $P_d(U^+(d) \leq k) = \alpha/2$ then

$[D_{(k+1)}, D_{(mn-k)}]$ has confidence coefficient $(1-\alpha)100\%$
where $D_{(1)} \leq \dots \leq D_{(mn)}$ are the ordered pairwise differences.

Mann-Whitney Test and CI: App Mag, Abs Mag

	N	Median
App Mag (M-31)	360	14.540
Abs Mag (MW)	81	-10.557

Point estimate for d is 24.900

95.0 Percent CI for d is (24.530,25.256)

$$W = 94140.0$$

Test of d=0 vs d not equal 0 is significant at 0.0000

What is W?

$$U^+ = \# Y_j > X_i$$

$$W = U^+ + \frac{n(n+1)}{2} = \sum_{j=1}^n R_j$$

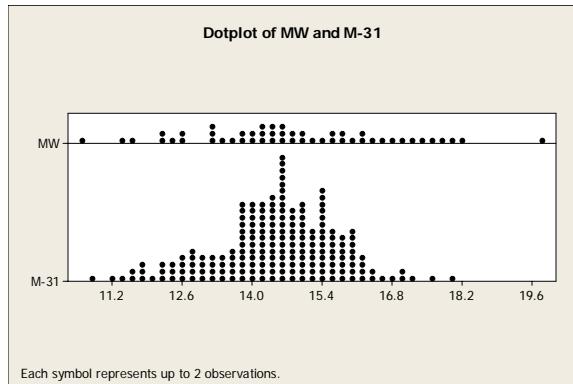
R_1, \dots, R_n are ranks of Y_1, \dots, Y_n in combined data

$$\bar{R}_Y - \bar{R}_X = \left(\frac{1}{n} + \frac{1}{m} \right) U^+ - \frac{n+m}{2}$$

Hence MWW can be written as the difference in average ranks rather than $\bar{Y} - \bar{X}$ in t-test.

What about spread or scale differences between the two populations?

Below we shift the MW observations to the right by 24.9 to line up with M-31.



Variable	StDev	IQR	PseudoStdev
MW	1.804	2.420	1.815
M-31	1.195	1.489	1.117

Levene's Rank Test

Compute $|Y - \text{Med}(Y)|$ and $|X - \text{Med}(X)|$, called absolute deviations.

Apply MWW to the absolute deviations. (Rank the absolute deviations)

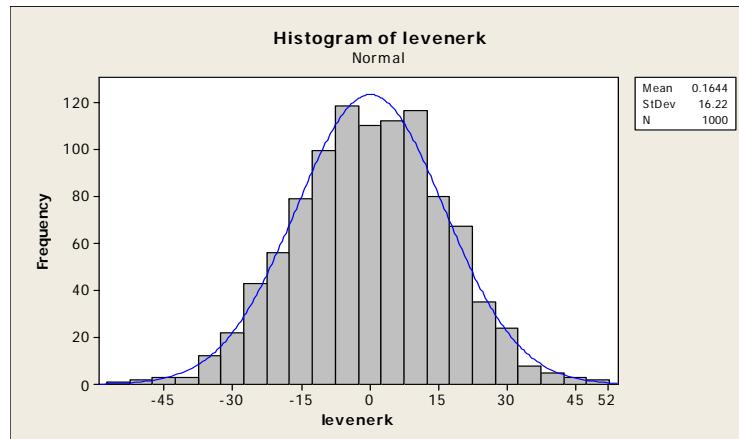
The test rejects equal spreads in the two populations when difference in average ranks of the absolute deviations is too large.

Idea: After we have centered the data, then if the null hypothesis of no difference in spreads is true, all permutations of the combined data are roughly equally likely. (Permutation Principle)

So randomly select a large set of the permutations say B permutations. Assign the first n to the Y sample and the remaining m to the X sample and compute MMW on the absolute deviations.

The approximate p-value is #MMW > original MMW divided by B.

Difference of rank mean abso devs 51.9793



So we easily reject the null hypothesis of no difference in spreads and conclude that the two populations have significantly different spreads.

Several Sample Methods

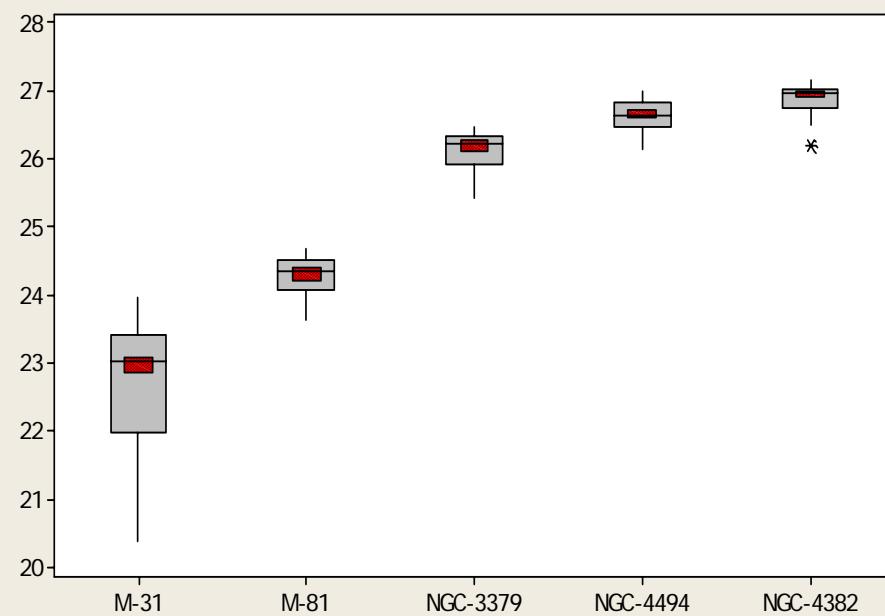
Variable	Mean	StDev	Median	.75IQR	Skew	Kurtosis
Messier 31	22.685	0.969	23.028	1.069	-0.67	-0.67
Messier 81	24.298	0.274	24.371	0.336	-0.49	-0.68
NGC 3379	26.139	0.267	26.230	0.317	-0.64	-0.48
NGC 4494	26.654	0.225	26.659	0.252	-0.36	-0.55
NGC 4382	26.905	0.201	26.974	0.208	-1.06	1.08

All one-sample and two-sample methods can be applied one at a time or two at a time. Plots, summaries, inferences.

We begin k-sample methods by asking if the location differences between the NGC nebulae are statistically significant.

We will briefly discuss issues of truncation.

85% CI-Boxplot Planetray Nebula Luminosities



Extending MWW to several samples

*Given $N = \text{total sample size with ranks of combined data}$
with $\bar{R}_1, \bar{R}_2,$ and \bar{R}_3 construct :*

$$\begin{aligned} KW &= \frac{12}{N(N+1)} \left\{ \frac{n_1 n_2}{N} (\bar{R}_1 - \bar{R}_2)^2 + \frac{n_1 n_3}{N} (\bar{R}_1 - \bar{R}_3)^2 + \frac{n_2 n_3}{N} (\bar{R}_2 - \bar{R}_3)^2 \right\} \\ &= \frac{12}{N(N+1)} \left\{ n_1 \left(\bar{R}_1 - \frac{N+1}{2} \right)^2 + n_2 \left(\bar{R}_2 - \frac{N+1}{2} \right)^2 + n_3 \left(\bar{R}_3 - \frac{N+1}{2} \right)^2 \right\} \end{aligned}$$

*Generally use a chisquare ($k-1=2$) Degrees of Freedom as
approximate sampling distribution for KW*

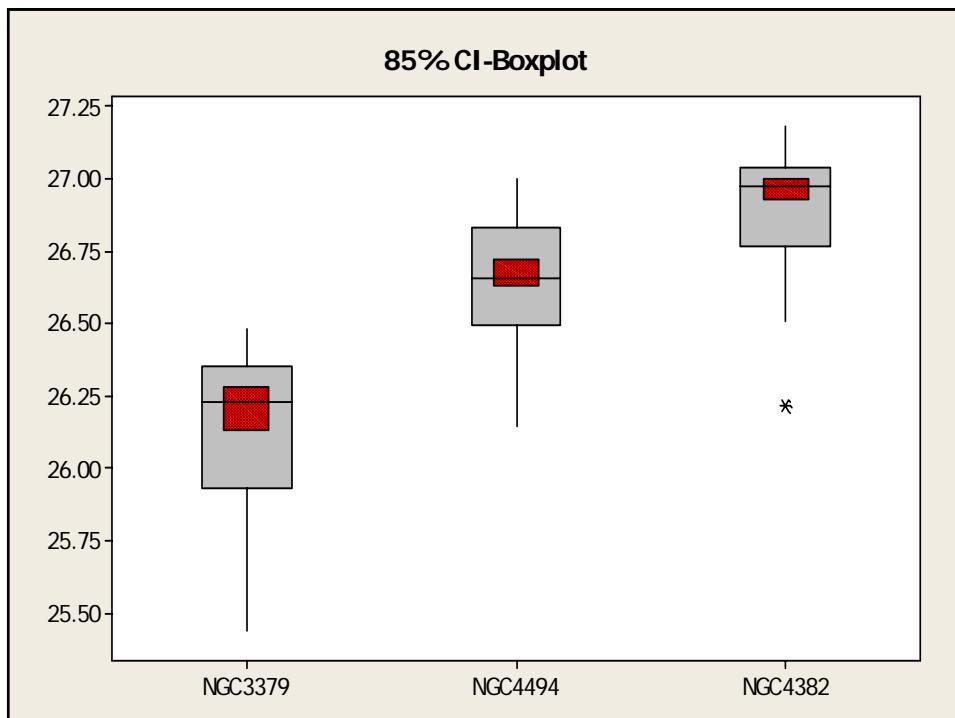
Kruskal-Wallis Test on NGC

sub	N	Median	Ave Rank	Z
1	45	26.23	29.6	-9.39
2	101	26.66	104.5	0.36
3	59	26.97	156.4	8.19
Overall	205		103.0	

KW = 116.70 DF = 2 P = 0.000

This test can be followed by **multiple comparisons**.

For example, if we assign a family error rate of .09, then we would conduct 3 MWW tests, each at a level of .03. (Bonferroni)



What to do about truncation.

1. See a statistician
2. Read the Johnson, Morrell, and Schick reference. and then see a statistician.

Here is the problem: Suppose we want to estimate the difference in locations between two populations: $F(x)$ and $G(y) = F(y - d)$.

But (with right truncation at a) the observations come from

$$F_a(x) = \frac{F(x)}{F(a)} \text{ for } x \leq a \text{ and } 1 \text{ for } x > a$$

$$G_a(y) = \frac{F(y-d)}{F(a-d)} \text{ for } y \leq a \text{ and } 1 \text{ for } y > a$$

Suppose $d > 0$ and so we want to shift the X-sample to the right toward the truncation point. As we shift the Xs, some will pass the truncation point and will be eliminated from the data set. This changes the sample sizes and requires adjustment when computing the corresponding MWU to see if it is equal to its expectation. See the reference for details.

Comparison of NGC4382 and NGC 4494

Data multiplied by 100 and 2600 subtracted.
Truncation point taken as 120.

Point estimate for d is 25.30 W = 6595.5

m = 101 and n = 59

Computation of shift estimate with truncation					
d	m	n	\hat{d}	W	E(W)
25.3	88	59	5.10	4750.5	4366.0
28.3	84	59	3.60	4533.5	4248.0
30.3	83	59	2.10	4372.0	4218.5
32.3	81	59	0.80	4224.5	4159.5
33.3	81	59	-0.20	4144.5	4159.5
33.1	81	59	-0.00	4161.5	4159.5

What more can we do?

1. Multiple regression
2. Analysis of designed experiments (AOV)
3. Analysis of covariance
4. Multivariate analysis

These analyses can be carried out using the website:

<http://www.stat.wmich.edu/slab/RGLM/>

Introduction to Bayesian Inference

Tom Loredo

Dept. of Astronomy, Cornell University

<http://www.astro.cornell.edu/staff/loredo/bayes/>

CAS Summer School — June 11, 2008

1 / 103

Outline

- ① The Big Picture
- ② Foundations—Logic & Probability Theory
- ③ Inference With Parametric Models
 - Parameter Estimation
 - Model Uncertainty
- ④ Simple Examples
 - Binary Outcomes
 - Normal Distribution
 - Poisson Distribution
- ⑤ Bayesian Computation
- ⑥ Measurement Error Applications
 - Number counts (flux distributions)
 - Coincidence assessment/cross-matching
- ⑦ Probability & Frequency
- ⑧ Outlook: Hotspots

2 / 103

Outline

- ① The Big Picture
- ② Foundations—Logic & Probability Theory
- ③ Inference With Parametric Models
 - Parameter Estimation
 - Model Uncertainty
- ④ Simple Examples
 - Binary Outcomes
 - Normal Distribution
 - Poisson Distribution
- ⑤ Bayesian Computation
- ⑥ Measurement Error Applications
 - Number counts (flux distributions)
 - Coincidence assessment/cross-matching
- ⑦ Probability & Frequency
- ⑧ Outlook: Hotspots

3 / 103

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 ≡ 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 constructs and appraises arguments that reason from data to interesting scientific conclusions (explanations, predictions)

4 / 103

The Role of Data

Data do not speak for themselves!

We don't just *tabulate* data, we *analyze* data.

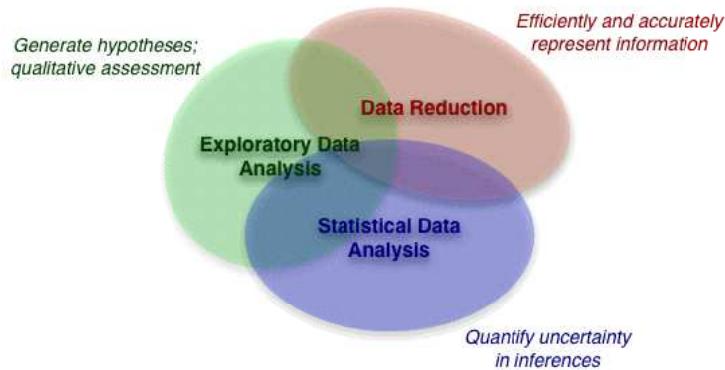
We gather data so they may speak for or against existing hypotheses, and guide the formation of new hypotheses.

A key role of data in science is to be among the premises in scientific arguments.

5 / 103

Data Analysis

Building & Appraising Arguments Using Data



Statistical inference is but one of several interacting modes of analyzing data.

6 / 103

Bayesian Statistical Inference

- A different approach to *all* statistical inference problems (i.e., not just another method in the list: BLUE, maximum likelihood, χ^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 / 103

Outline

- ① The Big Picture
- ② Foundations—Logic & Probability Theory
- ③ Inference With Parametric Models
 - Parameter Estimation
 - Model Uncertainty
- ④ Simple Examples
 - Binary Outcomes
 - Normal Distribution
 - Poisson Distribution
- ⑤ Bayesian Computation
- ⑥ Measurement Error Applications
 - Number counts (flux distributions)
 - Coincidence assessment/cross-matching
- ⑦ Probability & Frequency
- ⑧ Outlook: Hotspots

8 / 103

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 Λ CDM
 A : $\Omega_{\text{tot}} \in [0.9, 1.1]$
 B : Ω_Λ is not 0
 \bar{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

9 / 103

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*

10 / 103

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).

We want to make *sound* arguments. Formal logic and probability theory address validity, but there is no formal approach for addressing factual correctness → there is always a subjective element to an argument.

11 / 103

Factual Correctness

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*

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*

12 / 103

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

13 / 103

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.

14 / 103

Real Number Representation of Induction

$P(H|\mathcal{P}) \equiv$ strength of argument $H|\mathcal{P}$

- $P = 0 \rightarrow$ Argument is *invalid*
- $= 1 \rightarrow$ Argument is *valid*
- $\in (0, 1) \rightarrow$ Degree of deducibility

A mathematical model for induction:

$$\begin{aligned} \text{'AND' (product rule): } P(A \wedge B|\mathcal{P}) &= P(A|\mathcal{P}) P(B|A \wedge \mathcal{P}) \\ &= P(B|\mathcal{P}) P(A|B \wedge \mathcal{P}) \end{aligned}$$

$$\begin{aligned} \text{'OR' (sum rule): } P(A \vee B|\mathcal{P}) &= P(A|\mathcal{P}) + P(B|\mathcal{P}) \\ &\quad - P(A \wedge B|\mathcal{P}) \end{aligned}$$

$$\text{'NOT': } P(\bar{A}|\mathcal{P}) = 1 - P(A|\mathcal{P})$$

Bayesian inference explores the implications of this model.

15 / 103

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*

16 / 103

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

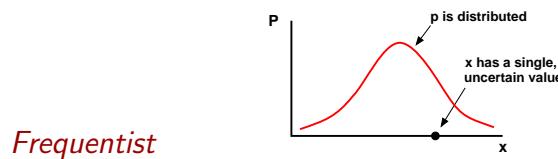
<i>Intuitive notion</i>	<i>Quantification</i>	<i>Calibration</i>
Hot, cold	Temperature, T	Cold as ice = 273K Boiling hot = 373K
uncertainty	Probability, P	Certainty = 0, 1 $p = 1/36$: plausible as “snake’s eyes” $p = 1/1024$: plausible as 10 heads

17 / 103

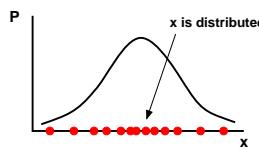
A Bit More On Interpretation

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:



Probabilities are always (limiting) rates/proportions/frequencies in an ensemble. $p(x)$ describes variability, how the **values of x** are distributed among the cases in the ensemble:



18 / 103

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_{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.

19 / 103

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:

$$\begin{aligned} \text{'OR' (sum rule): } P(H_1 \vee H_2|I) &= P(H_1|I) + P(H_2|I) \\ &\quad - P(H_1, H_2|I) \end{aligned}$$

$$\begin{aligned} \text{'AND' (product rule): } P(H_1, D|I) &= P(H_1|I) P(D|H_1, I) \\ &= P(D|I) P(H_1|D, I) \end{aligned}$$

$$\text{'NOT': } P(\overline{H_1}|I) = 1 - P(H_1|I)$$

20 / 103

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:

$$\begin{aligned} \text{posterior} &\propto \text{prior} \times \text{likelihood} \\ \text{norm. const. } P(D_{\text{obs}}|I) &= \text{prior predictive} \end{aligned}$$

21 / 103

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*:

$$P(A|I) = \sum_i P(A, B_i|I)$$

22 / 103

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_{obs} = Average likelihood for H_i
(a.k.a. *marginal likelihood*)

Normalization

For *exclusive, exhaustive* H_i ,

$$\sum_i P(H_i|\dots) = 1$$

23 / 103

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").

24 / 103

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: Make your appraisal account for all of your premises.
- LTP: If the premises allow multiple arguments for a hypothesis, its appraisal must account for all of them.

25 / 103

Outline

- ① The Big Picture
- ② Foundations—Logic & Probability Theory
- ③ Inference With Parametric Models
 - Parameter Estimation
 - Model Uncertainty
- ④ Simple Examples
 - Binary Outcomes
 - Normal Distribution
 - Poisson Distribution
- ⑤ Bayesian Computation
- ⑥ Measurement Error Applications
 - Number counts (flux distributions)
 - Coincidence assessment/cross-matching
- ⑦ Probability & Frequency
- ⑧ Outlook: Hotspots

26 / 103

Inference With Parametric Models

Models M_i ($i = 1$ to N), each with parameters θ_i , each imply a *sampling dist'n* (conditional predictive dist'n for possible data):

$$p(D|\theta_i, M_i)$$

The θ_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 θ_i (parameter uncertainty).

27 / 103

Three Classes of Problems

Parameter Estimation

Premise = choice of model (pick specific i)
 → What can we say about θ_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 ϕ w/o committing to one model?
 (Systematic error is an example)

28 / 103

Parameter Estimation

Problem statement

I = Model M with parameters θ (+ any add'l info)

H_i = statements about θ ; 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 θ :

$$\begin{aligned} P(\theta \in [\theta, \theta + d\theta] | \dots) &= f(\theta)d\theta \\ &= p(\theta | \dots)d\theta \end{aligned}$$

Posterior probability density

$$p(\theta | D, M) = \frac{p(\theta | M) \mathcal{L}(\theta)}{\int d\theta p(\theta | M) \mathcal{L}(\theta)}$$

29 / 103

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 Δ 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 ψ , nuisance parameters ϕ
 - Marginal dist'n for ψ : $p(\psi | D, M) = \int d\phi p(\psi, \phi | D, M)$

30 / 103

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 ?

31 / 103

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.

32 / 103

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

33 / 103

Odds and Bayes factors

A ratio of probabilities for two propositions using the same premises is called the *odds* favoring one over the other:

$$\begin{aligned} O_{ij} &\equiv \frac{p(M_i|D, I)}{p(M_j|D, I)} \\ &= \frac{p(M_i|I)}{p(M_j|I)} \times \frac{p(D|M_j, I)}{p(D|M_i, I)} \end{aligned}$$

The data-dependent part is called the *Bayes factor*:

$$B_{ij} \equiv \frac{p(D|M_j, I)}{p(D|M_i, I)}$$

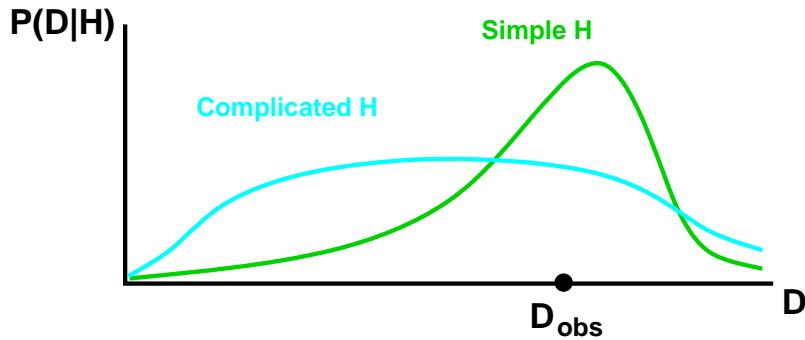
It is a *likelihood ratio*; the BF terminology is usually reserved for cases when the likelihoods are marginal/average likelihoods.

34 / 103

An Automatic Occam's Razor

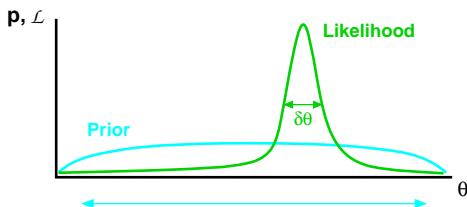
Predictive probabilities can favor simpler models

$$p(D|M_i) = \int d\theta_i p(\theta_i|M) \mathcal{L}(\theta_i)$$



35 / 103

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

36 / 103

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, ϕ

Each has different set of nuisance parameters η_i (or different prior info about them)

H_i = statements about ϕ

Model averaging

Calculate posterior PDF for ϕ :

$$\begin{aligned} p(\phi|D, I) &= \sum_i p(M_i|D, I) p(\phi|D, M_i) \\ &\propto \sum_i \mathcal{L}(M_i) \int d\eta_i p(\phi, \eta_i|D, M_i) \end{aligned}$$

The model choice is a (discrete) nuisance parameter here.

37 / 103

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”).

38 / 103

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.

39 / 103

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

40 / 103

Outline

- ① The Big Picture
- ② Foundations—Logic & Probability Theory
- ③ Inference With Parametric Models
 - Parameter Estimation
 - Model Uncertainty
- ④ Simple Examples
 - Binary Outcomes
 - Normal Distribution
 - Poisson Distribution
- ⑤ Bayesian Computation
- ⑥ Measurement Error Applications
 - Number counts (flux distributions)
 - Coincidence assessment/cross-matching
- ⑦ Probability & Frequency
- ⑧ Outlook: Hotspots

41 / 103

Binary Outcomes: Parameter Estimation

M = Existence of two outcomes, S and F ; each trial has same probability for S or F

H_i = Statements about α , the probability for success on the next trial → seek $p(\alpha|D, M)$

D = Sequence of results from N observed trials:

FFSSSFSSFS ($n = 8$ successes in $N = 12$ trials)

Likelihood:

$$\begin{aligned}
 p(D|\alpha, M) &= p(\text{failure}|\alpha, M) \times p(\text{success}|\alpha, M) \times \dots \\
 &= \alpha^n (1-\alpha)^{N-n} \\
 &= \mathcal{L}(\alpha)
 \end{aligned}$$

42 / 103

Prior

Starting with no information about α beyond its definition, use as an “uninformative” prior $p(\alpha|M) = 1$. Justifications:

- Intuition: Don’t prefer any α interval to any other of same size
- Bayes’s justification: “Ignorance” means that before doing the N trials, we have no preference for how many will be successes:

$$P(n \text{ success}|M) = \frac{1}{N+1} \quad \rightarrow \quad p(\alpha|M) = 1$$

Consider this a *convention*—an assumption added to M to make the problem well posed.

43 / 103

Prior Predictive

$$\begin{aligned} p(D|M) &= \int d\alpha \alpha^n (1-\alpha)^{N-n} \\ &= B(n+1, N-n+1) = \frac{n!(N-n)!}{(N+1)!} \end{aligned}$$

A *Beta integral*, $B(a, b) \equiv \int dx x^{a-1} (1-x)^{b-1} = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$.

44 / 103

Posterior

$$p(\alpha|D, M) = \frac{(N+1)!}{n!(N-n)!} \alpha^n (1-\alpha)^{N-n}$$

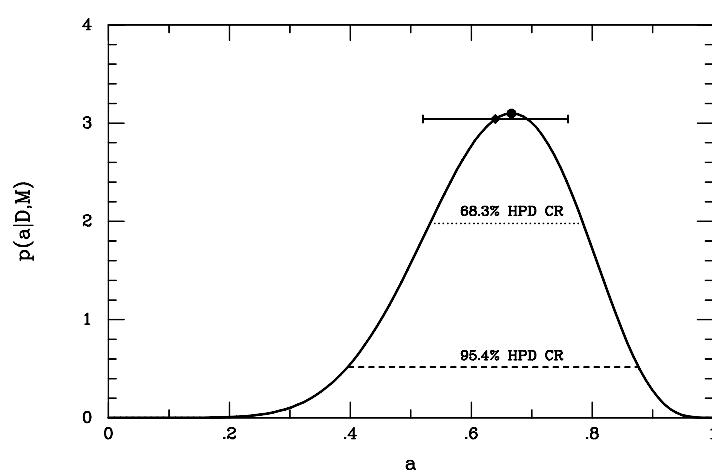
A *Beta distribution*. Summaries:

- Best-fit: $\hat{\alpha} = \frac{n}{N} = 2/3$; $\langle \alpha \rangle = \frac{n+1}{N+2} \approx 0.64$
- Uncertainty: $\sigma_\alpha = \sqrt{\frac{(n+1)(N-n+1)}{(N+2)^2(N+3)}} \approx 0.12$
Find credible regions numerically, or with incomplete beta function

Note that the posterior depends on the data only through n , not the N binary numbers describing the sequence.

n is a (minimal) *Sufficient Statistic*.

45 / 103



46 / 103

Binary Outcomes: Model Comparison

Equal Probabilities?

$$M_1: \alpha = 1/2$$

$M_2: \alpha \in [0, 1]$ with flat prior.

Maximum Likelihoods

$$M_1 : p(D|M_1) = \frac{1}{2^N} = 2.44 \times 10^{-4}$$

$$M_2 : \mathcal{L}(\hat{\alpha}) = \left(\frac{2}{3}\right)^n \left(\frac{1}{3}\right)^{N-n} = 4.82 \times 10^{-4}$$

$$\frac{p(D|M_1)}{p(D|\hat{\alpha}, M_2)} = 0.51$$

Maximum likelihoods favor M_2 (failures more probable).

47 / 103

Bayes Factor (ratio of model likelihoods)

$$p(D|M_1) = \frac{1}{2^N}; \quad \text{and} \quad p(D|M_2) = \frac{n!(N-n)!}{(N+1)!}$$

$$\begin{aligned} \rightarrow B_{12} \equiv \frac{p(D|M_1)}{p(D|M_2)} &= \frac{(N+1)!}{n!(N-n)!2^N} \\ &= 1.57 \end{aligned}$$

Bayes factor (odds) favors M_1 (equiprobable).

Note that for $n = 6$, $B_{12} = 2.93$; for this small amount of data, we can never be very sure results are equiprobable.

If $n = 0$, $B_{12} \approx 1/315$; if $n = 2$, $B_{12} \approx 1/4.8$; for extreme data, 12 flips can be enough to lead us to strongly suspect outcomes have different probabilities.

(Frequentist significance tests can reject null for any sample size.)

48 / 103

Binary Outcomes: Binomial Distribution

Suppose $D = n$ (number of heads in N trials), rather than the actual sequence. What is $p(\alpha|n, M)$?

Likelihood

Let \mathcal{S} = a sequence of flips with n heads.

$$\begin{aligned} p(n|\alpha, M) &= \sum_{\mathcal{S}} p(\mathcal{S}|\alpha, M) p(n|\mathcal{S}, \alpha, M) \\ &= \alpha^n (1-\alpha)^{N-n} C_{n,N} \end{aligned}$$

$\alpha^n (1-\alpha)^{N-n}$
[# successes = n]

$C_{n,N}$ = # of sequences of length N with n heads.

$$\rightarrow p(n|\alpha, M) = \frac{N!}{n!(N-n)!} \alpha^n (1-\alpha)^{N-n}$$

The *binomial distribution* for n given α, N .

49 / 103

Posterior

$$p(\alpha|n, M) = \frac{\frac{N!}{n!(N-n)!} \alpha^n (1-\alpha)^{N-n}}{p(n|M)}$$

$$\begin{aligned} p(n|M) &= \frac{N!}{n!(N-n)!} \int d\alpha \alpha^n (1-\alpha)^{N-n} \\ &= \frac{1}{N+1} \end{aligned}$$

$$\rightarrow p(\alpha|n, M) = \frac{(N+1)!}{n!(N-n)!} \alpha^n (1-\alpha)^{N-n}$$

Same result as when data specified the actual sequence.

50 / 103

Another Variation: Negative Binomial

Suppose $D = N$, the number of trials it took to obtain a predefined number of successes, $n = 8$. What is $p(\alpha|N, M)$?

Likelihood

$p(N|\alpha, M)$ is probability for $n - 1$ successes in $N - 1$ trials, times probability that the final trial is a success:

$$\begin{aligned} p(N|\alpha, M) &= \frac{(N-1)!}{(n-1)!(N-n)!} \alpha^{n-1} (1-\alpha)^{N-n} \alpha \\ &= \frac{(N-1)!}{(n-1)!(N-n)!} \alpha^n (1-\alpha)^{N-n} \end{aligned}$$

The *negative binomial distribution* for N given α, n .

51 / 103

Posterior

$$p(\alpha|D, M) = C'_{n,N} \frac{\alpha^n (1-\alpha)^{N-n}}{p(D|M)}$$

$$p(D|M) = C'_{n,N} \int d\alpha \alpha^n (1-\alpha)^{N-n}$$

$$\rightarrow p(\alpha|D, M) = \frac{(N+1)!}{n!(N-n)!} \alpha^n (1-\alpha)^{N-n}$$

Same result as other cases.

52 / 103

Final Variation: Meteorological Stopping

Suppose $D = (N, n)$, the number of samples and number of successes in an observing run whose total number was determined by the weather at the telescope. What is $p(\alpha|D, M')$?

(M' adds info about weather to M .)

Likelihood

$p(D|\alpha, M')$ is the binomial distribution times the probability that the weather allowed N samples, $W(N)$:

$$p(D|\alpha, M') = W(N) \frac{N!}{n!(N-n)!} \alpha^n (1-\alpha)^{N-n}$$

Let $C_{n,N} = W(N) \binom{N}{n}$. We get the *same result* as before!

53 / 103

Likelihood Principle

To define $\mathcal{L}(H_i) = p(D_{\text{obs}}|H_i, I)$, we must contemplate what other data we might have obtained. But the “real” sample space may be determined by many complicated, seemingly irrelevant factors; it may not be well-specified at all. Should this concern us?

Likelihood principle: The result of inferences depends only on how $p(D_{\text{obs}}|H_i, I)$ varies w.r.t. hypotheses. We can ignore aspects of the observing/sampling procedure that do not affect this dependence.

This is a sensible property that frequentist methods do not share. Frequentist probabilities are “long run” rates of performance, and depend on details of the sample space that are irrelevant in a Bayesian calculation.

Example: Predict 10% of sample is Type A; observe $n_A = 5$ for $N = 96$
Significance test *accepts* $\alpha = 0.1$ for binomial sampling;

$$p(>\chi^2|\alpha = 0.1) = 0.12$$

Significance test *rejects* $\alpha = 0.1$ for negative binomial sampling;

$$p(>\chi^2|\alpha = 0.1) = 0.03$$

54 / 103

Inference With Normals/Gaussians

Gaussian PDF

$$p(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad \text{over } [-\infty, \infty]$$

Common abbreviated notation: $x \sim N(\mu, \sigma^2)$

Parameters

$$\begin{aligned}\mu &= \langle x \rangle \equiv \int dx \times p(x|\mu, \sigma) \\ \sigma^2 &= \langle (x - \mu)^2 \rangle \equiv \int dx (x - \mu)^2 p(x|\mu, \sigma)\end{aligned}$$

55 / 103

Gauss's Observation: Sufficiency

Suppose our data consist of N measurements, $d_i = \mu + \epsilon_i$.
 Suppose the noise contributions are independent, and
 $\epsilon_i \sim N(0, \sigma^2)$.

$$\begin{aligned}p(D|\mu, \sigma, M) &= \prod_i p(d_i|\mu, \sigma, M) \\ &= \prod_i p(\epsilon_i = d_i - \mu|\mu, \sigma, M) \\ &= \prod_i \frac{1}{\sigma\sqrt{2\pi}} \exp \left[-\frac{(d_i - \mu)^2}{2\sigma^2} \right] \\ &= \frac{1}{\sigma^N (2\pi)^{N/2}} e^{-Q(\mu)/2\sigma^2}\end{aligned}$$

56 / 103

Find dependence of Q on μ by completing the square:

$$\begin{aligned} Q &= \sum_i (d_i - \mu)^2 \\ &= \sum_i d_i^2 + N\mu^2 - 2N\mu\bar{d} \quad \text{where } \bar{d} \equiv \frac{1}{N} \sum_i d_i \\ &= N(\mu - \bar{d})^2 + Nr^2 \quad \text{where } r^2 \equiv \frac{1}{N} \sum_i (d_i - \bar{d})^2 \end{aligned}$$

Likelihood depends on $\{d_i\}$ only through \bar{d} and r :

$$\mathcal{L}(\mu, \sigma) = \frac{1}{\sigma^N (2\pi)^{N/2}} \exp\left(-\frac{Nr^2}{2\sigma^2}\right) \exp\left(-\frac{N(\mu - \bar{d})^2}{2\sigma^2}\right)$$

The sample mean and variance are *sufficient statistics*.

This is a miraculous compression of information—the normal dist'n is highly *abnormal* in this respect!

57 / 103

Estimating a Normal Mean

Problem specification

Model: $d_i = \mu + \epsilon_i$, $\epsilon_i \sim N(0, \sigma^2)$, σ is known $\rightarrow I = (\sigma, M)$.

Parameter space: μ ; seek $p(\mu|D, \sigma, M)$

Likelihood

$$\begin{aligned} p(D|\mu, \sigma, M) &= \frac{1}{\sigma^N (2\pi)^{N/2}} \exp\left(-\frac{Nr^2}{2\sigma^2}\right) \exp\left(-\frac{N(\mu - \bar{d})^2}{2\sigma^2}\right) \\ &\propto \exp\left(-\frac{N(\mu - \bar{d})^2}{2\sigma^2}\right) \end{aligned}$$

58 / 103

"Uninformative" prior

Translation invariance $\Rightarrow p(\mu) \propto C$, a constant.
 This prior is *improper* unless bounded.

Prior predictive/normalization

$$\begin{aligned} p(D|\sigma, M) &= \int d\mu C \exp\left(-\frac{N(\mu - \bar{d})^2}{2\sigma^2}\right) \\ &= C(\sigma/\sqrt{N})\sqrt{2\pi} \end{aligned}$$

... minus a tiny bit from tails, using a proper prior.

59 / 103

Posterior

$$p(\mu|D, \sigma, M) = \frac{1}{(\sigma/\sqrt{N})\sqrt{2\pi}} \exp\left(-\frac{N(\mu - \bar{d})^2}{2\sigma^2}\right)$$

Posterior is $N(\bar{d}, w^2)$, with standard deviation $w = \sigma/\sqrt{N}$.

68.3% HPD credible region for μ is $\bar{d} \pm \sigma/\sqrt{N}$.

Note that C drops out \rightarrow limit of infinite prior range is well behaved.

60 / 103

Informative Conjugate Prior

Use a normal prior, $\mu \sim N(\mu_0, w_0^2)$

Posterior

Normal $N(\tilde{\mu}, \tilde{w}^2)$, but mean, std. deviation “shrink” towards prior.

Define $B = \frac{w^2}{w^2 + w_0^2}$, so $B < 1$ and $B = 0$ when w_0 is large.

Then

$$\begin{aligned}\tilde{\mu} &= (1 - B) \cdot \bar{d} + B \cdot \mu_0 \\ \tilde{w} &= w \cdot \sqrt{1 - B}\end{aligned}$$

“Principle of stable estimation:” The prior affects estimates only when data are not informative relative to prior.

61 / 103

Estimating a Normal Mean: Unknown σ

Problem specification

Model: $d_i = \mu + \epsilon_i$, $\epsilon_i \sim N(0, \sigma^2)$, σ is *unknown*

Parameter space: (μ, σ) ; seek $p(\mu | D, \sigma, M)$

Likelihood

$$\begin{aligned}p(D | \mu, \sigma, M) &= \frac{1}{\sigma^N (2\pi)^{N/2}} \exp\left(-\frac{Nr^2}{2\sigma^2}\right) \exp\left(-\frac{N(\mu - \bar{d})^2}{2\sigma^2}\right) \\ &\propto \frac{1}{\sigma^N} e^{-Q/2\sigma^2}\end{aligned}$$

$$\text{where } Q = N[r^2 + (\mu - \bar{d})^2]$$

62 / 103

Uninformative Priors

Assume priors for μ and σ are independent.
 Translation invariance $\Rightarrow p(\mu) \propto C$, a constant.
 Scale invariance $\Rightarrow p(\sigma) \propto 1/\sigma$ (flat in $\log \sigma$).

Joint Posterior for μ, σ

$$p(\mu, \sigma | D, M) \propto \frac{1}{\sigma^{N+1}} e^{-Q(\mu)/2\sigma^2}$$

63 / 103

Marginal Posterior

$$p(\mu | D, M) \propto \int d\sigma \frac{1}{\sigma^{N+1}} e^{-Q/2\sigma^2}$$

Let $\tau = \frac{Q}{2\sigma^2}$ so $\sigma = \sqrt{\frac{Q}{2\tau}}$ and $|d\sigma| = \tau^{-3/2} \sqrt{\frac{Q}{2}}$

$$\begin{aligned} \Rightarrow p(\mu | D, M) &\propto 2^{N/2} Q^{-N/2} \int d\tau \tau^{\frac{N}{2}-1} e^{-\tau} \\ &\propto Q^{-N/2} \end{aligned}$$

64 / 103

Write $Q = Nr^2 \left[1 + \left(\frac{\mu - \bar{d}}{r} \right)^2 \right]$ and normalize:

$$p(\mu|D, M) = \frac{(\frac{N}{2} - 1)!}{(\frac{N}{2} - \frac{3}{2})! \sqrt{\pi}} \frac{1}{r} \left[1 + \frac{1}{N} \left(\frac{\mu - \bar{d}}{r/\sqrt{N}} \right)^2 \right]^{-N/2}$$

"Student's t distribution," with $t = \frac{(\mu - \bar{d})}{r/\sqrt{N}}$

A "bell curve," but with power-law tails

Large N :

$$p(\mu|D, M) \sim e^{-N(\mu - \bar{d})^2/2r^2}$$

65 / 103

Poisson Dist'n: Infer a Rate from Counts

Problem: Observe n counts in T ; infer rate, r

Likelihood

$$\mathcal{L}(r) \equiv p(n|r, M) = p(n|r, M) = \frac{(rT)^n}{n!} e^{-rT}$$

Prior

Two simple standard choices (or conjugate gamma dist'n):

- r known to be nonzero; it is a scale parameter:

$$p(r|M) = \frac{1}{\ln(r_u/r_l)} \frac{1}{r}$$

- r may vanish; require $p(n|M) \sim \text{Const}$:

$$p(r|M) = \frac{1}{r_u}$$

66 / 103

Prior predictive

$$\begin{aligned}
 p(n|M) &= \frac{1}{r_u} \frac{1}{n!} \int_0^{r_u} dr (rT)^n e^{-rT} \\
 &= \frac{1}{r_u T} \frac{1}{n!} \int_0^{r_u T} d(rT) (rT)^n e^{-rT} \\
 &\approx \frac{1}{r_u T} \quad \text{for } r_u \gg \frac{n}{T}
 \end{aligned}$$

Posterior

A gamma distribution:

$$p(r|n, M) = \frac{T(rT)^n}{n!} e^{-rT}$$

67 / 103

Gamma Distributions

A 2-parameter family of distributions over nonnegative x , with shape parameter α and scale parameter s :

$$p_{\Gamma}(x|\alpha, s) = \frac{1}{s\Gamma(\alpha)} \left(\frac{x}{s}\right)^{\alpha-1} e^{-x/s}$$

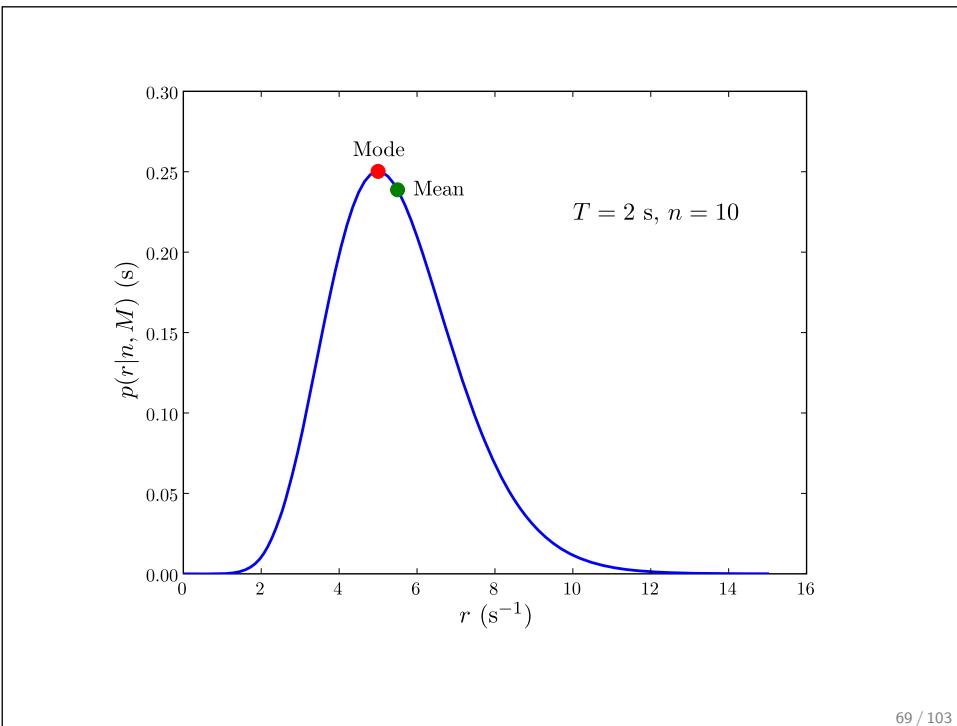
Moments:

$$\mathbb{E}(x) = s\nu \quad \text{Var}(x) = s^2\nu$$

Our posterior corresponds to $\alpha = n + 1$, $s = 1/T$.

- Mode $\hat{r} = \frac{n}{T}$; mean $\langle r \rangle = \frac{n+1}{T}$ (shift down 1 with $1/r$ prior)
- Std. dev'n $\sigma_r = \frac{\sqrt{n+1}}{T}$; credible regions found by integrating (can use incomplete gamma function)

68 / 103



The flat prior

Bayes's justification: *Not* that ignorance of $r \rightarrow p(r|I) = C$
 Require (discrete) predictive distribution to be flat:

$$\begin{aligned} p(n|I) &= \int dr p(r|I)p(n|r, I) = C \\ &\rightarrow p(r|I) = C \end{aligned}$$

Useful conventions

- Use a flat prior for a rate that may be zero
- Use a log-flat prior ($\propto 1/r$) for a nonzero scale parameter
- Use proper (normalized, bounded) priors
- Plot posterior with abscissa that makes prior flat

The On/Off Problem

Basic problem

- Look off-source; unknown background rate b
Count N_{off} photons in interval T_{off}
- Look on-source; rate is $r = s + b$ with unknown signal s
Count N_{on} photons in interval T_{on}
- Infer s

Conventional solution

$$\hat{b} = N_{\text{off}}/T_{\text{off}}; \quad \sigma_b = \sqrt{N_{\text{off}}}/T_{\text{off}}$$

$$\hat{r} = N_{\text{on}}/T_{\text{on}}; \quad \sigma_r = \sqrt{N_{\text{on}}}/T_{\text{on}}$$

$$\hat{s} = \hat{r} - \hat{b}; \quad \sigma_s = \sqrt{\sigma_r^2 + \sigma_b^2}$$

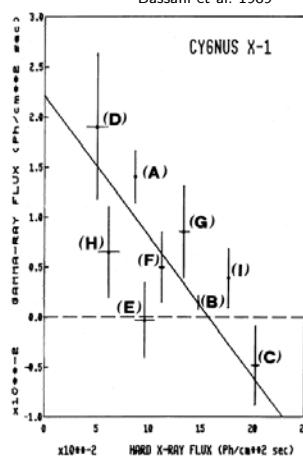
But \hat{s} can be **negative!**

71 / 103

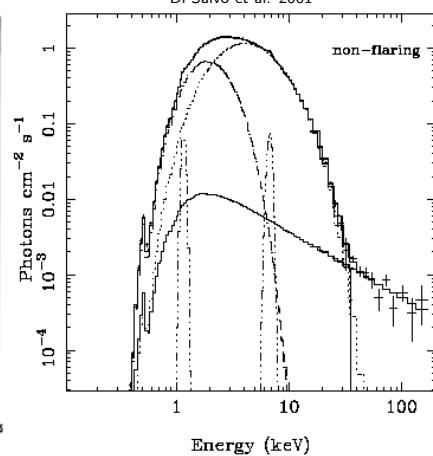
Examples

Spectra of X-Ray Sources

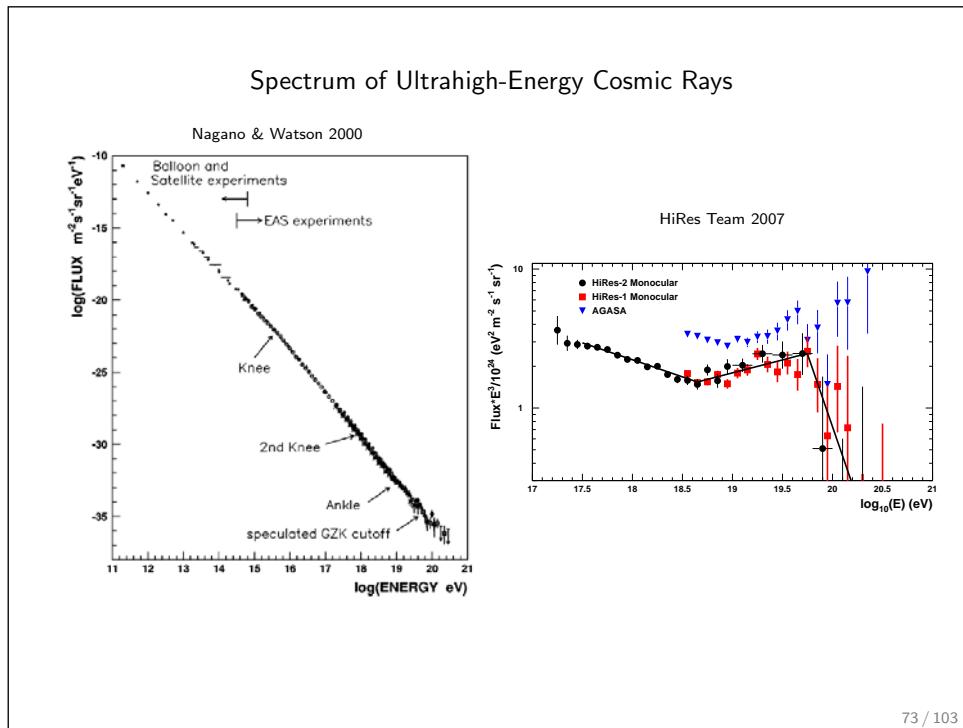
Bassani et al. 1989



Di Salvo et al. 2001



72 / 103



73 / 103

N is Never Large

"Sample sizes are never large. If N is too small to get a sufficiently-precise estimate, you need to get more data (or make more assumptions). But once N is 'large enough,' you can start subdividing the data to learn more (for example, in a public opinion poll, once you have a good estimate for the entire country, you can estimate among men and women, northerners and southerners, different age groups, etc etc). N is never enough because if it were 'enough' you'd already be on to the next problem for which you need more data."

"Similarly, you never have quite enough money. But that's another story."

— Andrew Gelman (blog entry, 31 July 2005)

74 / 103

Backgrounds as Nuisance Parameters

Background marginalization with Gaussian noise

Measure background rate $b = \hat{b} \pm \sigma_b$ with source off. Measure total rate $r = \hat{r} \pm \sigma_r$ with source on. Infer signal source strength s , where $r = s + b$. With flat priors,

$$p(s, b | D, M) \propto \exp \left[-\frac{(b - \hat{b})^2}{2\sigma_b^2} \right] \times \exp \left[-\frac{(s + b - \hat{r})^2}{2\sigma_r^2} \right]$$

75 / 103

Marginalize b to summarize the results for s (complete the square to isolate b dependence; then do a simple Gaussian integral over b):

$$p(s | D, M) \propto \exp \left[-\frac{(s - \hat{s})^2}{2\sigma_s^2} \right] \quad \hat{s} = \hat{r} - \hat{b} \quad \sigma_s^2 = \sigma_r^2 + \sigma_b^2$$

⇒ Background *subtraction* is a special case of background *marginalization*.

76 / 103

Bayesian Solution to On/Off Problem

First consider off-source data; use it to estimate b :

$$p(b|N_{\text{off}}, I_{\text{off}}) = \frac{T_{\text{off}}(bT_{\text{off}})^{N_{\text{off}}} e^{-bT_{\text{off}}}}{N_{\text{off}}!}$$

Use this as a prior for b to analyze on-source data. For on-source analysis $I_{\text{all}} = (I_{\text{on}}, N_{\text{off}}, I_{\text{off}})$:

$$p(s, b|N_{\text{on}}) \propto p(s)p(b)[(s+b)T_{\text{on}}]^{N_{\text{on}}} e^{-(s+b)T_{\text{on}}} \quad || I_{\text{all}}$$

$p(s|I_{\text{all}})$ is flat, but $p(b|I_{\text{all}}) = p(b|N_{\text{off}}, I_{\text{off}})$, so

$$p(s, b|N_{\text{on}}, I_{\text{all}}) \propto (s+b)^{N_{\text{on}}} b^{N_{\text{off}}} e^{-sT_{\text{on}}} e^{-b(T_{\text{on}}+T_{\text{off}})}$$

77 / 103

Now marginalize over b :

$$\begin{aligned} p(s|N_{\text{on}}, I_{\text{all}}) &= \int db p(s, b | N_{\text{on}}, I_{\text{all}}) \\ &\propto \int db (s+b)^{N_{\text{on}}} b^{N_{\text{off}}} e^{-sT_{\text{on}}} e^{-b(T_{\text{on}}+T_{\text{off}})} \end{aligned}$$

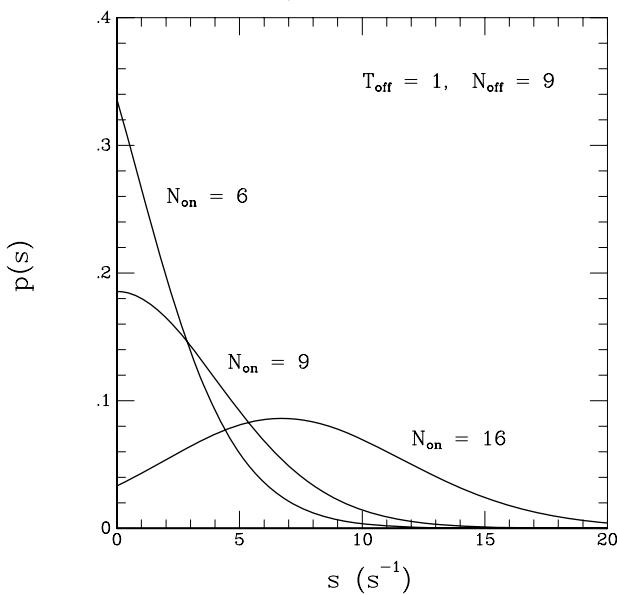
Expand $(s+b)^{N_{\text{on}}}$ and do the resulting Γ integrals:

$$\begin{aligned} p(s|N_{\text{on}}, I_{\text{all}}) &= \sum_{i=0}^{N_{\text{on}}} C_i \frac{T_{\text{on}}(sT_{\text{on}})^i e^{-sT_{\text{on}}}}{i!} \\ C_i &\propto \left(1 + \frac{T_{\text{off}}}{T_{\text{on}}}\right)^i \frac{(N_{\text{on}} + N_{\text{off}} - i)!}{(N_{\text{on}} - i)!} \end{aligned}$$

Posterior is a weighted sum of Gamma distributions, each assigning a different number of on-source counts to the source. (Evaluate via recursive algorithm or confluent hypergeometric function.)

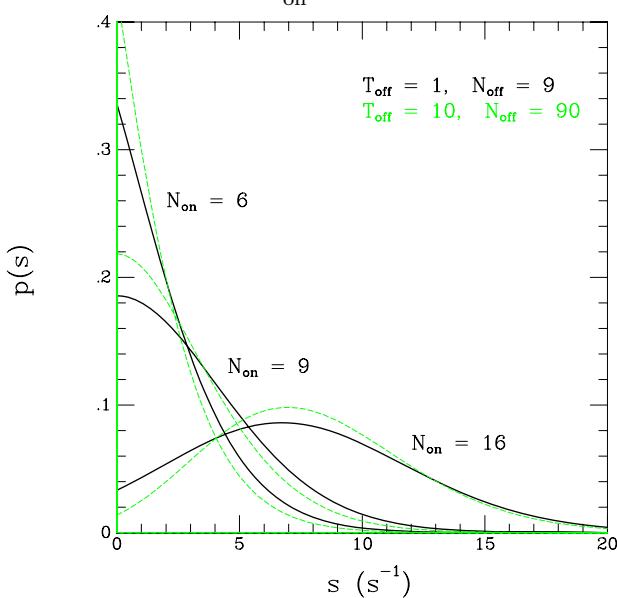
78 / 103

Example On/Off Posteriors—Short Integrations

 $T_{\text{on}} = 1$ 

79 / 103

Example On/Off Posteriors—Long Background Integrations

 $T_{\text{on}} = 1$ 

80 / 103

Multibin On/Off

The more typical on/off scenario:

Data = spectrum or image with counts in many bins

Model M gives signal rate $s_k(\theta)$ in bin k , parameters θ

To infer θ , we need the likelihood:

$$\mathcal{L}(\theta) = \prod_k p(N_{\text{on}k}, N_{\text{off}k} | s_k(\theta), M)$$

For each k , we have an on/off problem as before, only we just need the marginal likelihood for s_k (not the posterior). The same C_i coefficients arise.

XSPEC and CIAO/Sherpa provide this as an option.

CHASC approach does the same thing via data augmentation.

81 / 103

Outline

- ① The Big Picture
- ② Foundations—Logic & Probability Theory
- ③ Inference With Parametric Models
 - Parameter Estimation
 - Model Uncertainty
- ④ Simple Examples
 - Binary Outcomes
 - Normal Distribution
 - Poisson Distribution
- ⑤ Bayesian Computation
- ⑥ Measurement Error Applications
 - Number counts (flux distributions)
 - Coincidence assessment/cross-matching
- ⑦ Probability & Frequency
- ⑧ Outlook: Hotspots

82 / 103

Bayesian Computation

Large sample size: Laplace approximation

- Approximate posterior as multivariate normal $\rightarrow \det(\text{covar})$ factors
- Uses ingredients available in χ^2 /ML fitting software (MLE, Hessian)
- Often accurate to $O(1/N)$

Low-dimensional models ($d \lesssim 10$ to 20)

- Adaptive cubature
- Monte Carlo integration (importance sampling, quasirandom MC)

Hi-dimensional models ($d \gtrsim 5$)

- Posterior sampling—create RNG that samples posterior
- MCMC is most general framework



83 / 103

Outline

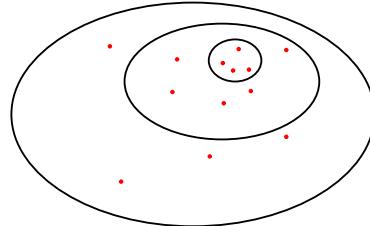
- ① The Big Picture
- ② Foundations—Logic & Probability Theory
- ③ Inference With Parametric Models
 - Parameter Estimation
 - Model Uncertainty
- ④ Simple Examples
 - Binary Outcomes
 - Normal Distribution
 - Poisson Distribution
- ⑤ Bayesian Computation
- ⑥ Measurement Error Applications
 - Number counts (flux distributions)
 - Coincidence assessment/cross-matching
- ⑦ Probability & Frequency
- ⑧ Outlook: Hotspots

84 / 103

Accounting For Measurement Error

Latent/hidden/incidental parameters

Suppose $f(x|\theta)$ is a distribution for an observable, x .

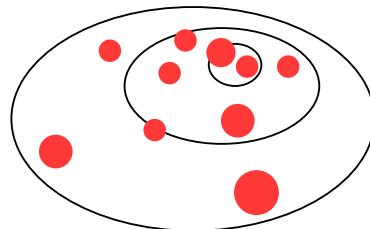


From N samples, $\{x_i\}$, we can infer θ from

$$\mathcal{L}(\theta) \equiv p(\{x_i\}|\theta) = \prod_i f(x_i|\theta)$$

85 / 103

But what if the x data are *noisy*, $d_i = x_i + \epsilon_i$?



We should somehow incorporate $\ell_i(x_i) = p(d_i|x_i)$

$$\begin{aligned} \mathcal{L}(\theta, \{x_i\}) &\equiv p(\{d_i\}|\theta, \{x_i\}) \\ &= \prod_i \ell_i(x_i) f(x_i|\theta) \end{aligned}$$

This is an example of *Bayesian multilevel (hierarchical) modeling*. Related to Eddington/Malmquist/Lutz-Kelker biases.

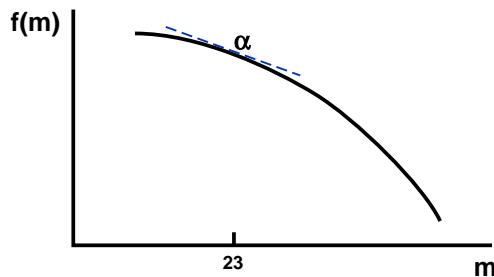
Key point: *Maximizing over x_i and integrating over x_i give very different results!*

86 / 103

Example—Distribution of Source Fluxes

Measure $m = -2.5 \log(\text{flux})$ from sources following a “rolling power law” distribution (inspired by trans-Neptunian objects)

$$f(m) \propto 10^{[\alpha(m-23) + \alpha'(m-23)^2]}$$

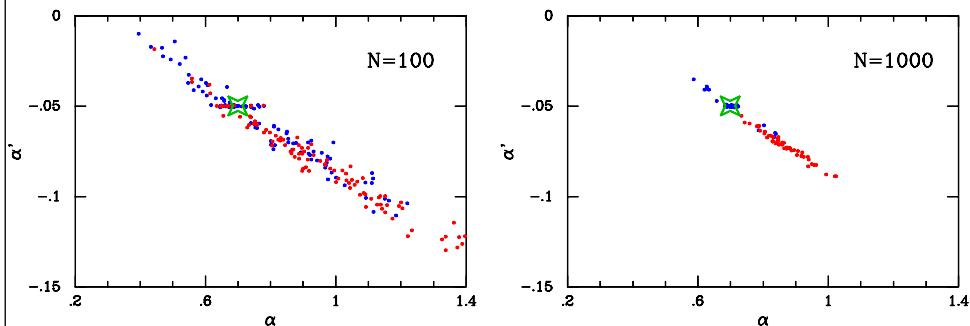


Measurements have uncertainties 1% (bright) to $\approx 30\%$ (dim)

Analyze simulated data with maximum likelihood and Bayes

87 / 103

Parameter estimates from Bayes (dots) and maximum likelihood (circles):



Uncertainties don't average out!

Crucial for “errors in variables,” [survey analysis](#)

Recent application areas: SN 1987A vs; GRBs; TNOs

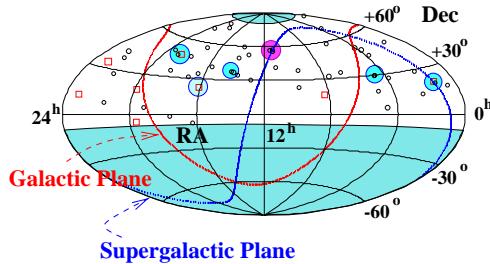
88 / 103

Bayesian Coincidence Assessment

Ultra-High Energy Cosmic Rays

AGASA data above GZK cutoff (Hayashida et al. 2000)

AGASA + A20

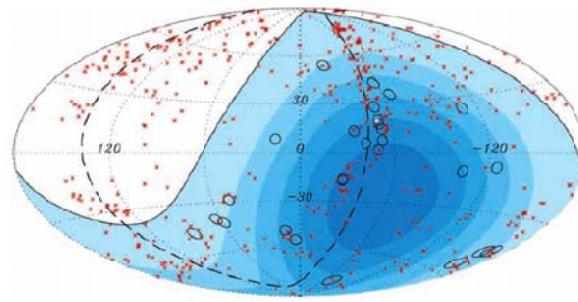


- 58 events with $E > 4 \times 10^{19}$ eV
- Energy-dependent direction uncertainty $\sim 2^\circ$
- Significance test — Search for coincidences $< 2.5^\circ$:
 - 6 pairs; $\lesssim 1\%$ significance
 - 1 triplet; $\lesssim 1\%$ significance

89 / 103

New Data: UHECRs–AGN Association?

Auger data above GZK cutoff (Nov 2007)



- 27 events with $E > 5.7 \times 10^{19}$ eV
- Energy-dependent direction uncertainty $\lesssim 1^\circ$
- Crosses = 472 AGN with distance $D < 75$ Mpc
- Significance test of correlation with AGN:
 - Tune E , D , angle cutoffs with early events
 - Apply to 13 new events $\rightarrow p\text{-value } 1.7 \times 10^{-3}$

90 / 103

Scientific Significance

- High-energy (not UHE) CRs have isotropic directions → cosmological sources
- HE and UHE CRs are protons or nuclei
- UHE CR “sees” CMB photons as having very high energy → GZK cutoff; CRs with $E > 50$ EeV are destroyed in ~ 100 Mpc
- If UHECRs are isotropic, there is *new physics*
- If no new physics, UHECR directions may point to local sources (cosmic B field distorts this)

91 / 103

Direction Uncertainties: Fisher Distribution

“Gaussian on the sky”

Let $\hat{\mathbf{n}}$ be the best-fit direction. For *azimuthally symmetric* uncertainties, use:

$$\mathcal{L}(\mathbf{n}) = \frac{\kappa}{4\pi \sinh \kappa} e^{\kappa \mathbf{n} \cdot \hat{\mathbf{n}}}$$

κ = concentration parameter. For small uncertainties,

$$\kappa \approx \frac{C}{\sigma^2}, \quad C \approx 2.3$$

If \mathbf{n} is near $\hat{\mathbf{n}}$ (angle θ)

$$\mathcal{L}(\mathbf{n}) \sim \exp \left[-\frac{C\theta^2}{2\sigma^2} \right]$$

(For asymmetric uncertainties, could use Kent dist'n.)

92 / 103

Bayesian Coincidence Assessment

Direction uncertainties accounted for via likelihoods for object directions:

$$\mathcal{L}_i(\mathbf{n}) = p(d_i|\mathbf{n}), \quad \text{normalized w.r.t. } \mathbf{n} \text{ (convention)}$$

H_0 : No repetition

$$\begin{aligned} p(d_1, d_2|H_0) &= \int d\mathbf{n}_1 p(\mathbf{n}_1|H_0) \mathcal{L}_1(\mathbf{n}_1) \times \int d\mathbf{n}_2 \dots \\ &= \frac{1}{4\pi} \int d\mathbf{n}_1 \mathcal{L}_1(\mathbf{n}_1) \times \frac{1}{4\pi} \int d\mathbf{n}_2 \dots \\ &= \frac{1}{(4\pi)^2} \end{aligned}$$

93 / 103

H_1 : Repeating (same direction!)

$$p(d_1, d_2|H_0) = \int d\mathbf{n} p(\mathbf{n}|H_0) \mathcal{L}_1(\mathbf{n}) \mathcal{L}_2(\mathbf{n})$$

Odds favoring repetition:

$$\begin{aligned} O &= 4\pi \int d\mathbf{n} \mathcal{L}_1(\mathbf{n}) \mathcal{L}_2(\mathbf{n}) \\ &\approx \frac{2C}{\sigma_{12}^2} \exp \left[-\frac{C\theta_{12}^2}{2\sigma_{12}^2} \right]; \quad \sigma_{12}^2 = \sigma_1^2 + \sigma_2^2 \end{aligned}$$

$$\begin{aligned} \text{E.g.: } \sigma_1 = \sigma_2 = 10^\circ \quad O &\approx 1.5 \text{ for } \theta_{12} = 26^\circ \\ &O \approx 75 \text{ for } \theta_{12} = 0^\circ \end{aligned}$$

$$\begin{aligned} \sigma_1 = \sigma_2 = 25^\circ \quad O &\approx 7 \text{ for } \theta_{12} = 26^\circ \\ &O \approx 12 \text{ for } \theta_{12} = 0^\circ \end{aligned}$$

94 / 103

Challenge: Large hypothesis spaces

For $N = 2$ events, there was a single coincidence hypothesis, M_1 above.

For $N = 3$ events:

- Three doublets: $1 + 2$, $1 + 3$, or $2 + 3$
- One triplet

The number of alternatives grows combinatorially; we must assign sensible priors to them, and sum over them (or at least all important ones).

95 / 103

Small- N Brute Force Example

Bayesian Coincidence Assessment for AGASA UHECRs

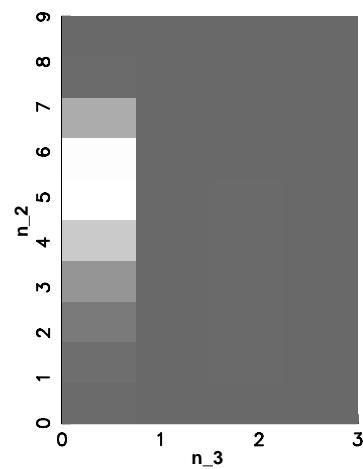
$N = 58$ directions; search for coincidences

n_2	n_3	\mathcal{N}
1	0	1653
2	0	1,272,810
3	0	607,130,370
0	1	30,856
0	2	404,753,580

Method:

- Identify all pairs (13) and triplets (3) with multiplet Bayes factors > 1
- Generate & sum over all partitions including those multiplets (gives lower bound)
- Use flat prior over all possible (n_2, n_3)

Odds for repetition: 1.4 (i.e., no significant evidence)



96 / 103

Outline

- ① The Big Picture
- ② Foundations—Logic & Probability Theory
- ③ Inference With Parametric Models
 - Parameter Estimation
 - Model Uncertainty
- ④ Simple Examples
 - Binary Outcomes
 - Normal Distribution
 - Poisson Distribution
- ⑤ Bayesian Computation
- ⑥ Measurement Error Applications
 - Number counts (flux distributions)
 - Coincidence assessment/cross-matching
- ⑦ Probability & Frequency
- ⑧ Outlook: Hotspots

97 / 103

Probability & Frequency

Frequencies are relevant when modeling repeated trials, or repeated sampling from a population or ensemble.

Frequencies are *observables*:

- When available, can be used to *infer* probabilities for next trial
- When unavailable, can be

Bayesian/Frequentist relationships:

- General relationships between probability and frequency
- Long-run performance of Bayesian procedures
- Examples of Bayesian/frequentist differences

98 / 103

Relationships Between Probability & Frequency

Frequency from probability

Bernoulli's law of large numbers: In repeated i.i.d. trials, given $P(\text{success} | \dots) = \alpha$, predict

$$\frac{N_{\text{success}}}{N_{\text{total}}} \rightarrow \alpha \quad \text{as} \quad N_{\text{total}} \rightarrow \infty$$

Probability from frequency

Bayes's "An Essay Towards Solving a Problem in the Doctrine of Chances" → First use of Bayes's theorem:
Probability for success in next trial of i.i.d. sequence:

$$E\alpha \rightarrow \frac{N_{\text{success}}}{N_{\text{total}}} \quad \text{as} \quad N_{\text{total}} \rightarrow \infty$$

99 / 103

Subtle Relationships For Non-IID Cases

Predict frequency in dependent trials

r_t = result of trial t ; $p(r_1, r_2 \dots r_N | M)$ known; predict f :

$$\langle f \rangle = \frac{1}{N} \sum_t p(r_t = \text{success} | M)$$

where $p(r_1 | M) = \sum_{r_2} \dots \sum_{r_N} p(r_1, r_2 \dots | M)$

Expected frequency of outcome in many trials = average probability for outcome across trials.
But also find that σ_f needn't converge to 0.

Infer probabilities for different but related trials

Shrinkage: Biased estimators of the probability that share info across trials are better than unbiased/BLUE/MLE estimators.

A formalism that distinguishes p from f from the outset is particularly valuable for exploring subtle connections. E.g., shrinkage is explored via hierarchical and empirical Bayes.

100 / 103

Frequentist Performance of Bayesian Procedures

Many results known for parametric Bayes performance:

- Estimates are consistent if the prior doesn't exclude the true value.
- Credible regions found with flat priors are typically confidence regions to $O(n^{-1/2})$; "reference" priors can improve their performance to $O(n^{-1})$.
- Marginal distributions have better frequentist performance than conventional methods like profile likelihood. (Bartlett correction, ancillaries, bootstrap are competitive but hard.)
- Bayesian model comparison is asymptotically consistent (not true of significance/NP tests, AIC).
- For separate (not nested) models, the posterior probability for the true model converges to 1 exponentially quickly.
- Wald's complete class theorem: *Optimal* frequentist methods are *Bayes rules* (equivalent to Bayes for some prior)
- ...

Parametric Bayesian methods are typically good frequentist methods.

(Not so clear in nonparametric problems.)

101 / 103

Outline

- ① The Big Picture
- ② Foundations—Logic & Probability Theory
- ③ Inference With Parametric Models
 - Parameter Estimation
 - Model Uncertainty
- ④ Simple Examples
 - Binary Outcomes
 - Normal Distribution
 - Poisson Distribution
- ⑤ Bayesian Computation
- ⑥ Measurement Error Applications
 - Number counts (flux distributions)
 - Coincidence assessment/cross-matching
- ⑦ Probability & Frequency
- ⑧ Outlook: Hotspots

102 / 103

Some Bayesian Astrostatistics Hotspots

- Cosmology
 - Parametric modeling of CMB, LSS, SNe Ia → cosmo params
 - Nonparametric modeling of SN Ia multicolor light curves
 - Nonparametric “emulation” of cosmological models
- Extrasolar planets
 - Parametric modeling of Keplerian reflex motion (planet detection, orbit estimation)
 - Optimal scheduling via Bayesian experimental design
- Photon counting data (X-rays, γ -rays, cosmic rays)
 - Upper limits, hardness ratios
 - Parametric spectroscopy (line detection, etc.)
- Gravitational wave astronomy
 - Parametric modeling of binary inspirals
 - Hi-multiplicity parametric modeling of white dwarf background

Summer School in Statistics for
Astronomers IV
June 12, 2008

Multivariate Analysis

Donald Richards

Department of Statistics
Center for Astrostatistics
Penn State University

Notes revised by

T.Krishnan

Cranes Software International Limited
Bangalore

Lectures delivered by

James L Rosenberger

Department of Statistics
Penn State University

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, mu-max, Mcz, e.Mcz, MCzml, ..., IFD, e.IFD

63,501 objects: galaxies

<http://astrostatistics.psu.edu/datasets/COMBO17.dat>

Rmag 2	mumax 5	Mcz 6	MCzml 8	chi2red 9	UjMAG 10	BjMAG 12	VjMAG 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^{th} observation on the j^{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}	
	:	:	:		:	
	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^{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^{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:

$$\mathbf{R} = \begin{bmatrix} 1 & r_{12} & \cdots & r_{1p} \\ r_{21} & 1 & \cdots & r_{2p} \\ \vdots & \vdots & & \vdots \\ r_{p1} & r_{p2} & \cdots & 1 \end{bmatrix}$$

S and **R** are *symmetric*

S and **R** are *positive semidefinite*: $\mathbf{v}^T S \mathbf{v} \geq 0$ 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 **S** is singular so is **R** and conversely.

If $n \leq p$ then **S** and **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,

$$s_{11} > 0, \begin{vmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{vmatrix} > 0, \begin{vmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{vmatrix} > 0,$$

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, μ max, Mcz, MCzml, chi2red, UjMAG, BjMAG, VjMAG

$p = 8$ and $n = 3462$

The sample mean vector:

Rmag	μ max	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	μ max	Mcz	MCzml	chi2red	UjMAG	BjMAG	VjMAG
Rmag	2.062	1.362	0.190	0.234	0.147	0.890	1.015	1.060
μ max	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

$$4 * \text{Rmag} + 3 * \text{mumax} + |\text{Mcz}-\text{MCzml}| - \text{chi2red} + (\text{UjMAG} + \text{BjMAG})^2 + \text{VjMAG}^2 < 70?$$

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} :

$$\Sigma = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1p} \\ \sigma_{21} & \cdots & \sigma_{2p} \\ \vdots & & \vdots \\ \sigma_{p1} & \cdots & \sigma_{pp} \end{bmatrix}$$

An easy result:

$$\Sigma = E(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^T$$

Also,

$$\Sigma = E(\mathbf{X}\mathbf{X}^T) - \boldsymbol{\mu}\boldsymbol{\mu}^T$$

To avoid pathological cases, we assume that Σ is nonsingular

Theory vs. Practice

Population 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 $\mu = E(\mathbf{X})$	Sample mean $\bar{\mathbf{x}} = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k$
Popn. cov. matrix $\Sigma =$ $E(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T$	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,
 $\bar{\mathbf{x}} \rightarrow \mu$ and $S \rightarrow \Sigma$ as $n \rightarrow \infty$

The Multivariate Normal Distribution

$\mathbf{X} = [X_1, \dots, X_p]^T$: 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, Σ diagonal:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & & 0 \\ 0 & 0 & \cdots & \sigma_p^2 \end{bmatrix}$$

$$|\Sigma| = \sigma_1^2 \sigma_2^2 \cdots \sigma_p^2$$

$$\Sigma^{-1} = \begin{bmatrix} \sigma_1^{-2} & 0 & \cdots & 0 \\ 0 & \sigma_2^{-2} & \cdots & 0 \\ \vdots & \vdots & & 0 \\ 0 & 0 & \cdots & \sigma_p^{-2} \end{bmatrix}$$

$$f(\mathbf{x}) = \prod_{j=1}^p \frac{1}{(2\pi\sigma_j^2)^{1/2}} \exp \left[-\frac{1}{2} \left(\frac{x_j - \mu_j}{\sigma_j} \right)^2 \right]$$

Conclusion: X_1, \dots, X_p are mutually independent and normally distributed

Recall: $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ if its p.d.f. is of the form

$$f(\mathbf{x}) = \text{const.} \times \exp \left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]$$

where

$$\text{const.} = \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}|^{1/2}}$$

Facts:

$$\boldsymbol{\mu} = E(\mathbf{X}),$$

$$\boldsymbol{\Sigma} = \text{Cov}(\mathbf{X})$$

$$\int_{\mathbb{R}^p} f(\mathbf{x}) d\mathbf{x} = 1$$

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 Σ 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

$$\boldsymbol{\mu}_u = E(\mathbf{X}_u), \quad \boldsymbol{\mu}_l = E(\mathbf{X}_l)$$

$$\boldsymbol{\Sigma}_{uu} = \text{Cov}(\mathbf{X}_u), \quad \boldsymbol{\Sigma}_{ll} = \text{Cov}(\mathbf{X}_l)$$

$$\boldsymbol{\Sigma}_{ul} = \text{Cov}(\mathbf{X}_u, \mathbf{X}_l)$$

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

A confidence region for μ

Naive method: Using only the data on the i th variable, construct a confidence interval for each μ_i

Use the collection of confidence intervals as a confidence region for μ

Good news: This can be done using elementary statistical methods

Bad news: A collection of 95% confidence intervals, one for each μ_i , does not result in a 95% confidence region for μ

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 μ 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 μ

Fact: Every positive definite symmetric matrix has a unique positive definite symmetric square root

$\Sigma^{-1/2}$: The p.d. square-root of Σ^{-1}

Recall (see p. 31): If A is a $p \times p$ nonsingular matrix and $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \Sigma)$ then

$$A\mathbf{X} + \mathbf{b} \sim N_p(A\boldsymbol{\mu} + \mathbf{b}, A\Sigma A^T)$$

Set $A = \Sigma^{-1/2}$, $\mathbf{b} = -\Sigma^{-1/2}\boldsymbol{\mu}$

Then $A\boldsymbol{\mu} + \mathbf{b} = \mathbf{0}$, $A\Sigma A^T = \Sigma^{-1/2}\Sigma\Sigma^{-1/2} = I_p$

$$\Sigma^{-1/2}(\mathbf{X} - \boldsymbol{\mu}) \sim N_p(\mathbf{0}, I_p)$$

$I_p = \text{diag}(1, 1, \dots, 1)$, 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

Principal Components Analysis (PCA)

COMBO-17: $p = 65$ (wow!)

Can we reduce the dimension of the problem?

\mathbf{X} : A p -dimensional random vector

Covariance matrix: Σ

Solve for λ : $|\Sigma - \lambda I| = 0$

Solutions: $\lambda_1, \dots, \lambda_p$, the *eigenvalues* of Σ

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 Σ

Scale each eigenvector to make its length 1

$\mathbf{v}_1, \dots, \mathbf{v}_p$ are orthogonal

The first PC: 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 Σ

$\mathbf{v}_1^T \mathbf{X}$ 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 Σ

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 Σ

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 : Rmag

X_2 : mumax

etc.

The PCs usually have no physical meaning, but they can provide insight into the data analysis

$\tilde{\lambda}_1 + \cdots + \tilde{\lambda}_p$: A measure of total variability of the data

$\frac{\tilde{\lambda}_k}{\tilde{\lambda}_1 + \cdots + \tilde{\lambda}_p}$: The proportion of total variability of the data “explained” by the k th PC

How many PC's should we calculate?

Stop when

$$\frac{\tilde{\lambda}_1 + \cdots + \tilde{\lambda}_k}{\tilde{\lambda}_1 + \cdots + \tilde{\lambda}_p} \geq 0.9$$

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

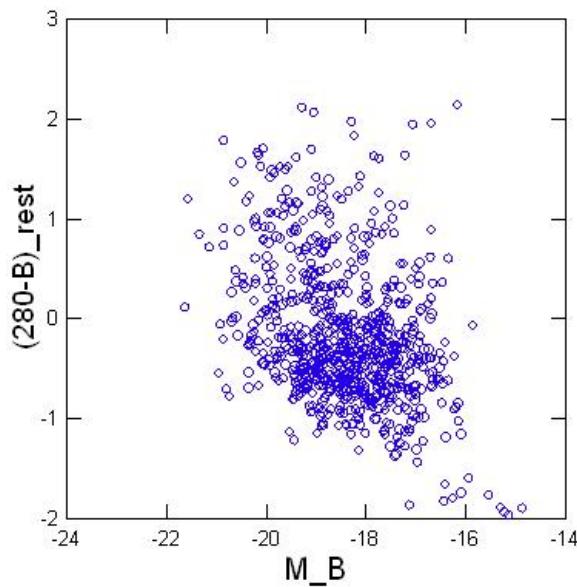
Argument against: If some components have significantly smaller means and variances than others then correlation-based PCs will give all components similarly-sized weights

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 Bjmag (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/astroph/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:

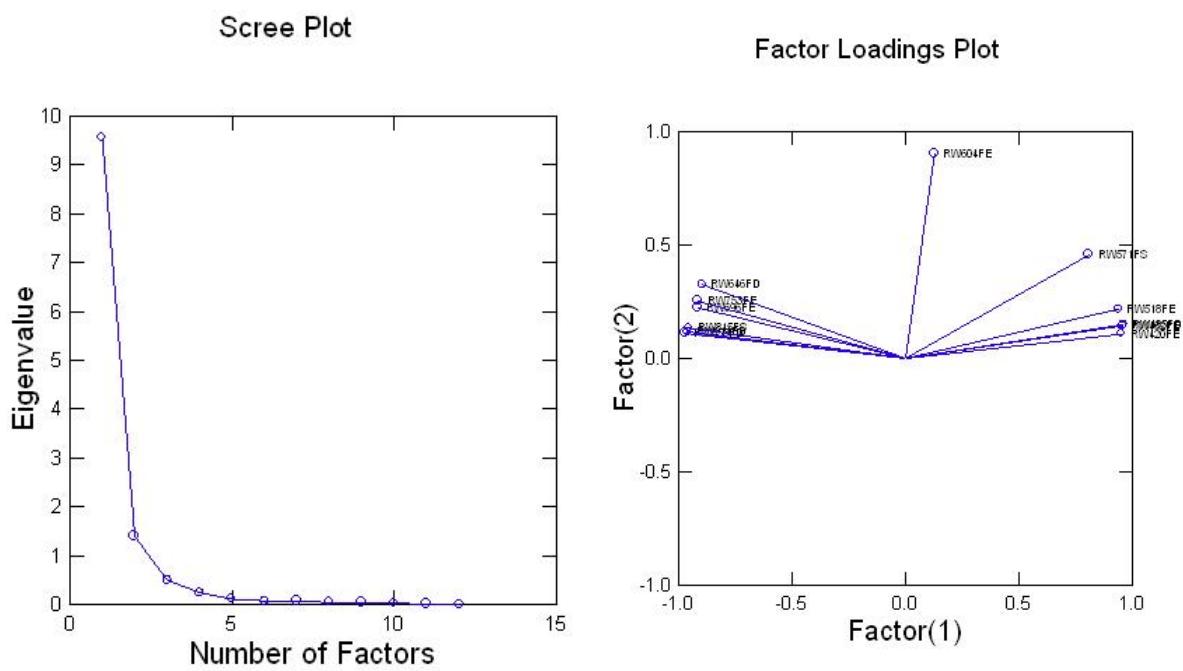
RW420FE	RW462FE	RW485FD	RW518FE	RW571FS	RW604FE	BJMAG	MCZ	U-V	COLOR
-0.018	-0.006	0.000	-0.001	-0.004	-0.002	-17.540	0.832	0.090	1
-0.003	0.002	-0.000	-0.002	0.007	0.006	17.860	0.927	-0.080	1
-0.010	-0.003	0.002	-0.007	-0.000	0.000	-19.910	1.202	0.660	2
0.006	0.010	-0.005	-0.004	-0.005	0.003	-17.390	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

Mean properties of the three galaxy SED class samples.

Property	Dust-free old	Dusty red-seq	Blue cloud
N_{galaxy}	294	168	333
$N_{\text{fieldcontamination}}$	6	7	49
N_{spectra}	144	69	36
z_{spec}	0.1646	0.1646	0.1658
$\sigma_{cz}/(1+z)$ /(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
$\sigma_{cz,N}/(1+z)$ /(km/s)	589	597	N/A
$\sigma_{cz,S}/(1+z)$ /(km/s)	522	546	N/A
$\log(\Sigma_{10}(\text{Mpc}/\text{h})^2)$	2.188	1.991	1.999
$EW_e(OII)/\text{\AA}$	N/A	4.2 ± 0.4	17.5 ± 1.5
$EW_a(H\delta)/\text{\AA}$	2.3 ± 0.5	2.6 ± 0.5	4.5 ± 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

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

$$\begin{aligned} & 0.345 + \text{RW420FE}*14.277 - \text{RW462FE}*0.844 \\ & - \text{RW485FD}*36.890 + \text{RW518FE}*6.541 \\ & + \text{RW571FS}*2.249 + \text{RW604FE}*25.670 \\ & + \text{RW646FD}*18.331 + \text{RW696FE}*15.123 - \text{RW753FE}*29.072 \\ & - \text{RW815FS}*16.970 - \text{RW856FD}*16.467 + \text{RW914FD}*2.024 \end{aligned}$$

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	Classified		
Group	Group		
	1	2	%correct
-----+-----			
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

Develop a formula for prediction of $U - V$ in a new galaxy from "R" values.

Results of such a multiple (linear) regression analysis:

Multiple correlation: a measure of how good the regression is: 0.344

Not very good—much as in Discriminant Analysis

Table below shows which "R" variables are useful for prediction of $U - V$: those with small p -values.

Regression Coefficients and their significance

Effect	Coefficient	Standard Error	t	p-value
CONSTANT	0.175	0.012	15.010	0.000
RW420FE	-1.624	0.839	-1.936	0.053
RW462FE	0.895	1.371	0.653	0.514
RW485FD	5.072	1.664	3.049	0.002
RW518FE	-1.921	1.199	-1.602	0.109
RW571FS	-1.126	1.178	-0.956	0.339
RW604FE	-4.636	1.456	-3.184	0.001
RW646FD	-2.345	1.340	-1.750	0.080
RW696FE	-2.729	0.917	-2.977	0.003
RW753FE	3.943	1.020	3.866	0.000
RW815FS	3.394	0.902	3.761	0.000
RW856FD	3.059	0.961	3.182	0.001
RW914FD	0.036	0.740	0.049	0.961

Multivariate Computations

This tutorial deals with a few multivariate techniques including clustering and principal components. We begin with a short introduction to generating multivariate normal random vectors.

Multivariate normal distributions

We'll start off by generating some multivariate normal random vectors. There are packages that do this automatically, such as the `mvtnorm` package available from CRAN, but it is easy and instructive to do from first principles.

Let's generate from a bivariate normal distribution in which the standard deviations of the components are 2 and 3 where the correlation between the components is -1/2. For simplicity, let the mean of the vectors be the origin. We need to figure out what the covariance matrix looks like.

The diagonal elements of the covariance matrix are the marginal variances, namely 4 and 9. The off-diagonal element is the covariance, which equals the correlation times the product of the marginal standard deviations, or -3:

```
sigma <- matrix(c(4,-3,-3,9),2,2)
sigma
```

We now seek to find a matrix M such that M times its transpose equals σ . There are many matrices that do this; one of them is the transpose of the Cholesky square root:

```
M <- t(chol(sigma))
M %*% t(M)
```

We now recall that if Z is a random vector and M is a matrix, then the covariance matrix of MZ equals $M \text{cov}(Z) M^t$. It is very easy to simulate normal random vectors whose covariance matrix is the identity matrix; this is accomplished whenever the vector components are independent standard normals. Thus, we obtain a multivariate normal random vector with covariance matrix σ if we first generate a standard normal vector and then multiply by the matrix M above. Let us create a dataset with 200 such vectors:

```
Z <- matrix(rnorm(400),2,200) # 2 rows, 200 columns
X <- t(M %*% Z)
```

The transpose above is taken so that X becomes a 200×2 matrix, since R prefers to have the columns as the vector components rather than the rows. Let us now plot the randomly generated normals and find the sample mean and covariance.

```
plot(X)
Xbar <- apply(X,2,mean)
S <- cov(X)
```

We can compare the S matrix with the σ matrix, but it is also nice to plot an ellipse to see what shape these matrices correspond to. The `car` package, which we used in the [EDA and regression](#) tutorial, has the capability to plot ellipses. You might not need to run the `install.packages` function below since this package may already have been installed during the earlier tutorials. However, the `library` function is necessary.

```
install.packages("car",lib="V:/")
library(car,lib.loc="V:/")
```

To use the `ellipse` function in the `car` package, we need the center (mean), shape (covariance), and the radius. The radius is the radius of a circle that represents the "ellipse" for a standard bivariate normal distribution. To understand how to provide a radius, it is helpful to know that if we sum the squares of k independent standard normal random variables, the result is (by definition) a chi-squared random variable

on k degrees of freedom. Thus, for a standard bivariate normal vector, the squares of the radii should be determined by the quantiles of the chi-squared distribution on 2 degrees of freedom. Let us then construct an ellipse with radius based on the median of the chi-squared distribution. Thus, this ellipse should contain roughly half of the points generated. We'll also produce a second ellipse, based on the true mean and covariance matrix, for purposes of comparison.

```
ellipse(xbar, s, sqrt(qchisq(.5,2)))
ellipse(c(0,0), sigma,
       sqrt(qchisq(.5,2)), col=3, lty=2)
```

Singular value decomposition

The [COMBO-17](#) dataset provides brightness measurements on 3462 galaxies. Here, we use a subset of this dataset to try to reproduce a figure that appears on the [COMBO17.html](#) web page (not completely successfully, it turns out, though I'm not quite sure why).

```
combo <- read.csv("http://astrostatistics.psu.edu/datasets/COMBO17.csv",na.strings="NA")
names(combo)
attach(combo)
xy <- cbind(BjMAG, S280MAG-BjMAG)
xy <- xy[McZ>.7 & McZ<.9,]
xy <- na.omit(xy)
plot(xy[,1],xy[,2], pch=20,
     xlab=expression(M[B]),
     ylab=expression("(280-B)"[rest]),
     main="z between 0.7 and 0.9",
     cex.lab=1.5)
```

Notice that the `plot` command uses the `expression` function. This is one way to put mathematical notation to an R plot. See the help function for `plotmath` for details.

To duplicate the covariance and correlation matrices seen in the lecture, let us consider a subset of the 65 combo variables:

```
subcombo <- combo[,c(2,5,6,8,9,10,12,14)]
names(subcombo)
var(subcombo) # Covariance matrix
cor(subcombo) # Correlation matrix
```

Now let's do a quick principal components analysis using the singular value decomposition. First, we obtain the SVD:

```
s <- svd(subcombo)
names(s)
U <- s$u
V <- s$v
D <- diag(s$d)
```

There are three matrices that make up the SVD: U, D, and V. The U and V matrices are 3462x8 and 8x8 matrices, respectively, and they are both orthogonal. We can check the orthogonality of U and V by simply multiplying:

```
t(U) %*% U
t(V) %*% V
```

The SVD has the property that X (in this case, subcombo) is equal to U %*% D %*% t(V):

```
max(abs(subcombo - U %*% D %*% t(V))) # Should be zero
```

Now let's obtain the principal components scores. To do this, we must first center each column of the subcombo matrix. To do this, we'll use the sweep function:

```
cent.subcombo <- sweep(subcombo, 2, apply(subcombo, 2, mean))
```

To check that this has worked, let's verify that the mean of each column of cent.subcombo equals zero:

```
apply(cent.subcombo, 2, mean)
```

We don't need the old 's' object any more, so let's rename it now so that it is the SVD for the centered dataset:

```
s <- svd(cent.subcombo)
U <- s$u
V <- s$v
D <- diag(s$d)
```

To obtain the principal component scores for each galaxy, we merely multiply U by D.

```
pcscores <- U %*% D
plot(pcscores[,1:2])
```

Each of the first two principal components (plotted above) is a linear combination of the original eight variables. To find out what these linear combinations are, we may examine the V matrix. Remember, if we multiply the principal component scores by t(V), we obtain the original centered dataset.

Model-based clustering

Let's apply some of the bivariate normal results seen earlier to looking for clusters in the COMBO-17 dataset. In model-based clustering, the assumption is (usually) that the multivariate sample is a random sample from a mixture of multivariate normal distributions. A **mixture** in this case is a weighted sum of different normal distributions (recall our discussion regarding mixtures in the [Likelihood Computations and Random Numbers](#) tutorial).

As a refresher for mixtures, suppose there exist k multivariate normal distributions (subpopulations). To select our sample, someone in a closed room first rolls a k-sided die (not necessarily a fair die), then selects a random member of the subpopulation indicated by the die. The *only* thing we get to observe is the final observation; we do not know which number came up on the die or any of the characteristics (parameters) of the normal subpopulations.

To take a simple example, suppose you are given a dataset consisting *only* of the heights of a sample of individuals. You know that there are two subpopulations, males and females, and that heights in each subpopulation are roughly normally distributed. Is it possible, without knowing the sexes corresponding to the measurements you are given, to estimate the means and standard deviations for each sex, along with the proportion of males? The answer is yes.

Note here that model-based clustering using mixture models is not the same thing as discriminant analysis, in which we are given not only observations but also their known class memberships. The goal in discriminant analysis is to build a rule for classifying future observations based on a training sample, whereas clustering usually has broader goals than this: To discover the classes in the first place.

There is a CRAN package that does model-based clustering assuming normal distributions. It is called mclust. There is an updated version of mclust on CRAN, but we'll use an older version called mclust02.

```
install.packages("mclust02", lib="V:/") # The lib= part might not be needed
library(mclust02, lib.loc="V:/")
```

Let's first fit a two-component normal mixture model (i.e., search for two multivariate normal clusters).

```
mc2 <- Mclust(xy, minG=2, maxG=2)
```

```
names(mc2)
mc2$mu
mc2$sigma
```

Let's take a look at 50% ellipses of the clustering solution to see what the solution "looks like".

```
plot(xy[,1],xy[,2], pch=20,
      xlab=expression(M[B]),
      ylab=expression("(280-B)[rest]"),
      main="z between 0.7 and 0.9",
      cex.lab=1.5)
r <- sqrt(qchisq(.5,2))
for(i in 1:mc2$G)
  ellipse(mc2$mu[,i], mc2$sigma[,,i], r, col=1+i)
```

Is two components the "best" solution? There is no best answer to this question in general, but we can do things like consider model selection criteria to try to decide. By default, the Mclust uses BIC to search for a best model from 1 to 9 components:

```
mc <- Mclust(xy)
mc$G
```

So let's see what the three-component solution looks like:

```
plot(xy[,1],xy[,2], pch=20,
      xlab=expression(M[B]),
      ylab=expression("(280-B)[rest]"),
      main="z between 0.7 and 0.9",
      cex.lab=1.5)
for(i in 1:mc$G)
  ellipse(mc$mu[,i], mc$sigma[,,i], r, col=1+i)
```

Principal components

We now look at a different dataset, the SDSS quasar dataset described at http://www.astrostatistics.psu.edu/datasets/SDSS_quasar.html.

```
quas <- read.table("http://astrostatistics.psu.edu/datasets/SDSS_quasar.dat", header=T)
dim(quas)
```

We want to get rid of the missing values. However, in this case missing values create more havoc than usual due to the fact that we will be working with covariance matrices. Thus, we will eliminate all rows with missing values:

```
quas[quas==0 | quas==1 | quas==9] <- NA
quas <- na.omit(quas)
dim(quas)
```

This leaves us with a much smaller dataset, but for purposes of illustration it will serve well. Once the principal component loadings are determined, we can then apply these loadings, or a simplified version thereof, to the *whole* dataset.

In principal components analysis, we wish to reduce the number of variables. The method is to find the "best" linear combinations of all existing variables. To understand what is "best" in this context, consider the 22 quantitative measurement columns in the quas dataset (the first column is the SDSS designation of the object). Each row may be considered to be a point in 22-dimensional Euclidean space. Thus, the entire dataset consists of a cloud of 279 22-dimensional points. The "best" linear combination here will be the single vector in 22-space parallel to which the variance of these 279 points is the greatest. The second-best will be the single vector orthogonal to the first along which the variance is the greatest, and so on.

We will implement principal components in R using two distinct approaches. One approach is to use the [princomp](#) function. Another is to obtain the same results from scratch using an eigenvalue decomposition. We will use the former approach for analysis and interpretation; the latter approach is presented only to help you understand how the method works mathematically.

To create a single object containing all the principal components information you will need, type

```
pc <- princomp(quas[, -1])
```

Note that we omit the first column from the analysis since it is not a quantitative measurement.

Let's see what kind of information is carried in pc.

```
names(pc)
?princomp
```

Before explaining what each of these things means, let's briefly show how to obtain the important bits, namely pc\$sdev and pc\$loadings, from scratch using an eigenvalue/eigenvector decomposition of the sample covariance matrix. The square roots of the eigenvalues give pc\$sdev and the matrix of normalized eigenvectors gives pc\$loadings. (Note, however, that a normalized eigenvector is still a normalized eigenvector if multiplied by -1; therefore, some of the columns of the eigenvector matrix differ from the corresponding columns of pc\$loadings by a sign change.)

In other words, it is possible to reconstruct all of the information in pc by using

```
s <- cov(quas[, -1])
es <- eigen(s)
```

One may compare sqrt(es\$val) with pc\$sdev and es\$vec with pc\$load to verify that they are the same except for sign changes in some columns of pc\$load.

If one invokes the princomp command with cor=TRUE, then the eigen decomposition is performed on the correlation matrix, obtained via cor(quas[, -1]), rather than the covariance matrix. Which method is more appropriate in this case? To answer this question, let's examine the standard deviations of the columns of quas:

```
apply(quas[, -1], 2, sd)
```

Note that the variation of the R.A and Dec. columns is far larger than that of any other column. Thus, we should not be surprised if these two columns dominate the first two principal components. In fact, since these two columns together with z give position of the object, we might want to extract them from the principal components analysis altogether, retaining them unchanged in the reduced data set. However, we could essentially put all variables on an equal footing in terms of variability by using the correlation rather than the covariance (this is equivalent to standardizing each of the variables to have standard deviation equal to 1 before performing the princomp analysis). In the following development, we use essentially the first approach.

The two most important pieces of information in a principal components analysis are the variances explained (eigenvalues) and variable loadings (eigenvectors). The former may be viewed graphically using a technique called a [screeplot](#):

```
screeplot(pc)
```

In the above plot, we see that the variance of the first two PCs dwarfs the others. To see what this means, we must look at the loadings for these first two PCs:

```
loadings(pc)
```

This last command prints a lot of information. Scroll up to see the loadings of components 1 and 2, with

any loadings less than 0.1 in absolute value suppressed as unimportant. (In reality, the loadings for the first principal component are a vector of real numbers, scaled so that the sum of their squares equals 1. Each element in the vector gives the relative weight of the corresponding variable.)

To see *all* of the loadings for the first principal component (and only those), type

```
pc$load[,1]
```

We may conclude from the output above that the first principal component consists essentially of nothing other than R.A (recall that the standard deviation of R.A was much larger than that of the other variables, so this result is really not surprising).

It is also unsurprising to see that the second principal component consists almost entirely of the Declination:

```
pc$load[,2]
```

These two principal components together comprise over 99.8% of the total variance of these variables, which makes it difficult to see easily the effect of the remaining principal components. As explained earlier, one way to deal with the problem of variables on vastly different scales is by analyzing the correlation matrix rather than the covariance matrix. However, in this case, the two variables causing the trouble are easy to identify; thus, we'll proceed by performing a new principal components analysis on the remaining columns of quas after R.A and Dec. are removed:

```
pc2 <- princomp(quas[,-(1:3)])
screeplot(pc2)
```

In the new screeplot, we see three or four PCs with relatively large variance, one to four PCs with moderate variance, and the rest with relatively small variance. Let's see what the variable [loadings](#) for these first five PCs are:

```
loadings(pc2)
```

Again, it is necessary to scroll up to see the important output.

Examining these loadings, the first PC is somewhat difficult to interpret, but the second is basically an average of all the "_mag" variables. Notably, the three variables (u_mag, g_mag, r_mag) always occur with roughly the same weights in the first few PCs, indicating that we may replace these three with a single variable equal to their mean. The same is true of (i_mag, z_mag) and (J_mag, H_mag, K_mag). We could thus reduce these 8 variables to 3.

Another approach is to analyze only the principal component scores themselves, which are contained in pc\$scores. This 279x22 matrix contains exactly the same information as the original dataset, but the axes have been rotated so that the first axis is the most important in explaining information, followed by the second, etc. Based on our analysis, only 5 or 6 of these PCs should be very variable.

```
pairs(pc2$scores[,1:6], pch=".")
```

The drawback to the above plots, of course, is that many of them are difficult to interpret.

A [biplot](#) for a principal components analysis is a way of seeing both the PC scores and the factor loadings simultaneously.

```
biplot(pc2, choices=1:2)
```

In summary, principal components provides an objective way to decide, based on data alone, how to reduce the dimensionality of a dataset to ease interpretability. However, substantive astronomical knowledge should be at least as important as such considerations (e.g., if M_i is known to be important, then maybe it should be kept regardless of what PC analysis says).

Clustering via agglomerative nesting ([agnes](#))

We turn now to a few of the many methods of clustering. The goal of a clustering algorithm is to identify structure within a multivariate cloud of points by assigning each point to one of a small number of groups (some clustering algorithms don't provide specific assignments for each point but instead tell how likely each point is to belong to each group).

We will analyze the [Shapley galaxy dataset](#), which may be downloaded by typing

```
shap <- read.table("http://www.astrostatistics.psu.edu/datasets/Shapley_galaxy.dat", head=T)
```

Let us have a look:

```
dim(shap)
names(shap)
pairs(shap, pch=46)
```

It looks like we have to deal with some missing Mag observations in column 3:

```
shap[shap[,3]==0,3] <- NA
pairs(shap, pch=46)
```

Next, let's make a rough cut using the V variable and color those points red:

```
attach(shap)
a <- V>12000 & V<16000
pairs(shap, pch=46, col=1+a)
```

We'd like to search for clusters in space. Let's plot R.A against Dec and use different colors for different V values: black, red, green, and blue for V/1000 in (12,13), (13,14), (14,15), and (15,16), respectively.

```
plot(shap[a,1:2], pch=20, col=V[a]%/1000 - 12)
```

Now we begin to apply some clustering techniques. Most of these are contained in the cluster package, which must be loaded. We will consider the technique called [agnes](#) (agglomerative nesting) first.

```
library(cluster)
?agnes
```

There are two options of [agnes](#) that we care about: "stand" and "method". The first of these should be set to TRUE if we want [agnes](#) to standardize the variables before clustering. Let's decide whether this makes sense by checking the variability of each variable (first, we'll reduce the dataset to just those variables and quasars we want to use for the clustering):

```
shap2 <- shap[a,c(1,2,4)]
apply(shap2, 2, sd)
```

We see that because of the units used, the V variable has much higher variance than the other two variables. Therefore, if we were to apply [agnes](#) using "stand=FALSE", we would essentially be clustering only on V, which would not be very interesting. One solution here is to convert the (R.A, Dec., V) points into (x, y, z) points in Euclidean space. Another, slightly rougher, solution is simply to use "stand=TRUE", which is what we'll do here:

```
ag <- agnes(shap2, stand=TRUE)
```

Note that we have used the default value of "method", which is "average". Let's take a look at a dendrogram:

```
plot(ag, which=2)
```

You can see the plotting options for an R object of class "agnes" by reading the help file for [plot.agnes](#). You can also see what information is included in the ag object by looking at the help file for [agnes.object](#).

The dendrogram is hard to read near the leaves because there are 1629 observations in this dataset. To obtain the order in which the original observations must be rearranged to obtain the order seen in the dendrogram, look at `ag$order`. Since we don't really care about retaining the original order of the galaxies, let's simply reorder them as in the dendrogram:

```
shap2 <- shap2[ag$order, ]
```

In `ag$height`, there are 1628 values, one for each of the merges. (Each merge reduces the number of clusters by one, so in reducing from 1629 observations to 1 cluster we must have 1628 merges.) We can use these values to determine where to make cuts in the dataset: Whenever `ag$height[i]` is high enough, we should make a cut between observations i and $i+1$.

Let's try making a cut at a height of 4:

```
pltree(ag)
abline(col=2, h=4)
which(ag$hei>4)
abline(col=2, v=.5+
      which(ag$hei>4))
```

Although the four cuts identified by the `which` function result in 5 clusters, several of these clusters consist of only a few observations and could perhaps be lumped together with other clusters. On the other hand, using a height cutoff of 3.5 instead of 4 leads to four good-sized clusters:

```
which(ag$hei>3.5)
```

Let's produce a categorical variable for the clusters:

```
agclust <- cut(1:1629, lab=1:4,
               breaks <- c(0,188,1278,1535,1629))
table(agclust)
```

Now we may use this categorical variable to add color to a pairs plot:

```
pairs(shap2, pch=20,
      col=as.numeric(agclust))
```

Note that the `factor` `agclust` must be explicitly coerced to a numeric vector using `as.numeric`.

Above, we used the `method="average"` option. Two other commonly used options are "single", which tends to produce stringy clusters because two clusters are considered as close as their closest elements; and "complete", which is the opposite of "single" in the sense that two clusters are considered as close as the most distant elements.

There are many clustering/partitioning algorithms, far more than we can present here. One way to see the many options in R is to look at the list of functions for the [cluster package](#). There are also a couple of clustering algorithms in the standard R package, namely hierarchical clustering (`hclust`) and k-means clustering (`kmeans`).

Bootstrap

G. Jogesh Babu

Penn State University
<http://www.stat.psu.edu/~babu>

Director of Center for Astrostatistics

<http://astrostatistics.psu.edu>

Outline

- ① Motivation
- ② Simple statistical problem
- ③ Resampling
- ④ What is bootstrap
- ⑤ Regression
- ⑥ Fortran code
- ⑦ References

Motivation

- It is often relatively easy to devise a statistic (estimator of a parameter) that measures the property of interest, but is difficult or impossible to determine the distribution or variance (sampling variability) of that statistic.
- One might fit a parametric model to the dataset, yet not be able to assign confidence intervals to see how accurately the parameters are determined.
- In the past Statisticians concentrated estimators which have a simple closed form and which could be analyzed mathematically. Except for a few important but simple nonparametric statistics, these methods involve often unrealistic assumptions about the data; e.g. that it is generated from a Gaussian or exponential population.

Simple Statistical Problem

X_1, \dots, X_n are random variables from a distribution F with mean μ and variance σ^2 .

Sample mean $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$ estimates the population mean μ

Data vs. Sampling distribution of \bar{X}

Sampling (unknown) distribution of $\bar{X} - \mu$
 $G_n(x) = P(\bar{X} - \mu \leq x)$

If F is normal, then G_n is normal. Otherwise, for large n

$$G_n(x\sigma/\sqrt{n}) \approx \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{1}{2}y^2} dy$$

G_n may not be symmetric in the non-normal case.

How to improve the approximation?

Resampling

- Resampling methods help evaluate statistical properties using data rather than an assumed Gaussian or power law or other distributions.
- Resampling methods construct hypothetical ‘populations’ derived from the observed data, each of which can be analyzed in the same way to see how the statistics depend on plausible random variations in the data.
- Astronomers have often used *Monte Carlo methods* to simulate datasets from uniform or Gaussian populations. While helpful in some cases, this does not avoid the assumption of a simple underlying distribution.

Resampling

- Resampling the original data preserves (adaptively) whatever distributions are truly present, including selection effects such as truncation (flux limits or saturation).
- Resampling procedure is a Monte Carlo method of simulating datasets from an existing dataset, without any assumption on the underlying population.
- Resampling procedures are supported by solid theoretical foundation.

What is Bootstrap

$\mathbf{X} = (X_1, \dots, X_n)$ - a sample from F

$\mathbf{X}^* = (X_1^*, \dots, X_n^*)$ - a simple random sample from the data.

$\hat{\theta}$ is an estimator of θ

θ^* is based on X_i^*

Examples:

$$\hat{\theta} = \bar{X}, \quad \theta^* = \bar{X}^*$$

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2, \quad \theta^* = \frac{1}{n} \sum_{i=1}^n (X_i^* - \bar{X}^*)^2$$

$\theta^* - \hat{\theta}$ behaves like $\hat{\theta} - \theta$

Correlation Coefficient

Sample correlation coefficient $\hat{\rho}$

$$\hat{\rho} = \frac{\frac{1}{n} \sum_{i=1}^n (X_i Y_i - \bar{X}_n \bar{Y}_n)}{\sqrt{\left(\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X}_n)^2\right) \left(\frac{1}{n} \sum_{i=1}^n (Y_i - \bar{Y}_n)^2\right)}}$$

Its bootstrap version

$$\rho^* = \frac{\frac{1}{n} \sum_{i=1}^n (X_i^* Y_i^* - \bar{X}_n^* \bar{Y}_n^*)}{\sqrt{\left(\frac{1}{n} \sum_{i=1}^n (X_i^* - \bar{X}_n^*)^2\right) \left(\frac{1}{n} \sum_{i=1}^n (Y_i^* - \bar{Y}_n^*)^2\right)}}$$

Statistical inference requires sampling distribution, G_n given by $G_n(x) = P(T_n \leq x)$

$$\begin{array}{ll} T_n & T_n^* \\ \sqrt{n}(\bar{X} - \mu)/\sigma & \sqrt{n}(\bar{X}^* - \bar{X})/s_n \\ \sqrt{n}(\bar{X} - \mu)/s_n & \sqrt{n}(\bar{X}^* - \bar{X})/s_n^* \end{array}$$

where $s_n^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$ and $s_n^{*2} = \frac{1}{n} \sum_{i=1}^n (X_i^* - \bar{X}^*)^2$

Bootstrap distribution (Histogram) G_B given the data
 $G_B(x) = P(T_n^* \leq x | \mathbf{X})$

$G_n(x) \approx G_B(x)$, G_B is completely known

Example

G_n denotes the sampling distribution of $\sqrt{n}(\bar{X} - \mu)/\sigma$

$$G_n(x) = P(\sqrt{n}(\bar{X} - \mu)/\sigma \leq x)$$

G_B is the corresponding *bootstrap distribution* (Histogram)
given the data

$$G_B(x) = P(\sqrt{n}(\bar{X}^* - \bar{X})/s_n \leq x | \mathbf{X})$$

$$G_n(x) \approx G_B(x),$$

G_B is completely known

Bootstrap Distribution

$M = n^n$ bootstrap samples possible

$$\begin{array}{ll} X_1^{*(1)}, \dots, X_n^{*(1)} & r_1 = \sqrt{n}(\bar{X}^{*(1)} - \bar{X})/s_n \\ X_1^{*(2)}, \dots, X_n^{*(2)} & r_2 = \sqrt{n}(\bar{X}^{*(2)} - \bar{X})/s_n \\ \ddots & \ddots \quad \ddots \\ X_1^{*(M)}, \dots, X_n^{*(M)} & r_M = \sqrt{n}(\bar{X}^{*(M)} - \bar{X})/s_n \end{array}$$

Frequency table or histogram based on r_1, \dots, r_M gives G_B

For $n = 10$ data points, $M = \text{ten billion}$

$N \sim n(\log n)^2$ bootstrap replications suffice

– Babu and Singh (1983) Ann Stat

Confidence Intervals

Compute

$$\sqrt{n}(\bar{X}^{*(j)} - \bar{X})/s_n$$

for N bootstrap samples

Arrange them in increasing order

$$r_1 < r_2 < \dots < r_N \quad k = [0.05N], \quad m = [0.95N]$$

90% Confidence Interval for μ is

$$\bar{X} - r_m \frac{s_n}{\sqrt{n}} \leq \mu < \bar{X} - r_k \frac{s_n}{\sqrt{n}}$$

Bootstrap at its best

Pearson's correlation coefficient

$$\hat{\rho} = \frac{\frac{1}{n} \sum_{i=1}^n (X_i Y_i - \bar{X} \bar{Y})}{\sqrt{\left(\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2\right) \left(\frac{1}{n} \sum_{i=1}^n (Y_i - \bar{Y})^2\right)}}$$

Smooth function model

$$\hat{\rho} = H(\bar{\mathbf{Z}}), \text{ where } \mathbf{Z}_i = (X_i Y_i, X_i^2, Y_i^2, X_i, Y_i)$$

$$\begin{aligned} H(a_1, a_2, a_3, a_4, a_5) &= \frac{(a_1 - a_4 a_5)}{\sqrt{((a_2 - a_4^2)(a_3 - a_5^2))}} \\ \mathbf{Z}_i^* &= (X_i^* Y_i^*, X_i^{*2}, Y_i^{*2}, X_i^*, Y_i^*) \\ \rho^* &= H(\bar{\mathbf{Z}}^*) \end{aligned}$$

Studentization

Functions of random variables or statistics are often normalized (divided) by the standard deviation to make these units free. When standard deviations are estimated, the normalization is known as studentization.

$$\begin{aligned} t_n &= \sqrt{n}(H(\bar{\mathbf{Z}}) - H(E(\mathbf{Z}_1))) / \hat{\sigma}_n \\ t_n^* &= \sqrt{n}(H(\bar{\mathbf{Z}}^*) - H(\bar{\mathbf{Z}})) / \sigma_n^* \end{aligned}$$

$\hat{\sigma}_n^2 = \ell'(\bar{\mathbf{Z}}) \Sigma_n \ell(\bar{\mathbf{Z}})$ and $\sigma_n^{*2} = \ell'(\bar{\mathbf{Z}}^*) \Sigma_n^* \ell(\bar{\mathbf{Z}}^*)$ are the variances of the numerators.

$\ell = \partial H$ vector of first partial derivatives of H

Σ_n sample dispersion of \mathbf{Z}

Σ_n^* dispersion of bootstrap sample \mathbf{Z}^*

$\hat{\theta} = H(\bar{\mathbf{Z}})$ is an estimator of the parameter $\theta = H(E(\mathbf{Z}_1))$

Randomly choose $N \sim n(\log n)^2$ bootstrap samples

Compute $t_n^{*(j)}$ for each

Arrange them in increasing order

$u_1 < u_2 < \dots < u_N \quad k = [0.05N], m = [0.95N]$

90% Confidence Interval for the parameter θ is

$$\hat{\theta} - u_m \frac{\hat{\sigma}_n}{\sqrt{n}} \leq \theta < \hat{\theta} - r_k \frac{\hat{\sigma}_n}{\sqrt{n}}$$

This is called bootstrap PERCENTILE - t confidence interval

Theory

Under $\ell(\bar{\mathbf{Z}}) \neq 0$

$P(t_n \leq x) = \Phi(x) + \frac{1}{\sqrt{n}} p(x) \phi(x) + \text{error}$

$P^*(t_n^* \leq x) = \Phi(x) + \frac{1}{\sqrt{n}} p_n(x) \phi(x) + \text{error}$

$\sqrt{n}|P(t_n \leq x) - P^*(t_n^* \leq x)| \rightarrow 0$

\hat{F}_n an estimator of F we could Bootstrap from \hat{F}_n

If $F \sim N(\mu, \sigma^2)$, then $\hat{F}_n \sim N(\hat{\mu}, \hat{\sigma}^2)$

Same theory works.

- Babu and Singh (1983) Ann Stat
- Babu and Singh (1984) Sankhyā
- Babu and Singh (1990) Scand J. Stat

When does bootstrap work well

- 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

When does Bootstrap fail

- $\hat{\theta} = \max_{1 \leq i \leq n} X_i$ non-smooth statistic
– Bickel and Freedman (1981) Ann. Stat.
- $\hat{\theta} = \bar{X}$ and $E\bar{X}_1^2 = \infty$ heavy tails
– Babu (1984) Sankhyā
– Athreya (1987) Ann. Stat.
- $\hat{\theta} - \theta = H(\bar{\mathbf{Z}}) - H(E(\mathbf{Z}_1))$ and $\partial H(E(\mathbf{Z}_1)) = 0$
Limit distribution is like linear combinations of Chi-squares. But here a modified version works
– Babu (1984) Sankhyā.

Linear Regression

$$Y_i = \alpha + \beta X_i + \epsilon_i$$

$$\mathbb{E}(\epsilon_i) = 0 \text{ and } \text{Var}(\epsilon_i) = \sigma_i^2$$

Least squares estimators of β and α

$$\hat{\beta} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2}$$

$$\hat{\alpha} = \bar{Y} - \hat{\beta}\bar{X}$$

$$\text{Var}(\hat{\beta}) = \frac{\sum_{i=1}^n (X_i - \bar{X})^2 \sigma_i^2}{L_n^2}$$

$$L_n = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Classical Bootstrap

Estimate the residuals $e_i = Y_i - \hat{\alpha} - \hat{\beta}X_i$
 $\hat{e}_i = e_i - \frac{1}{n} \sum_{j=1}^n e_j$

Draw e_1^*, \dots, e_n^* from $\hat{e}_1, \dots, \hat{e}_n$

Bootstrap estimators

$$\beta^* = \hat{\beta} + \frac{\sum_{i=1}^n (X_i - \bar{X})(e_i^* - \bar{e}^*)}{\sum_{i=1}^n (X_i - \bar{X})^2}$$

$$\alpha^* = \hat{\alpha} + (\hat{\beta} - \beta^*)\bar{X} + \bar{e}^*$$

$$E_B(\beta^* - \hat{\beta})^2 \approx \text{Var}(\hat{\beta})$$

Efficient if $\sigma_i = \sigma$

V_B does not approximate the variance of $\hat{\beta}$ under heteroscedasticity (i.e. unequal variances σ_i)

Paired Bootstrap

Resample the pairs $(X_1, Y_1), \dots, (X_n, Y_n)$
 $(\tilde{X}_1, \tilde{Y}_1), \dots, (\tilde{X}_n, \tilde{Y}_n)$

$$\tilde{\beta} = \frac{\sum_{i=1}^n (\tilde{X}_i - \bar{\tilde{X}})(\tilde{Y}_i - \bar{\tilde{Y}})}{\sum_{i=1}^n (\tilde{X}_i - \bar{\tilde{X}})^2}, \quad \tilde{\alpha} = \bar{\tilde{Y}} - \tilde{\beta} \bar{\tilde{X}}$$

Repeat the resampling N times and get

$$\beta_{PB}^{(1)}, \dots, \beta_{PB}^{(N)}$$

$$\frac{1}{N} \sum_{i=1}^N (\beta_{PB}^{(i)} - \hat{\beta})^2 \approx \text{Var}(\hat{\beta})$$

even when not all σ_i are the same

FORTRAN code

```
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
```

FORTRAN code illustrating the paired bootstrap resampling
for a two dimensional dataset $(x_i, y_i), i = 1, \dots, N$.

Comparison

- **The Classical Bootstrap**

- Efficient when $\sigma_i = \sigma$
- But inconsistent when σ_i 's differ

- **The Paired Bootstrap**

- Robust against heteroscedasticity
- Works well even when σ_i are all different

References

G. J. Babu and C. R. Rao (1993). *Bootstrap Methodology*, Handbook of Statistics, Vol 9, Chapter 19.

Michael R. Chernick (2007). *Bootstrap Methods - A guide for Practitioners and Researchers*, (2nd Ed.) Wiley Inter-Science.

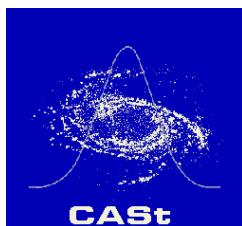
Abdelhak M. Zoubir and D. Robert Iskander (2004). *Bootstrap Techniques for Signal Processing*, Cambridge University Press.

It is a handbook on 'bootstrap' for engineers, to analyze complicated data with little or no model assumptions. Bootstrap has found many applications including, artificial neural networks, biomedical engineering, environmental engineering, image processing, and Radar and sonar signal processing. Majority of the applications are taken from signal processing literature.

Bootstrap for Goodness of Fit

G. Jogesh Babu
Center for Astrostatistics

<http://astrostatistics.psu.edu>

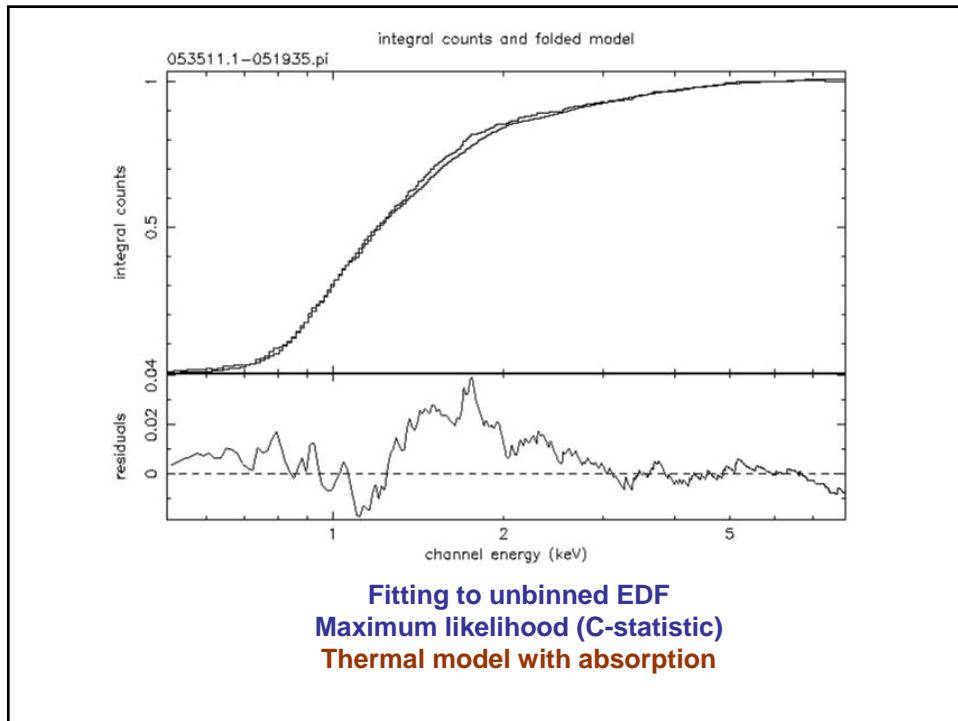
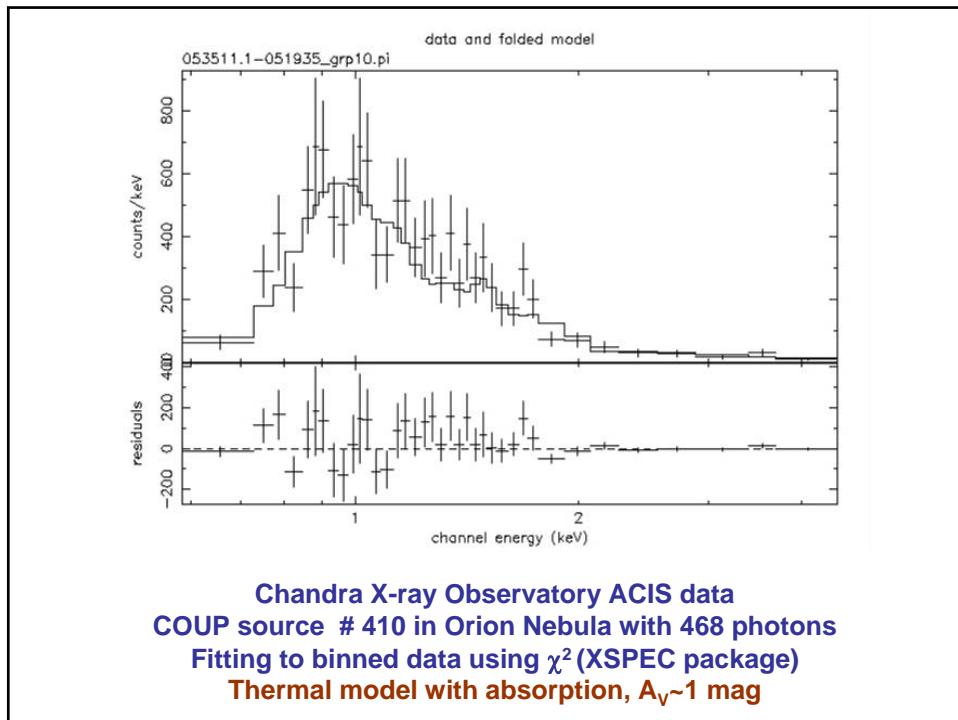


VOSTAT

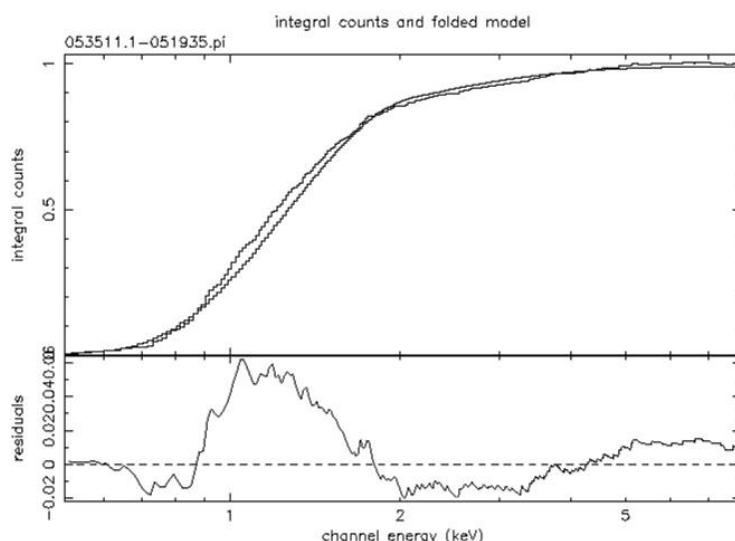
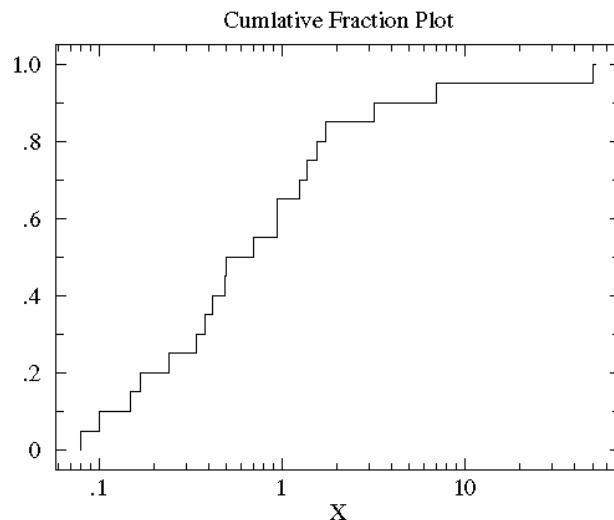
Astrophysical Inference from astronomical data

Fitting astronomical data

- Non-linear regression
- Density (shape) estimation
- Parametric modeling
 - Parameter estimation of assumed model
 - Model selection to evaluate different models
 - Nested (in quasar spectrum, should one add a broad absorption line BAL component to a power law continuum)
 - Non-nested (is the quasar emission process a mixture of blackbodies or a power law?)
- Goodness of fit



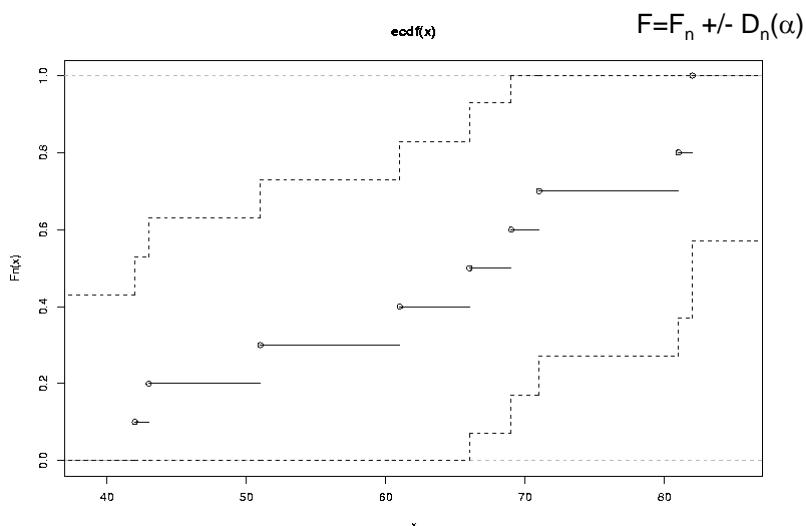
Empirical Distribution Function



Incorrect model family
Power law model, absorption $A_V \sim 1$ mag

Question : Can a power law model be excluded with 99% confidence?

K-S Confidence bands



Model fitting

Find most parsimonious ‘best’ fit to answer:

- Is the underlying nature of an X-ray stellar spectrum a non-thermal power law or a thermal gas with absorption?
- Are the fluctuations in the cosmic microwave background best fit by Big Bang models with dark energy or with quintessence?
- Are there interesting correlations among the properties of objects in any given class (e.g. the Fundamental Plane of elliptical galaxies), and what are the optimal analytical expressions of such correlations?

Statistics Based on EDF

Kolmogorov-Smirnov: $\sup_x |F_n(x) - F(x)|$,
 $\sup_x (F_n(x) - F(x))^+$, $\sup_x (F_n(x) - F(x))^-$

Cramer - van Mises: $\int (F_n(x) - F(x))^2 dF(x)$

Anderson - Darling: $\int \frac{(F_n(x) - F(x))^2}{F(x)(1-F(x))} dF(x)$

All of these statistics are distribution free

Nonparametric statistics.

But they are no longer distribution free if the parameters are estimated or the data is multivariate.

Table I. Limiting Distribution of the Kolmogorov Smirnov Statistic
(from Smirnov (1948))

x	$L(x)$	x	$L(x)$	x	$L(x)$	x	$L(x)$
0.28	0.000001	0.73	0.399113	1.18	0.876548	1.76	0.995922
0.29	0.000004	0.74	0.395981	1.19	0.882528	1.78	0.995460
0.30	0.000010	0.75	0.392849	1.20	0.888507	1.80	0.995028
0.31	0.000021	0.76	0.389640	1.21	0.893030	1.82	0.997246
0.32	0.000046	0.77	0.406374	1.22	0.898104	1.84	0.997707
0.33	0.000091	0.78	0.423002	1.23	0.902972	1.86	0.998023
0.34	0.000171	0.79	0.439505	1.24	0.907648	1.88	0.998297
0.35	0.000303	0.80	0.455857	1.25	0.912132	1.90	0.998526
0.36	0.000606	0.81	0.471970	1.26	0.916432	1.92	0.998744
0.37	0.000986	0.82	0.486230	1.27	0.920704	1.94	0.998944
0.38	0.001285	0.83	0.502808	1.28	0.924602	1.96	0.999079
0.39	0.001582	0.84	0.518366	1.29	0.928288	1.98	0.999213
0.40	0.002080	0.85	0.534692	1.30	0.931908	2.00	0.999359
0.41	0.002972	0.86	0.549744	1.31	0.935370	2.02	0.999428
0.42	0.00476	0.87	0.564546	1.32	0.938682	2.04	0.999516
0.43	0.007277	0.88	0.579707	1.33	0.941948	2.06	0.999588
0.44	0.011002	0.89	0.593316	1.34	0.944972	2.08	0.999650
0.45	0.015980	0.90	0.606636	1.35	0.947992	2.10	0.999705
0.46	0.016005	0.91	0.620928	1.36	0.950512	2.12	0.999750
0.47	0.020222	0.92	0.634286	1.37	0.953142	2.14	0.999790
0.48	0.024682	0.93	0.647330	1.38	0.955650	2.16	0.999822
0.49	0.030017	0.94	0.660085	1.39	0.958040	2.18	0.999852
0.50	0.036265	0.95	0.672546	1.40	0.960318	2.20	0.999874
0.51	0.042914	0.96	0.684658	1.41	0.962562	2.22	0.999896
0.52	0.050306	0.97	0.696444	1.42	0.964962	2.24	0.999912
0.53	0.058304	0.98	0.707940	1.43	0.966516	2.26	0.999926
0.54	0.067497	0.99	0.719126	1.44	0.968952	2.28	0.999940
0.55	0.077183	1.00	0.730000	1.45	0.970158	2.30	0.999949
0.56	0.087577	1.01	0.740566	1.46	0.971846	2.32	0.999958
0.57	0.098656	1.02	0.750566	1.47	0.973448	2.34	0.999965
0.58	0.110365	1.02	0.760746	1.48	0.974979	2.36	0.999970
0.59	0.122965	1.04	0.770434	1.49	0.976417	2.38	0.999976
0.60	0.135718	1.05	0.779794	1.50	0.977782	2.40	0.999980
0.61	0.149229	1.06	0.789860	1.52	0.980310	2.42	0.999984
0.62	0.163226	1.07	0.797636	1.54	0.982578	2.44	0.999987
0.63	0.177753	1.08	0.806128	1.56	0.984610	2.46	0.999991
0.64	0.192677	1.09	0.814342	1.58	0.986426	2.48	0.999991
0.65	0.207907	1.10	0.822464	1.60	0.988149	2.50	0.999992
0.66	0.222937	1.11	0.830860	1.62	0.989492	2.55	0.999996
0.67	0.239502	1.12	0.837356	1.64	0.990777	2.60	0.999997
0.68	0.255790	1.13	0.844502	1.66	0.991917	2.65	0.999998
0.69	0.272189	1.14	0.851394	1.68	0.992328	2.70	0.999999
0.70	0.288766	1.15	0.858038	1.70	0.993823	2.80	0.999997
0.71	0.305471	1.16	0.864442	1.72	0.994612	2.90	0.999999
0.72	0.322385	1.17	0.870612	1.74	0.995309	3.00	0.999999

KS Probabilities are invalid when the model parameters are estimated from the data. Astronomers often used them incorrectly.

(Lillifors 1964)

Multivariate Case

Warning: K-S does not work in multidimensions

Example – Paul B. Simpson (1951)

$$F(x,y) = ax^2 y + (1 - a) y^2 x, \quad 0 < x, y < 1$$

(X_1, Y_1) data from F , F_1 EDF of (X_1, Y_1)

$$\begin{aligned} P(|F_1(x,y) - F(x,y)| < 0.72, \text{ for all } x, y) \text{ is} \\ &> 0.065 \text{ if } a = 0, \quad (F(x,y) = y^2 x) \\ &< 0.058 \text{ if } a = 0.5, \quad (F(x,y) = xy(x+y)/2) \end{aligned}$$

Numerical Recipe's treatment of a 2-dim KS test is mathematically invalid.

Processes with estimated Parameters

$\{F(\cdot; \theta) : \theta \in \Theta\}$ - a family of distributions

X_1, \dots, X_n sample from F

Kolmogorov-Smirnov, Cramer-von Mises etc.,
when θ is estimated from the data, are
Continuous functionals of the empirical process

$$Y_n(x; \theta_n) = \sqrt{n} (F_n(x) - F(x; \theta_n))$$

In the Gaussian case,

$$\theta = (\mu, \sigma^2) \text{ and } \theta_n = (\bar{X}, s_n^2)$$

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

$$s_n^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$$

Bootstrap

G_n is an estimator of F , based on X_1, \dots, X_n

X_1^*, \dots, X_n^* i.i.d. from G_n

$$\theta_n^* = \theta_n(X_1^*, \dots, X_n^*)$$

$F(\cdot; \theta)$ is Gaussian with $\theta = (\mu, \sigma^2)$

and $\theta_n = (\bar{X}, s_n^2)$, then $\theta_n^* = (\bar{X}_n^*, s_n^{*2})$

Parametric bootstrap if $G_n = F(\cdot; \theta_n)$

X_1^*, \dots, X_n^* i.i.d. from $F(\cdot; \theta_n)$

Nonparametric bootstrap if $G_n = F_n$ (EDF)

Parametric Bootstrap

X_1^*, \dots, X_n^* sample generated from $F(\cdot; \theta_n)$.

In Gaussian case $\theta_n^* = (\bar{X}_n^*, s_n^{*2})$.

Both $\sqrt{n} \sup_x |F_n(x) - F(x; \theta_n)|$ and

$\sqrt{n} \sup_x |F_n^*(x) - F(x; \theta_n^*)|$

have the same limiting distribution

(In the XSPEC packages, the parametric bootstrap is command FAKEIT, which makes Monte Carlo simulation of specified spectral model)

Nonparametric Bootstrap

X_1^*, \dots, X_n^* i.i.d. from F_n .

A bias correction

$B_n(x) = F_n(x) - F(x; \theta_n)$

is needed.

$\sqrt{n} \sup_x |F_n(x) - F(x; \theta_n)|$ and

$\sqrt{n} \sup_x |F_n^*(x) - F(x; \theta_n^*) - B_n(x)|$

have the same limiting distribution

(XSPEC does not provide a nonparametric bootstrap capability)

- **Chi-Square type statistics** – (Babu, 1984, Statistics with linear combinations of chi-squares as weak limit. *Sankhya, Series A*, **46**, 85-93.)
- **U-statistics** – (Arcones and Giné, 1992, On the bootstrap of U and V statistics. *Ann. of Statist.*, **20**, 655–674.)

Confidence limits under misspecification of model family

X_1, \dots, X_n data from unknown H .

H may or may not belong to the family $\{F(\cdot; \theta) : \theta \in \Theta\}$.

H is closest to $F(\cdot; \theta_0)$, in Kullback - Leibler information

$$\int h(x) \log (h(x)/f(x; \theta)) dv(x) \geq 0$$

$$\int h(x) |\log (h(x))| dv(x) < \infty$$

$$\int h(x) \log f(x; \theta_0) dv(x) = \max_{\theta} \int h(x) \log f(x; \theta) dv(x)$$

For any $0 < \alpha < 1$,

$$P(\sqrt{n} \sup_x |F_n(x) - F(x; \theta_n) - (H(x) - F(x; \theta_0))| < C_\alpha) \rightarrow \alpha$$

C_α is the α -th quantile of

$$\sqrt{n} \sup_x |F_n^*(x) - F(x; \theta_n^*) - (F_n(x) - F(x; \theta_n))|$$

This provide an estimate of the distance between the true distribution and the family of distributions under consideration.

References

- G. J. Babu and C. R. Rao (1993). Handbook of Statistics, Vol 9, Chapter 19.
- G. J. Babu and C. R. Rao (2003). Confidence limits to the distance of the true distribution from a misspecified family by bootstrap. *J. Statist. Plann. Inference* **115**, 471-478.
- G. J. Babu and C. R. Rao (2004). Goodness-of-fit tests when parameters are estimated. *Sankhya, Series A*, **66** (2004) no. 1, 63-74.

Hypothesis Testing and Bootstrapping

This tutorial demonstrates some of the many statistical tests that R can perform. It is impossible to give an exhaustive list of such testing functionality, but we hope not only to provide several examples but also to elucidate some of the logic of statistical hypothesis tests with these examples.

T tests

In the [EDA and regression](#) tutorial, we used exploratory techniques to identify 92 stars from the [Hipparcos](#) data set that are associated with the Hyades. We did this based on the values of right ascension, declination, principal motion of right ascension, and principal motion of declination. We then excluded one additional star with a large error of parallax measurement:

```
hip <- read.table("http://astrostatistics.psu.edu/datasets/HIP_star.dat",
  header=T,fill=T)
attach(hip)
filter1 <- (RA>50 & RA<100 & DE>0 & DE<25)
filter2 <- (pmRA>90 & pmRA<130 & pmDE>-60 & pmDE< -10)
filter <- filter1 & filter2 & (e_Plx<5)
sum(filter)
```

In this section of the tutorial, we will compare these Hyades stars with the remaining stars in the Hipparcos dataset on the basis of the color (B minus V) variable. That is, we are comparing the groups in the boxplot below:

```
color <- B.V
boxplot(color~filter,notch=T)
```

For ease of notation, we define vectors H and nH (for "Hyades" and "not Hyades") that contain the data values for the two groups.

```
H <- color[filter]
nH <- color[!filter & !is.na(color)]
```

In the definition of nH above, we needed to exclude the NA values.

A two-sample t-test may now be performed with a single line:

```
t.test(H,nH)
```

Because it is instructive and quite easy, we may obtain the same results without resorting to the [t.test](#) function. First, we calculate the variances of the sample means for each group:

```
v1 <- var(H)/92
v2 <- var(nH)/2586
c(var(H),var(nH))
```

The t statistic is based on the standardized difference between the two sample means. Because the two samples are assumed independent, the variance of this difference equals the sum of the individual variances (i.e., $v_1 + v_2$). Nearly always in a two-sample t-test, we wish to test the null hypothesis that the true difference in means equals zero. Thus, standardizing the difference in means involves subtracting zero and then dividing by the square root of the variance:

```
tstat <- (mean(H)-mean(nH))/sqrt(v1+v2)
tstat
```

To test the null hypothesis, this t statistic is compared to a t distribution. In a Welch test, we assume that the variances of the two populations are not necessarily equal, and the degrees of freedom of the t

distribution are computed using the so-called Satterthwaite approximation:

$$(v1 + v2)^2 / (v1^2/91 + v2^2/2585)$$

The two-sided p-value may now be determined by using the cumulative distribution function of the t distribution, which is given by the [pt](#) function.

$$2 * pt(tstat, 97.534)$$

Incidentally, one of the assumptions of the t-test, namely that each of the two underlying populations is normally distributed, is almost certainly not true in this example. However, because of the central limit theorem, the t-test is robust against violations of this assumption; even if the populations are not roughly normally distributed, the sample means are.

In this particular example, the Welch test is probably not necessary, since the sample variances are so close that an assumption of equal variances is warranted. Thus, we might conduct a slightly more restrictive t-test that assumes equal population variances. Without going into the details here, we merely present the R output:

$$t.test(H, nH, var.equal=T)$$

Permutation tests

Let's look at another example in which a t-test might be warranted. The [globular cluster luminosity dataset](#) gives measurements for two different galaxies, Milky Way galaxy (MWG) and Andromeda galaxy (M31). The t-test gives us a very simplistic way of comparing these galaxies, namely, by the mean luminosity of their globular clusters. Because the apparent magnitudes for M31 are offset by the distance modulus of 24.44, the t-test should compare the difference in sample means with 24.44 instead of the usual zero:

$$\begin{aligned} gc &\leftarrow \text{read.csv("http://astrostatistics.psu.edu/datasets/glob_clus.csv") } \\ t.test(lum~gal, data=gc, mu=24.44) \end{aligned}$$

As before, the t-statistic here (1.6254) is accompanied by a p-value (0.1074). This p-value may be interpreted as follows: IF the two samples really are representative samples from populations whose means differ by 24.44, THEN the probability of obtaining a t-statistic at least as far from zero as 1.6254 would be roughly 0.1074. Since most people don't consider 0.1074 to be a very small probability, the conclusion here is that there is not strong statistical evidence of a difference in true means other than 24.44.

The p-value of 0.1074 is an approximation calculated under certain distributional assumptions on the populations. An alternative method for calculating a p-value corresponding to the t-statistic of 1.6254 is called a permutation test. Conceptually, the permutation test follows directly from the definition (given in the preceding paragraph) of a p-value. Let us assume that the luminosities for the MWG globular clusters are distributed exactly like the luminosities for the M31 globular clusters, except that the latter are all shifted down by 24.44. Under this assumption (called the *null hypothesis*), we may add 24.44 to each of the M31 luminosities and then the combined samples are equivalent to one large sample from the common distribution. In other words, if the null hypothesis is true and we randomly permute the labels (81 instances of MWG and 360 of M31), then the original sample is no different than the permuted sample.

By definition, then, the p-value should be equal to the proportion of all possible t-statistics resulting from label permutations that are at least as extreme as the observed t-statistic (1.6254). Since there are far too many such permutations (roughly 10^{90}) for us to make this exact calculation, we will have to settle for a random sample of permutations:

$$\begin{aligned} tlist &\leftarrow 1:5000 \\ lum &\leftarrow gc[, "lum"] \\ gal &\leftarrow gc[, "gal"] \\ lum[gal=="M31"] &\leftarrow lum[gal=="M31"] - 24.44 \end{aligned}$$

```

for(i in 1:5000) {
  s <- sample(441,81) # choose a sample
  newtstat <- t.test(lum[s], lum[-s])$stat
  tlist[i] <- newtstat # add new null t-stat to list
}

```

Note: The above code is *not* built for speed!

By definition, the p-value is the probability of obtaining a test statistic more extreme than the observed test statistic under the null hypothesis. Let's take a look at the null distribution of the t-statistic we just calculated, along with the observed value:

```

hist(tlist)
abline(v=c(-1,1)*1.6254,lty=2,col=2)
sum(abs(tlist)>=1.6254)/5000

```

This p-value is quite close to the 0.1074 obtained earlier.

Empirical distribution functions

Suppose we are curious about whether a given sample comes from a particular distribution. For instance, how normal is the random sample 'tlist' of t-statistics obtained under the null hypothesis in the previous example? How normal (say) are the colors of 'H' and 'nH'?

A simple yet very powerful graphical device is called a Q-Q plot, in which some quantiles of the sample are plotted against the same quantiles of whatever distribution we have in mind. If a roughly straight line results, this suggests that the fit is good. Roughly, a pth quantile of a distribution is a value such that a proportion p of the distribution lies below that value.

A Q-Q plot for normality is so common that there is a separate function, [qqnorm](#), that implements it. (Normality is vital in statistics not merely because many common populations are normally distributed -- which is actually not true in astronomy -- but because the central limit theorem guarantees the approximate normality of sample means.)

```

par(mfrow=c(2,2))
qqnorm(tlist,main="Null luminosity t statistics")
abline(0,1,col=2)
qqnorm(H,main="Hyades")
qqnorm(nH,main="non-Hyades")

```

Not surprisingly, the tlist variable appears extremely nearly normally distributed (more precisely, it is nearly *standard* normal, as evidenced by the proximity of the Q-Q plot to the line x=y, shown in red). As for H and nH, the distribution of B minus V exhibits moderate non-normality in each case.

In the bottom right corner of the plotting window, let's reconstruct the Q-Q plot from scratch for tlist. This is instructive because the same technique may be applied to any comparison distribution, not just normal. If we consider the 5000 entries of tlist in increasing order, let's call the ith value the $((2i-1)/10000)$ th quantile for all i from 1 to 5000. We merely graph this point against the corresponding quantile of standard normal:

```

plot(qnorm((2*(1:5000)-1)/10000),sort(tlist))
par(mfrow=c(1,1)) # reset plotting window

```

Related to the Q-Q plot is a distribution function called the *empirical (cumulative) distribution function*, or EDF. (In fact, the EDF is almost the same as a Q-Q plot against a uniform distribution.) The EDF is, by definition, the cumulative distribution function for the discrete distribution represented by the sample itself -- that is, the distribution that puts mass 1/n on each of the n sample points. We may graph the EDF using the [ecdf](#) function:

```
plot(ecdf(tlist))
```

While it is generally very difficult to interpret the EDF directly, it is possible to compare an EDF to a theoretical cumulative distribution function or two another EDF. Among the statistical tests that implement such a comparison is the Kolmogorov-Smirnov test, which is implemented by the R function [ks.test](#).

```
ks.test(tlist, "pnorm")
ks.test(H, nH)
```

Whereas the first result above gives a surprisingly small p-value, the second result is not surprising; we already saw that H and nH have statistically significantly different means. However, if we center each, we obtain

```
ks.test(H-mean(H), nH-mean(nH))
```

In other words, the Kolmogorov-Smirnov test finds no statistically significant evidence that the distribution of B.V for the Hyades stars is anything other than a shifted version of the distribution of B.V for the other stars. We can perform the same test for the luminosities of globular clusters from both the Milky Way and Andromeda:

```
lum.mwg <- gc[gal=="MWG", "lum"]
lum.m31 <- gc[gal=="M31", "lum"]
ks.test(lum.mwg, lum.m31-24.44)
```

The above test does give a statistically significant difference. The K-S test tests whether the M31 dataset, shifted by 24.44, comes from the same distribution as the MWG dataset. The t-test, on the other hand, only tests whether these distributions have the same mean.

Chi-squared tests for categorical data

We begin with a plot very similar to one seen in the [EDA and regression](#) tutorial:

```
bvcat <- cut(color, breaks=c(-Inf,.5,.75,1,Inf))
boxplot(Vmag~bvcat, varwidth=T,
        ylim=c(max(Vmag),min(Vmag)),
        xlab=expression("B minus V"),
        ylab=expression("V magnitude"),
        cex.lab=1.4, cex.axis=.8)
```

The cut values for bvcat are based roughly on the quartiles of the B minus V variable. We have created, albeit artificially, a second categorical variable ("filter", the Hyades indicator, is the first). Here is a summary of the dataset based only on these two variables:

```
table(bvcat, filter)
```

Note that the Vmag variable is irrelevant in the table above.

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, filter)
```

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 p-value produced above is based on the fact that the chi-squared statistic is approximately distributed like a true chi-squared distribution (on 3 degrees of freedom, in this case) if the null hypothesis is true. However, it is possible to obtain exact p-values, if one wishes to calculate the chi-squared statistic for all possible tables of counts with the same row and column sums as the given table. Since this is rarely

practical computationally, the exact p-value may be approximated using a Monte Carlo method (just as we did earlier for the permutation test). Such a method is implemented in the [chisq.test](#) function:

```
chisq.test(bvcat,filter,sim=T,B=50000)
```

The two different p-values we just generated a numerically similar but based on entirely different mathematics. The difference may be summed up as follows: The first method produces the exact value of an approximate p-value, whereas the second method produces an approximation to the exact p-value!

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 a chi-squared test of goodness-of-fit.

Nonparametric bootstrapping of regression standard errors

Let us consider a linear model for the relationship between DE and pmDE among the 92 Hyades stars:

```
x <- DE[filter]
y <- pmDE[filter]
plot(x,y,pch=20)
model1 <- lm(y ~ x)
abline(model1,lwd=2,col=2)
```

The red line on the plot is the usual least-squares line, for which estimation is easy and asymptotic theory gives easy-to-calculate standard errors for the coefficients:

```
summary(model1)$coef
```

However, suppose we wish to use a resistant regression method such as [lqs](#).

```
library(MASS)
model2 <- lqs(y ~ x)
abline(model2,lwd=2,col=3)
model2
```

In this case, it is not so easy to obtain standard errors for the coefficients. Thus, we will turn to bootstrapping. In a standard, or nonparametric, bootstrap, we repeatedly draw samples of size 92 from the empirical distribution of the data, which in this case consist of the (DE, pmDE) pairs. We use [lqs](#) to fit a line to each sample, then compute the sample covariance of the resulting coefficient vectors. The procedure works like this:

```
model2B <- matrix(0,200,2)
for (i in 1:200) {
  s <- sample(92,replace=T)
  model2B[i,] <- lqs(y[s]~x[s])$coef
}
```

We may now find the sample covariance matrix for model2B. The (marginal) standard errors of the coefficients are obtained as the square roots of the diagonal entries of this matrix:

```
cov(model2B)
se <- sqrt(diag(cov(model2B)))
se
```

The logic of the bootstrap procedure is that we are estimating an approximation of the true standard errors. The approximation involves replacing the true distribution of the data (unknown) with the empirical distribution of the data. This approximation may be estimated with arbitrary accuracy by a Monte Carlo approach, since the empirical distribution of the data is known and in principle we may sample from it as many times as we wish. In other words, as the bootstrap sample size increases, we get a better estimate of the true value of the *approximation*. On the other hand, the quality of this approximation depends on the

original sample size (92, in our example) and there is nothing we can do to change it.

An alternative way to generate a bootstrap sample in this example is by generating a new value of each response variable (y) by adding the predicted value from the original lqs model to a randomly selected residual from the original set of residuals. Thus, we resample not the entire bivariate structure but merely the residuals. As an exercise, you might try implementing this approach in R. Note that this approach is not a good idea if you have reason to believe that the distribution of residuals is not the same for all points. For instance, if there is heteroscedasticity or if the residuals contain structure not explained by the model, this residual resampling approach is not warranted.

Using the [boot](#) package in R

There is a [boot](#) package in R that contains many functions relevant to bootstrapping. As a quick example, we will show here how to obtain the same kind of bootstrap example obtained above (for the lqs model of pmDE regressed on DE for the Hyades stars.)

```
library(boot)
mystat <- function(a,b)
  lqs(a[b,2]-a[b,1])$coef
model2B.2 <- boot(cbind(x,y),
  mystat, 200)
names(model2B.2)
```

As explained in the help file, the [boot](#) function requires as input a function that accepts as arguments the whole dataset and an index that references an observation from that dataset. This is why we defined the mystat function above. To see the output that is similar to that obtained earlier for the m2B object, look in m2B2\$t:

```
cov(model2B.2$t)
sqrt(diag(cov(model2B.2$t)))
```

Compare with the output provided by [print.boot](#) and the plot produced by [plot.boot](#):

```
model2B.2
plot(model2B.2)
```

Another related function, for producing bootstrap confidence intervals, is [boot.ci](#).

Parametric bootstrapping of regression standard errors

We now return to the regression problem studied earlier.

Sometimes, resampling is done from a theoretical distribution rather than from the original sample. For instance, if simple linear regression is applied to the regression of pmDE on DE, we obtain a parametric estimate of the distribution of the residuals, namely, normal with mean zero and standard deviation estimated from the regression:

```
summary(model1)
```

Remember that model1 was defined above as lm(y~x). We observe a residual standard error of 4.449.

A parametric bootstrap scheme proceeds by simulating a new set of pmDE (or y) values using the model

```
y <- 21.9 - 3.007*x + 4.449*rnorm(92)
```

Then, we refit a linear model using y as the new response, obtaining slightly different values of the

regression coefficients. If this is repeated, we obtain an approximation of the joint distribution of the regression coefficients for this model.

Naturally, the same approach could be tried with other regression methods such as those discussed in the [EDA and regression](#) tutorial, but careful thought should be given to the parametric model used to generate the new residuals. In the normal case discussed here, the parametric bootstrap is simple, but it is really not necessary because standard linear regression already gives a very good approximation to the joint distribution of the regression coefficients when errors are heteroscedastic and normal. One possible use of this method is in a model that assumes the absolute residuals are exponentially distributed, in which case least absolute deviation regression as discussed in the [EDA and regression](#) tutorial can be justified. The reader is encouraged to implement a parametric bootstrap using the `rq` function found in the "quantreg" package.

```
?rq
```

Kolmogorov-Smirnov: Bootstrapped p-values

Earlier, we generated 5000 samples from the null distribution of the t-statistic for testing the null hypothesis that the distribution of MWG luminosities is the same as the distribution of M31 luminosities, shifted by 24.44. Let's do a similar thing for the color comparison between Hyades and non-Hyades:

```
tlist2 <- NULL
all <- c(H,nH)
for(i in 1:5000) {
  s <- sample(2586,92) # choose a sample
  tlist2 <- c(tlist2, t.test(all[s],all[-s],
    var.eq=T)$stat) # add t-stat to list
}
```

Let's look at two different ways of assessing whether the values in the `tlist2` vector appear to be a random sample from a standard normal distribution. The first is graphical, and the second uses the Kolmogorov-Smirnov test.

```
plot(qnorm((2*(1:5000)-1)/10000), sort(tlist2))
abline(0,1,col=2)
ks.test(tlist2, "pnorm")
```

First note that after implementing `ks.test` a warning message was likely generated. This warning message appears because we are using the Kolmogorov-Smirnov to test if `tlist2` is coming from a continuous distribution (i.e., the normal distribution) and due to rounding error, some values of `tlist2` may appear to be tied. However, the probability of obtaining the same value twice in a continuous distribution is 0. For our example, it does not present any difficulties since the p-value is sufficiently approximated. Also note that the graphical method does not show any major deviation from standard normality, but this graphical "test" is better able to detect departures from an overall normal shape than to detect, say, a shift of the mean away from zero. The following procedures illustrate this fact:

```
plot(qnorm((2*(1:5000)-1)/10000), sort(tlist2)-mean(tlist2))
abline(0,1,col=2)
ks.test(tlist2-mean(tlist2), "pnorm")
```

The graphical plot shows no discernable difference from before, but we see a vast difference in the new p-value returned by the Kolmogorov-Smirnov test (!!).

However, let us now consider the p-value returned by the last use of `ks.test` above. It is not quite valid because the theoretical null distribution against which we are testing depends upon an estimate (the mean) derived from the data. To get a more accurate p-value, we may use a bootstrap approach.

First, obtain the Kolmogorov-Smirnov test statistic from the test above:

```
obs.ksstat <- ks.test(tlist2-mean(tlist2),
  "pnorm")$stat
```

Now we'll generate a new bunch of these statistics under the null hypothesis that tlist2 really represents a random sample from *some* normal distribution with variance 1 and unknown mean:

```
random.ksstat <- NULL
for(i in 1:1000) {
  x <- rnorm(5000)
  random.ksstat <- c(random.ksstat,
    ks.test(x,pnorm,mean=mean(x))$stat)
}
```

Now let's look at a histogram of the test statistics and estimate a p-value:

```
hist(random.ksstat,nclass=40)
abline(v=obs.ksstat,lty=2,col=2)
mean(random.ksstat>=obs.ksstat)
```

Note that the approximate p-value above is smaller than the original p-value reported by newkstest, though it is still not small enough to provide strong evidence that the tlist2 sample is not normal with unit variance.

The bootstrap procedure above relied on multiple resamples with replacement. Since these samples were drawn from a theoretical population (in this case, a normal distribution with parameters that might be determined by the data), it is considered a *parametric* bootstrap procedure.

In a *nonparametric* bootstrap procedure, the resamples are taken from the empirical distribution of the data (that is, from a distribution that places mass $1/n$ on each of the n observed values). It is possible to implement a nonparametric bootstrap procedure to calculate a p-value for the Kolmogorov-Smirnov test here, but to do so is a bit tricky. The "straightforward" method for obtaining the null distribution of K-S statistics would be repeated usage of the following:

```
s <- sample(5000,replace=T)
ks.test(tlist[s],pnorm,mean=mean(tlist[s]))$stat
```

However, this statistic has a bias that must be corrected. To make this correction, it is necessary to rewrite the [ks.test](#) function itself. There is no way to use the output of the [ks.test](#) function to produce the bias-corrected statistics.

SUMMER SCHOOL IN STATISTICS FOR ASTRONOMERS & PHYSICISTS-2

Statistical Inference for Astronomers

MODEL SELECTION AND EVALUATION

Goodness of fit and likelihood ratio tests

C.R.Rao
Statistics Department
Pennsylvania State University

JUNE 2008

FISHERIAN FRAMEWORK

(1922) On the mathematical foundations of theoretical statistics. *Philos. Trans. Roy. Soc. A222*, 309-368.

THREE METHODOLOGICAL PROBLEMS OF STATISTICS

- Specification: selection of a model (family of probability distributions) applicable to observed data

How to choose a model?

Pearson-Fisher controversy, (χ^2, p) of Pearson

H.F. Inman. *Am. Statistician*, 48, 2-11 (1994)

- Estimation: concepts of consistency, efficiency, asymptotic variance, sufficiency and information. (Maximum likelihood)

- Testing of null hypothesis: choice of a test statistic?

Interpretation of p-values. Purpose of a test.

GOODNESS OF FIT TESTS

Karl Pearson Chi-Square (1900)

The Holy Trinity

Neyman Pearson likelihood ratio test (1928)

Wald test (1948)

Rao's Score test (1948)

Other tests

Kolmogorov- Smirnoff test

Cramer- von Mises test

A general theorem

References:

E. Lehmann(1998). *Elements of Large Sample Theory*
Springer, pages 526-535.

C.R.Rao (1973). *Linear Statistical Inference and its
Applications*, John Wiley, Chapter 6.

INFORMATION THEORETIC CRITERIA FOR MODEL SELECTION

AIC, AICc, GIC, QAIC, QAICc
TIC, WIC, BIC

OTHERS

NIC, Mallow's Cp

CROSS VALIDATION FOR MODEL ACCURACY

Training Sample
Test Sample
Bootstrap Sample

References:

C.R.Rao and Y.Wu (2000). On model selection, In *Model Selection*, Ed. Lahiri, IMS Lecture Notes, 38, 1-64.

K.P.Burnham and D.R. Anderson (1998). *Model Selection and Inference, A Practical Information Theoretic Approach*, Springer

PRELIMINARIES

- Probability model for a random variable X

Probability density at $x = \bar{x}$: $f(x, \theta)$, x continuous

Probability of event $x = \bar{x}$: $f(x, \theta)$, x discrete

- Sample of size n

$$\underset{\sim}{S} : \bar{x} = (x_1, x_2, \dots, x_n)$$

n independent observations drawn from $f(x, \theta)$.

- Likelihood and log likelihood of $\theta = (\theta_1, \dots, \theta_q)$

$$L(\theta|S) = f(x_1, \theta) \dots f(x_n, \theta)$$

$$l(\theta|S) = \log L(\theta|x) = \log f(x_1, \theta) + \dots + \log f(x_n, \theta)$$

- Score function is a q -vector

$$s = (s_1, \dots, s_q)$$

$$s_i = \frac{\partial l}{\partial \theta_i} = \sum_{j=1}^n \frac{1}{f(x_j, \theta)} \frac{\partial f(x_j, \theta)}{\partial \theta_i}, i = 1, \dots, q$$

- Maximum likelihood estimate

$$\hat{\theta} = \arg \max_{\theta} l(\theta|S)$$

usually obtained as some root of the equations (why?)

$$s_i = 0, \quad i = 1, \dots, n.$$

Information matrix

$$I(\theta) = \mathbb{E} \left(\frac{\partial^2}{\partial \theta^2} \right)$$

$q \times q$
matrix $q \times q$

$$\frac{\partial^2}{\partial \theta^2} = E(s_i s_j) \quad s_i = i\text{-th score function}$$

Important result

$$\hat{\theta} \sim (\text{asymptotically}) \sim N_q(\theta, I^{-1}) \text{ as } n \rightarrow \infty$$

under some conditions.

Karl Pearson Chi-Square Test

The dawn of statistical inference

In an article entitled, *Trial by Number*, Hacking (1984) says that the goodness-of-fit chi-square test introduced by Karl Pearson (1900), "ushered in a new kind of decision making" and gives it a place among the top 20 discoveries since 1900 considering all branches of science and technology. R.A. Fisher, who was involved in bitter controversies with Pearson, was appreciative of the chi-square test. In his book on *Statistical Methods for Research Workers* (1958, 13th edition, p.22), Fisher says, "This (chi-square), I believe is the great contribution to statistical methodology which the unsurpassed energy of Professor Pearson's work will be remembered," and devoted one full chapter on numerous ingenious applications of the chi-square test.

Pearson's chi-square is ideally applicable to qualitative data with a finite number, say s , of natural categories and the data are in the form of frequencies of individuals in different categories. The specified hypothesis is of the form

$$\pi_i = \pi_i(\theta), i = 1, \dots, s$$

where the probability π_i in category i is a given function of a k -vector parameter θ .

Class Interval (bins)	Observed frequency (O)	Expected Frequency (E)
$a_1 - a_2$	O_1	$n \pi_1(\hat{\theta})$
$a_2 - a_3$	O_2	$n \pi_2(\hat{\theta})$
\vdots	\vdots	\vdots
$a_s - a_{s+1}$	O_s	$n \pi_s(\hat{\theta})$
Total	n	n

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}, \text{ d.f.} = s-1-k$$

FITTING A BINOMIAL DISTRIBUTION

$$P(n) = \binom{k}{n} \pi^n (1-\pi)^{k-n}, k \text{ trials, } n \text{ successes}$$

$n = 0, \dots, k, k+1 \text{ classes}$

FREQUENCY DISTRIBUTION OF NUMBER OF BOYS IN FAMILIES OF SIZE EIGHT

FIT OF BINOMIAL DISTRIBUTION

Number of Boys.	Number of Families Observed.	Expected.	Excess (x).	$\frac{x^2}{m}$
0	215	165.22	+ 49.78	14.998
1	1485	1401.69	+ 83.31	4.952
2	5331	5202.65	+ 128.35	3.166
3	10649	11034.65	- 385.65	13.478
4	14959	14627.60	+ 331.40	7.508
5	11929	12409.87	- 480.87	18.633
6	6678	6580.24	+ 97.76	1.452
7	2092	1993.78	+ 98.22	4.839
8	342	264.30	+ 77.70	22.843
	53680	53680.00		91.869

Estimate of $\pi = .61$

$$\chi^2 = \sum_0^8 \frac{(O_i - E_i)^2}{E_i} = 91.869$$

d.f. = no. of classes (bins) - 1 - no. of parameters estimated

$$= 9 - 1 - 1 = 7$$

$$P < .005$$

FITTING A POISSON DISTRIBUTION

Prob for n events

$$e^{-\mu} \frac{\mu^n}{n!}, n=0, 1, \dots$$

FREQUENCY DISTRIBUTION OF DEATHS DUE TO HORSE KICKS

(RECORD OF 10 ARMY CORPS OVER 20 YEARS)

FIT OF POISSON DISTRIBUTION

Deaths.	Frequency observed.	Expected.	χ^2
0	109	108.67	.0001
1	65	66.29	.6251
2	22	20.22	.1566
3	3	4.11	
4	1	.63	
508	
601	
	200	200	<u>0.3244</u>

$d.f = 2$

$P > .85$

Estimate of $\mu = 0.61$

LIKELIHOOD RATIO TEST (LRT)

- Probability model

$$f(x, \theta), x \in R^p, \theta \in \Theta \subset R^q$$

- Sample

$$S = (x_1, \dots, x_n), n \text{ i.i.d.'s}$$

- Log likelihood

$$l(\theta|S) = \sum_1^n \log f(x_i, \theta).$$

TEST OF A SIMPLE HYPOTHESIS

$$H_{0s} : \theta = \theta_0 \text{ (specified)}$$

against the alternative

$$H_1 : \theta \neq \theta_0 \text{ (\theta unspecified)}$$

LRT for H_0

$$2 \left[l(\hat{\theta}|S) - l(\theta_0|S) \right] \sim \chi^2(q)$$

where $\hat{\theta}$ is the ML of θ and q is the dimension of the parameter θ .

TEST OF A COMPOSITE HYPOTHESIS

$H_{0c} : \theta$ belongs to a subset $\Theta_0 \subset \Theta$, especially defined by a set of k independent restrictions $g_1(\theta) = 0, \dots, g_r(\theta) = 0$

$$H_1 : \theta \in \Theta - \Theta_0.$$

LRT for H_{0c}

$$2 \left[l(\hat{\theta}|S) - l(\hat{\theta}_r|S) \right] \sim \chi^2(r)$$

where

$$\hat{\theta} = \arg \max_{\theta \in \Theta} [l(\theta|S)], \text{ the full MLE}$$

$$\hat{\theta}_r = \arg \max_{g_1(\theta)=\dots=g_k(\theta)=0} [l(\theta|S)]$$

$r =$ the number of independent restrictions.

Note: the null hypothesis is a subset of Θ , and is referred to as a nested set.

WALD TEST

Test of a Simple hypothesis

$$H_{0s} : \theta_0 \in \Theta \subset R^q$$

against the alternative

$$H_1 : \theta \neq \theta_0, \theta \in \Theta.$$

Let $\hat{\theta}$ is the MLE of θ . The Wald test for H_0 is

$$W_s = n (\hat{\theta} - \theta_0)' I(\hat{\theta}) (\hat{\theta} - \theta_0) \sim \chi^2(q)$$

where $I(\theta)$ is the information matrix. W_s is a quadratic form. For example if $q = 2$,

$$\hat{\theta} = (5, 3), \theta_0 = (1, 2), \text{ specified}$$

$$I(\hat{\theta}) = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$$

$$W_s = (5, 3) \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} 5 \\ 3 \end{pmatrix} = (13, 14) \begin{pmatrix} 5 \\ 3 \end{pmatrix} = 107(2df)$$

Test of a composite hypothesis

$$H_0 : g_1(\theta) = \dots = g_r(\theta) = 0.$$

Let

$$g(\theta) = (g_1(\theta), \dots, g_r(\theta))'$$

$$M(\theta) = \begin{pmatrix} \frac{\partial g_1}{\partial \theta_1}, & \dots & \frac{\partial g_1}{\partial \theta_q} \\ \vdots & \dots & \vdots \\ \frac{\partial g_r}{\partial \theta_1}, & \dots & \frac{\partial g_r}{\partial \theta_q} \end{pmatrix}$$

$$W_c = ng(\hat{\theta})' \left[M(\hat{\theta}) I(\hat{\theta})^{-1} M(\hat{\theta})' \right]^{-1} g(\hat{\theta}) \sim \chi^2(r)$$

$$(r \times q)(q \times q)(q \times r).$$

The test is not invariant to transformation of H_0 .

RAO'S SCORE TEST

Test of a simple hypothesis

$$H_0 : \theta = \theta_0$$

against the alternative

$$H_1 : \theta \neq \theta_0.$$

Recall the Score function which is q -vector

$$s(\theta) = \left(\frac{\partial l}{\partial \theta_1}, \dots, \frac{\partial l}{\partial \theta_q} \right)'$$

The score test for H_0 is

$$R_s = [s(\theta_0)]' [I(\theta_0)]^{-1} [s(\theta_0)] \sim \chi^2(q).$$

Note: The test does not involve the computation of $\hat{\theta}$, the MLE of θ unlike in likelihood ratio and Wald tests.

The score test for the composite hypothesis

$$\begin{aligned} H_0 : g_1(\theta) &= \dots = g_r(\theta) = 0 \\ [r(\tilde{\theta})]' [I(\tilde{\theta})]^{-1} [r(\tilde{\theta})] &\sim \chi^2(r) \end{aligned}$$

where $\tilde{\theta}$ is the MLE under the restriction of H_0 i.e.

$$\tilde{\theta} = \arg \max_{g_1(\theta)=\dots=g_r(\theta)=0} l(\theta|S).$$

Note: All the three tests of Holy Trinity are asymptotically equivalent.

Karl Pearson's chisquare tests is a special case of Rao's score test.

Reference may be made to Rao (1973) and Lehmann (1998) for some applications and comments on these tests.

All these tests are not applicable when MLE's do not exist and are not well behaved. Further, they may not be applicable when θ under null hypothesis is on the border of the admissible set Θ . Consider for instance the null hypothesis

$$H_0 : f(x|\theta) = N(x|\mu_1, \sigma_1^2)$$

and the alternative

$$H_1 : f(x|\theta) = \alpha_1 N(x|\mu_1, \sigma_1^2) + \alpha_2 N(x|\mu_2, \sigma_2^2)$$

$$\alpha_1 \geq 0, \alpha_2 \geq 0, \alpha_1 + \alpha_2 = 1.$$

All the above tests are not applicable in this case.

KOLMOGOROV-SMIRNOV AND CRAMÈR-VON MISES TESTS

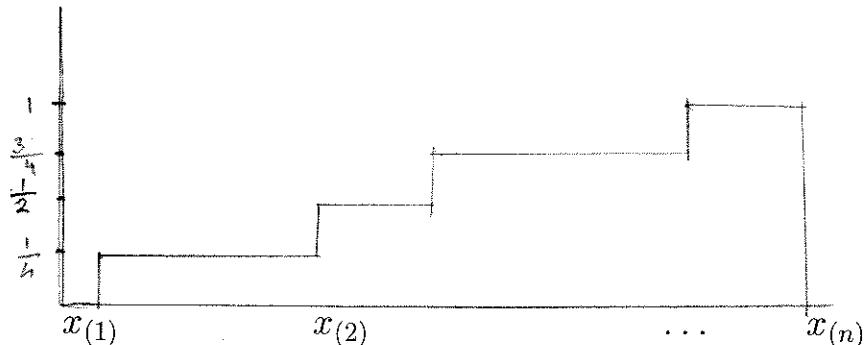
Tests of simple hypothesis

$H_{os} : F(x) \text{ is specified as } F_0(x)$

$H_1 : F(x) \text{ is arbitrary}$

where $F(x)$ is distribution function of the random variable X .

Based on the sample $S = (x_1, \dots, x_n)$ the *ML* estimate of $F(x)$ is the empirical distribution function, a step function, $\hat{F}_n(x)$ as shown in the graph. Each step is of magnitude $(1/n)$.



Kolmogorov-Smirnov test for H_0

$$KS_s = \sup_x |\hat{F}_n(x) - F_0(x)|.$$

Cramèr-von Mises test for H_0

$$CM_s = \int \left(\hat{F}_n(x) - F_0(x) \right)^2 dF_0(x).$$

The percentile points in each case can be obtained from bootstrap distribution.

TEST OF COMPOSITE HYPOTHESIS

$$H_{oc} : F(x) \in \{F(x, \theta), \theta \in \Theta\}$$

H_1 : $F(x)$ is arbitrary.

Let $\hat{\theta}$ be the MLE of θ based on the sample S . Then KS test statistic for H_{oc} is

$$KS_c = \sup_x \left| \hat{F}_n(x) - F(x, \hat{\theta}) \right| dF(x, \hat{\theta})$$

and CM test statistic for H_{oc} is

$$CM_c = \int \left(\hat{F}_n(x) - F(x, \hat{\theta}) \right)^2 dF(x, \hat{\theta})$$

The percentile points can be estimated by bootstrap sampling from $F(x, \hat{\theta})$.

References:

1. Shao, J. and Tu, D. (1995). *The Jackknife and Bootstrap*, Springer.
2. Babu, G. and Rao, C.R. (1984) Goodness-of-fit tests when parameters are estimated. *Sankhya 66*, 63-74

BOOTSTRAP METHODOLOGY

F is a distribution function DF .

Sample: $\tilde{x} = (x_1, \dots, x_n)$, i.i.d. random variables.

Sample statistics: $t(\tilde{x})$, estimator or test criterion.

Distribution of $t(\tilde{x})$.

Draw independent samples of size n , if F is known

$$\tilde{x}_1, \tilde{x}_2, \dots$$

and compute

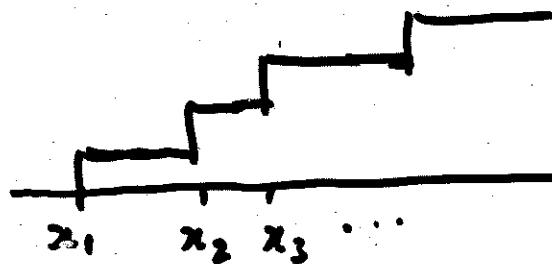
$$t(\tilde{x}_1), t(\tilde{x}_2), \dots, t(\tilde{x}_N), \dots$$

The sequence characterizes the sampling distribution F_t of $t(\tilde{x})$. By choosing N sufficiently large, we can estimate the quantiles of the distribution of $T(\tilde{x})$ with any desired accuracy.

If F is not known, we may proceed as follows

Estimate F using the sample $\tilde{x} = (x_1, \dots, x_n)$

$$\hat{F} : \begin{cases} \text{Value} & x_1, \dots, x_n \\ \text{Probability} & \frac{1}{n}, \dots, \frac{1}{n} \end{cases}$$



Bootstrap distribution of $t(\tilde{x})$

Draw samples $\tilde{x}_1^*, \tilde{x}_2^*, \dots, \tilde{x}_N^*$, ... from $\hat{F}(x)$ and compute

$$t(\tilde{x}_1^*), t(\tilde{x}_2^*), \dots, t(\tilde{x}_N^*), \dots$$

The distribution F_t^* defined by the sequence is called *the bootstrap distribution of $t(\tilde{x})$* .

Reference: Shao, J. and Tu, D. (1995). *The Jackknife and Bootstrap*. Springer

A GENERAL THEOREM

Let y_1, \dots, y_n be random variables such that

$$y_i \sim N(g(x_i, \theta), \sigma_i^2) \\ i = 2, \dots, n$$

where x_i are fixed covariates, θ is unknown p -vector parameter and σ_i^2 are known. The maximum likelihood estimate of θ is

$$\hat{\theta} = \arg \min_{\theta} \sum_{i=1}^n \left(\frac{y_i - g(x_i, \theta)}{\sigma_i} \right)^2.$$

The χ^2 goodness-of-fit is

$$\sum_{i=1}^n \left(\frac{y_i - g(x_i, \hat{\theta})}{\sigma_i} \right)^2, \text{ d.f. } n - p$$

where p is the number of unknown parameters.

Table 2. XRT 2-10 keV flux

T(mid) ^a (s)	T(exp) (s)	Flux ^b	T(mid) ^a (s)	T(exp) (s)	Flux ^b
133	5.	122.7 ± 5.7	578	10.	29.8 ± 2.0
143	5.	109.5 ± 5.4	598	10.	28.5 ± 2.0
153	5.	101.4 ± 5.2	618	10.	29.1 ± 2.0
163	5.	92.0 ± 4.9	638	10.	24.8 ± 1.8
173	5.	86.8 ± 4.8	658	10.	27.3 ± 1.9
183	5.	83.7 ± 4.7	678	10.	24.6 ± 1.8
193	5.	77.2 ± 4.5	708	20.	24.2 ± 1.3
203	5.	69.4 ± 4.3	748	20.	20.4 ± 1.2
213	5.	69.2 ± 4.3	788	20.	19.8 ± 1.2
223	5.	62.4 ± 4.1	828	20.	19.0 ± 1.1
233	5.	65.0 ± 4.1	868	20.	16.3 ± 1.1
243	5.	57.2 ± 3.9	908	20.	18.5 ± 1.1
253	5.	54.6 ± 3.8	948	20.	17.6 ± 1.1
263	5.	54.3 ± 3.8	988	20.	16.3 ± 1.1
278	10.	50.8 ± 2.6	1028	20.	15.5 ± 1.4
298	10.	49.8 ± 2.6	6009	150.	1.072 ± 0.134
318	10.	45.4 ± 2.5	6309	150.	1.331 ± 0.153
338	10.	45.5 ± 2.5	6609	150.	0.987 ± 0.126
358	10.	46.9 ± 2.5	6909	150.	1.010 ± 0.152
378	10.	40.2 ± 2.3	11809	350.	0.560 ± 0.056
398	10.	41.7 ± 2.4	12509	350.	0.482 ± 0.052
418	10.	39.4 ± 2.3	18109	350.	0.331 ± 0.052
438	10.	39.9 ± 2.3	23859	2000.	0.173 ± 0.039
458	10.	34.8 ± 2.2	27859	2000.	0.108 ± 0.028
478	10.	31.9 ± 2.1	35859	2000.	0.079 ± 0.021
498	10.	31.5 ± 2.1	81459	5400.	0.0361 ± 0.0088
518	10.	32.5 ± 2.1	99102	11850.	0.0273 ± 0.0078
538	10.	27.4 ± 1.9	165485	22000.	0.0080 ± 0.0016
558	10.	32.4 ± 2.1	412515	42000.	0.0013 ± 0.0006

^atime since trigger^bflux in units of 10^{-11} erg cm $^{-2}$ s $^{-1}$ 2-10 keV

be added to the statistical error. The corrections and the systematic error are posted.¹ The count spectra are binned between 16 and 148.8 keV in ~ 2 keV bins. Table 1 summarizes the spectral fits to the entire burst (12.80 s) and to the peak 1 s, with 90% confidence limits. We fit the count spectra with three nested models, here presented as energy spectra²: a power law $F(E) \propto E^\beta$; a power law with an exponential cutoff $F(E) \propto E^\beta \exp[-E/E_0]$; and the 'Band' model (Band et al. 1993), a low energy power law with an exponential cutoff that transitions into a high energy power law $F(E) \propto E^{\beta_2}$. The peak energy $E_p = (1 + \beta)E_0$ is both physically more relevant and less correlated with β than E_0 ; E_p is the energy of the peak of $E F(E) \propto \nu f_\nu$. The power law with an exponential cutoff is the same as the Band model with $\beta_2 = -\infty$, and the power law model is the same as the other two models with $E_0 = \infty$.

Table 1. BAT Spectral Fits

Parameter	Entire Burst			Peak Flux		
	Power Law ^a	Power Law, Cutoff ^b	Band Model ^c	Power Law ^a	Power Law, Cutoff ^b	Band Model ^c
β^d	-0.78 ^e	$0.01^{+0.11}_{-0.12}$	$0.01^{+0.11}_{-0.12}$	-0.42 ^e	$0.45^{+0.14}_{-0.14}$	$-0.45^{+0.14}_{-0.14}$
β_2^f	—	—	$-7.84^{+6.40}_{-1.16}$	—	—	$-8.27^{+7.14}_{-0.73}$
E_p^g	—	$78.8^{+3.9}_{-3.1}$	$78.8^{+3.7}_{-3.1}$	—	$102.4^{+7.1}_{-6.3}$	$102.4^{+8.1}_{-6.3}$
Norm	$7.15^{e,h}$	$14.2^{+5.9}_{-4.2}^i$	$15.0^{+1.65}_{-1.45} h$	$19.35^{e,h}$	$7.62^{+3.91}_{-2.65} i$	$44.00^{+6.05}_{-5.20} h$
χ^2/dof	181.7/57	15.2/56	15.2/55	169.8/57	31.6/56	31.6/55

P <.005 >.99 ≥ .99 <.005 ≥ .95 >.95

^aPower law model, $F(E) \propto E^\beta$.

^bPower law with an exponential cutoff, $F(E) \propto E^\beta \exp[-E/E_0]$.

^cBand model (Band et al. 1993), a low energy power law with an exponential cutoff transitioning to a high energy power law $F(E) \propto E^{\beta_2}$.

^dThe low energy spectral index.

^eFit too poor to produce uncertainty range.

^fThe high energy spectral index. The fit is insensitive to $\beta_2 < -2.5$ for the fitted E_p .

^gThe energy of the peak of $EN(E) \propto \nu f_\nu$, and $E_p = (2 + \beta)E_0$.

^hThe normalization of the spectrum at 50 keV, in $\text{keV cm}^{-2} \text{ s}^{-1} \text{ keV}^{-1}$.

ⁱThe normalization of the spectrum at 1 keV, in $\text{keV cm}^{-2} \text{ s}^{-1} \text{ keV}^{-1}$.

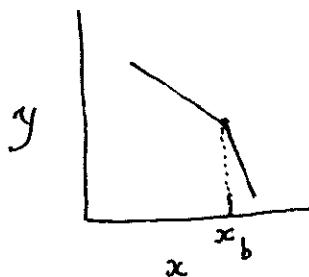
Number of free parameters

- A single polynomial of degree k

$$\alpha_0 + \alpha_1 x + \dots + \alpha_k x^k$$

$$\text{No.} = k+1$$

- Two lines and a break



$$\begin{matrix} \alpha_1 + \beta_1 x \\ 1 \\ 2 \end{matrix}, \quad \begin{matrix} \alpha_2 + \beta_2 x \\ 3 \\ 4 \end{matrix}$$

$$\alpha_1 + \beta_1 x_b = \alpha_2 + \beta_2 x_b \quad (\text{restriction})$$

$$5$$

$$5 - 1 = 4$$

- Quadratic and line with a break

$$\alpha_1 + \beta_1 x + \gamma_1 x^2, \quad \alpha_2 + \beta_2 x + \gamma_2 x^2, \quad x_b$$

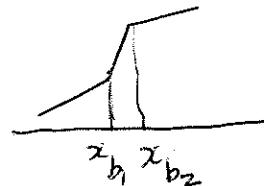
$$6 - 1 = 5$$

- Three lines and 2 breaks

$$\alpha_1 + \beta_1 x, \quad \alpha_2 + \beta_2 x, \quad \alpha_3 + \beta_3 x, \quad x_{b1}, x_{b2} \quad (\text{breaks or knots})$$

$$\alpha_1 + \beta_1 x_{b1} = \alpha_2 + \beta_2 x_{b1}$$

$$\alpha_2 + \beta_2 x_{b2} = \alpha_3 + \beta_3 x_{b2}$$

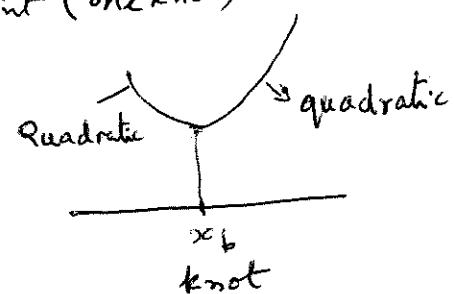


$$8 - 2(\text{restrictions}) = 6 \text{ free parameters}$$

- Two quadratics with a differentiable function at the break point (one knot)

$$\alpha_1 + \beta_1 x + \gamma_1 x^2 \quad x < x_b$$

$$\alpha_2 + \beta_2 x + \gamma_2 x^2 \quad x > x_b$$



$$\alpha_1 + \beta_1 x_{b*} + \gamma_1 x_{b*}^2 = \alpha_2 + \beta_2 x_b + \gamma_2 x_b^2$$

$$\beta_1 + 2\gamma_1 x_b = \beta_2 + 2\gamma_2 x_b$$

$$6 - 2 = 4 \text{ parameters}$$

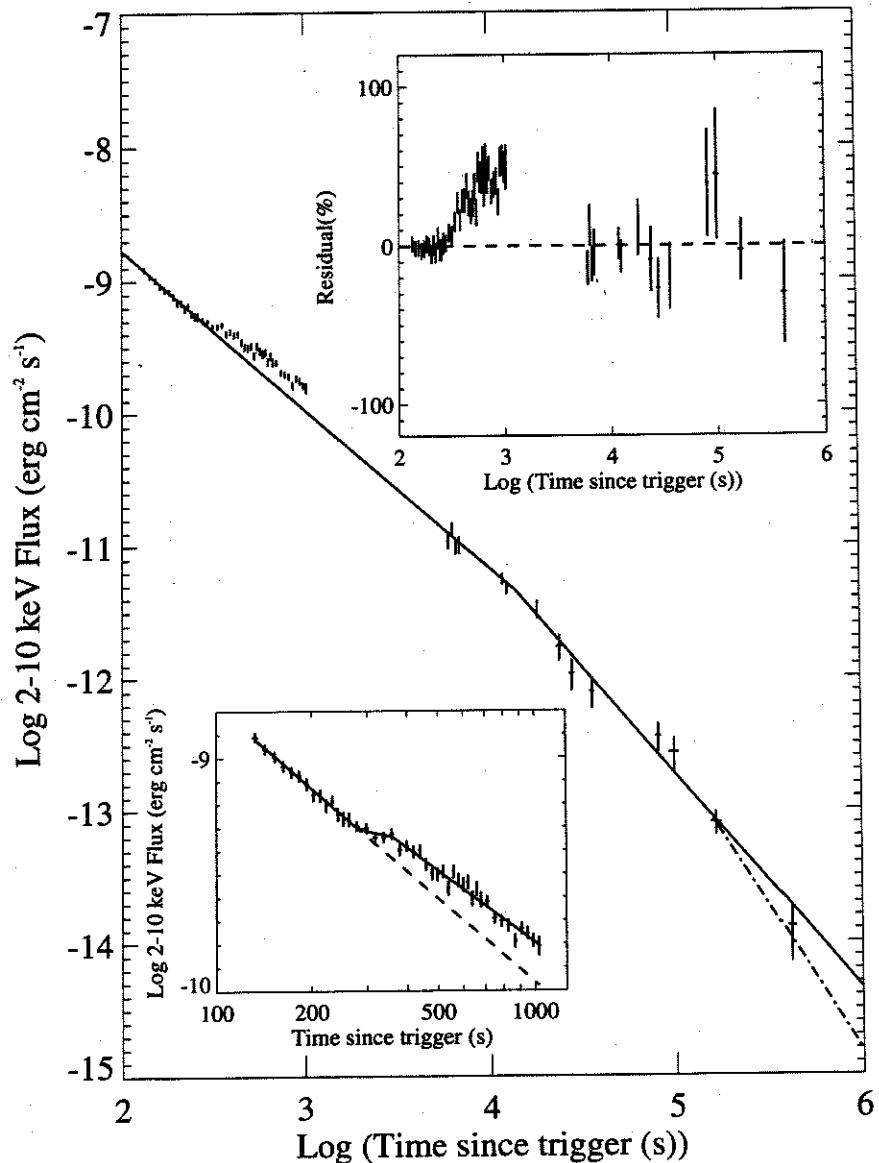


Fig. 3.— XRT decay light curve of GRB050525a including both Photodiode Mode ($T < 2000$ s) and Photon Counting Mode ($T > 2000$ s) data. The solid line is a broken power law fit to the combined data excluding those Photodiode Mode points colored green (see text). The dash-dot line is shown for illustration and has a slope of $\alpha = -2.2$, which is the value expected from simple modelling of a jet break (see text). The lower inset shows the data taken in photodiode mode only, during the first ~ 1000 s after the BAT trigger. The solid line is a fit to the data with a power law model that includes two temporal breaks to different decay rates. The dashed line is an extrapolation of a simple power law fit (single slope) to the first segment of data prior to about 300 s. The upper inset shows the residuals with respect to the two power-law model fit to all the data, expressed as a percentage of the predicted model flux.

Table 3. UVOT multicolor data

T(mid) ^a (s)	T(exp) (s)	Mag	Flux Density ^b	T(mid) ^a (s)	T(exp) (s)	Mag	Flux Density ^b
V filter							
66	1.	13.21 ± 0.24	189.8 ± 41.2	148	5.	13.84 ± 0.12	106.2 ± 11.5
67	1.	12.90 ± 0.23	254.0 ± 53.6	153	5.	13.87 ± 0.12	103.2 ± 11.3
68	1.	12.86 ± 0.23	263.3 ± 55.5	158	5.	14.06 ± 0.12	87.1 ± 10.0
69	1.	13.01 ± 0.23	227.9 ± 48.5	163	5.	14.00 ± 0.12	91.7 ± 10.3
70	1.	12.97 ± 0.23	236.3 ± 50.1	168	5.	14.01 ± 0.12	90.8 ± 10.3
71	1.	13.31 ± 0.23	172.8 ± 34.0	173	10.	14.08 ± 0.13	83.3 ± 10.6
72	1.	13.13 ± 0.23	204.3 ± 43.9	258	10.	14.64 ± 0.14	49.8 ± 6.8
73	1.	13.01 ± 0.23	227.9 ± 48.5	342	10.	14.79 ± 0.15	43.3 ± 6.4
78	5.	13.13 ± 0.10	204.3 ± 19.6	426	10.	15.22 ± 0.17	29.2 ± 4.9
83	5.	13.26 ± 0.10	181.5 ± 17.7	511	10.	15.47 ± 0.19	23.2 ± 4.4
88	5.	13.18 ± 0.10	195.5 ± 18.9	595	10.	16.06 ± 0.24	13.4 ± 3.3
93	5.	13.24 ± 0.11	185.6 ± 18.1	680	10.	15.83 ± 0.22	16.6 ± 3.7
98	5.	13.25 ± 0.11	184.2 ± 17.9	764	10.	16.06 ± 0.25	13.4 ± 3.5
103	5.	13.51 ± 0.11	144.9 ± 14.7	849	10.	15.78 ± 0.22	17.4 ± 3.9
108	5.	13.44 ± 0.11	154.4 ± 15.5	933	10.	15.85 ± 0.24	16.3 ± 4.0
113	5.	13.67 ± 0.11	124.7 ± 13.0	1243	100.	16.34 ± 0.15	10.4 ± 1.5
118	5.	13.48 ± 0.11	148.4 ± 15.0	18575	156.	18.15 ± 0.41	2.0 ± 0.9
123	5.	13.62 ± 0.11	130.2 ± 13.5	22163	580.	19.10 ± 0.27	0.8 ± 0.2
128	5.	13.86 ± 0.12	104.2 ± 11.4	35638	750.	18.86 ± 0.27	1.0 ± 0.3
133	5.	13.70 ± 0.11	121.5 ± 12.8	49320	4982.	> 20.62	< 0.2
138	5.	13.83 ± 0.12	107.2 ± 11.6	971360	33800.	> 22.09	< 0.1
143	5.	13.81 ± 0.12	109.2 ± 11.8	1171176	6081.	> 21.16	< 0.1
B filter							
229	10.	14.79 ± 0.12	72.2 ± 8.4	904	10.	16.44 ± 0.20	15.8 ± 3.2
313	10.	15.19 ± 0.12	49.9 ± 5.8	1034	100.	16.61 ± 0.11	13.5 ± 1.4
397	10.	15.51 ± 0.13	37.2 ± 4.7	12671	390.	18.59 ± 0.18	2.2 ± 0.4
482	10.	15.63 ± 0.14	33.3 ± 4.6	16182	190.	18.69 ± 0.17	2.0 ± 0.3
571	10.	15.70 ± 0.14	31.2 ± 4.3	30031	388.	19.82 ± 0.52	0.7 ± 0.4
651	10.	16.13 ± 0.16	21.0 ± 3.3	33898	900.	20.84 ± 0.45	0.3 ± 0.1
735	10.	16.03 ± 0.16	23.0 ± 3.7	45468	896.	> 20.70	< 0.3
820	10.	16.56 ± 0.20	14.1 ± 2.9	62549	6513.	> 21.55	< 0.1
U filter							
215	10.	13.70 ± 0.18	110.3 ± 19.9	890	10.	15.29 ± 0.21	25.5 ± 5.4
299	10.	14.08 ± 0.18	77.8 ± 14.0	975	10.	15.32 ± 0.22	24.8 ± 5.6
419	10.	14.47 ± 0.19	54.3 ± 10.4	12019	900.	17.66 ± 0.17	2.9 ± 0.5

The background subtracted 2–10 keV light curve in the time interval $T+128\text{ s} - T+1048\text{ s}$ (PD mode) is shown in Figure 3 (inset). The X-ray afterglow of GRB 050525 is clearly fading. The early afterglow decay was first fitted with a single power-law model, resulting in a best fit decay index $\alpha = -0.95 \pm 0.03$, with $\chi^2_r = 1.17$ (42 dof). Inspection of the residuals to the best fit model suggests that a flattening of the decay curve or a re-brightening of the source occurs at ~ 300 seconds after the trigger. A better fit is provided by a broken power law model with slopes α_1, α_2 and a break at t_b . This model gave $\chi^2_r = 0.98$ (40 dof), with best fit parameters $\alpha_1 = -1.23^{+0.03}_{-0.02}$, $\alpha_2 = -0.91$ and $t_b = 203\text{ s}$.

Again, however, the residuals suggest systematic deviations from this model. We thus tried a broken power law with two temporal breaks. This model provided a very good fit to the data, with $\chi^2_r = 0.72$ (38 dof) and is plotted in Figure 3 (inset) as a solid line. The best fit parameters are $\alpha_1 = -1.19$, $t_b^1 = 282\text{ s}$, $\alpha_2 = -0.30$, $t_b^2 = 359\text{ s}$, and $\alpha_3 = -1.02$.

Next, we fitted the X-ray data taken in PC mode at times more than 5000 s after the trigger. We first used a single power-law model, obtaining a best fit decay index $\alpha = -1.51 \pm 0.07$, with $\chi^2_r = 1.40$ (12 dof). The poor fit is the result of a clear steepening of the light curve with time. We thus tried a broken power law model. The model provided a very good fit with $\chi^2_r = 0.97$ (10 dof) and best fit parameters $\alpha_1 = -1.16$, $\alpha_2 = -1.62$ and $t_b = 13177\text{ s}$.

Finally, we tried fitting the total light curve derived from the combined PD and PC mode data (see Figure 3). We find that the power law fit to the pre-brightening PD mode data ($T < 280\text{ s}$) extrapolates well to the pre-break PC mode data. Moreover the decay index before 280 s agrees well with that of the PC mode data before the 13ks break. In contrast, if we extrapolate the post brightening PD mode data to later times using the best fit slope, a significant excess is predicted compared with the measured PC mode data. To join the post brightening PD mode data to the PC mode data requires a model with at least two temporal breaks, which are not constrained because of the intervening gap in X-ray coverage. We conclude that the brightening at about 280 s in the PD mode data represents a flare in the X-ray flux, possibly similar to the sometimes much larger flares that are seen at early times in other bursts (Burrows et al 2005b; Piro et al. 2005), and that the flux returns to the pre-flare decay curve prior to the start of our PC mode data.

We thus fit the combined PD and PC mode data excluding PD data at times $t > T+288\text{ s}$ (green points in Figure 3). A broken power law model provided a good fit (solid line of Figure 3), with $\chi^2_r = 0.50$ (25 dof) and best fit parameters $\alpha_1 = -1.20 \pm 0.03$, $\alpha_2 = -1.62^{+0.11}_{-0.16}$ and $t_b = 13726^{+7469}_{-5123}\text{ s}$. The break time is thus ~ 3.8 hours.

The complete XRT data are recorded in Table 2.

INFORMATION THEORETIC CRITERIA FOR MODEL SELECTION

Problem

Let $g(x)$ be the true unknown probability distribution which gave rise to observed data, $S = (x_1, \dots, x_n)$, a sample of n independent observations. Suppose that we have a set of r candidate models

$f_i(x, \theta_i)$, $\theta_i \in \Theta_i$, is a k_i -vector parameter, $i = 1, \dots, r$.

The true $g(x)$ may or may not belong to any of the models.

The Kullback-Leibler measure of separation of the model f_i from g is

$$K(g, f_i) = \min_{\theta_i} \int \log \frac{g(x)}{f_i(x, \theta_i)} dG = \\ \int \log g(x) dG(x) - \max_{\theta_i} \int \log f_i(x, \theta_i) dG(x)$$

where G is the distribution function DF of g . The best choice of the model is s such that

$$\max_{\theta_s} \int \log f_s(x, \theta_s) dG \geq \max_{\theta_i} \int \log f_i(x, \theta_i) dG \forall i. \quad (C)$$

We call such $f_s(x, \theta_s)$ a quasi-true model. The criterion (C) cannot be used as G is unknown. However we have the empirical DF, \hat{G}_n of G , based on the observed sample. Substituting \hat{G}_n for G , we have

$$\max_{\theta_i} \int n \log f_i(x, \theta_i) d\hat{G} = \sum \log f_i(x, \hat{\theta}_i) = l_i(\hat{\theta}_i | S) \quad (L)$$

which is maximum likelihood under the model f_i

Unfortunately, the log likelihood in (L) is a biased estimate of the right hand side of (C). We estimate the bias and write the criterion as

$$C_i = -2l_i(\hat{\theta}_i|S) + 2b_i(\hat{G}_n) \quad (D)$$

and choose the model for which C_i is a minimum. There are various choices of $b_i(\hat{G}_n)$, and the criteria based on (D) are called information theoretic criteria (ITC). Some choices which have been made in various applications are as follows.

$$1. \quad AIC = -2l_i(\hat{\theta}_i|S) + 2k_i \quad (E)$$

where k_i is the number of parameters in model i , is called Akaike information criteria. The choice of $b_i(\hat{G}_n)$ as k_i is strictly valid if g the true model, is a member of the i -th family.

2. Slight improvement over (E) are given in (F)

$$AIC_c = -2l_i(\hat{\theta}_i|S) + 2k_i[n/(n - k_i - 1)] \quad (F).$$

3. Takechi information criterion

$$TIC = -2l_i(\hat{\theta}_i|S) + 2 \operatorname{tr} \left[J(\hat{\theta}_i) I(\theta_i)^{-1} \right] \quad (G)$$

where

$$I(\hat{\theta}_i) = \left(I_{rs}(\hat{\theta}_i) \right)$$

$$I_{rs}(\hat{\theta}_i) = \left| \frac{\partial^2 l(\theta_i|S)}{\partial \theta_{ir} \partial \theta_{is}} \right|_{\hat{\theta}_i}$$

$$J(\hat{\theta}_i) = \Sigma_1^r a_r a'_r$$

$$a'_r = \left(\frac{\partial}{\partial \theta_{i1}} \log f_i(x_r|\theta_i), \dots, \frac{\partial}{\partial \theta_{ik_i}} \log f_i(x_r|\theta_i) \right) |_{\hat{\theta}_i}.$$

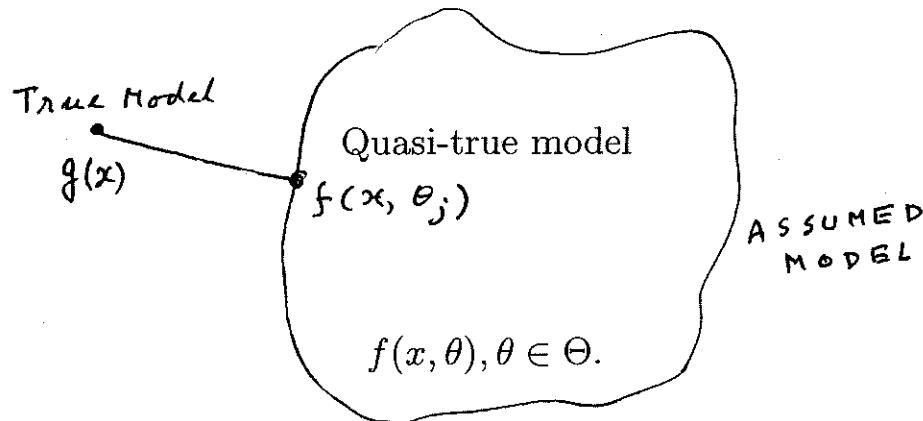
KULLBACK-LEIBLER (KL) MEASURE OF SEPARATION

KL is a measure of separation of a probability distribution $f(\theta, x)$ from a specified distribution $g(x)$.

$$KL(f, g) = \int f(x, \theta) \log \frac{f(x, \theta)}{g(x)} dx \geq 0$$

where

$-\log \frac{f(x)}{g(x)}$ is Boltzman Entropy.



$$\theta_g = \arg \min_{\theta} KL(f(x, \theta), g(x)).$$

Property of maximum likelihood

$$\hat{\theta} \rightarrow \theta_g \text{ as } n \rightarrow \infty.$$

Reference:

1. Nishi, R. (1988). Maximum likelihood principle and model selection when the true model is unspecified. *J. Multivariate Analysis*, **27**, 392-403.

CRITERIA PROVIDING CONSISTENT ESTIMATE OF QUASI-TRUE MODEL

The information criteria based on bias corrected maximum log likelihood such as AIC, TIC, etc., may not provide consistent estimate of the quasi-true model, i.e., model closest to the true model in the set of candidate models as $n \rightarrow \infty$. Alternative criteria developed to provide consistent estimates as $n \rightarrow \infty$ are of the form

$$-2l_i(\hat{\theta}_i|S) + C_n k_i$$

where C_n is chosen such that

$$\lim_{n \rightarrow \infty} \frac{C_n}{n} = 0 \text{ and } \frac{C_n}{\log \log n} = +\infty.$$

Some choices of C_n are $\log n$, $a \log(\log n)$ with $a > 2$, ~~< 0~~. Bai, Rao and Wu (1999) presented a data oriented value for C_n .

Reference:

1. Bai, Z.D., Rao, C.R. and Wu, Y. (1999). Model selection with data oriented penalty, *J. Statist. Plann. Inference*, 77, 103-117.

$$BIC \text{ (or } SBC) = -2 l_i(\hat{\theta}_i|S) + k_i \log n$$

~~GIVE~~ *SBC* : Schwarz's Bayesian criterion provides an approximate Bayes factor

MODELLING COVARIANCE STRUCTURE FOR REPEATED MEASURES DATA

Table III. Akaike's information criterion (AIC) and Schwarz's Bayesian criterion (SBC) for six covariance structures.

Structure name	AIC	SBC
1. Simple	+459.5	+461.6
2. Compound symmetric	+175.6	+179.9
3. Autoregressive (1)	+139.5	+143.8
4. Autoregressive (1) with random effect for patients	+126.5	+132.9
5. Toeplitz (banded)	+121.9	+139.2
6. Unstructured	+110.1	+187.7

where $L(\hat{\theta})$ is the maximized log-likelihood or restricted log-likelihood (REML), q is the number of parameters in the covariance matrix, p is the number of fixed effect parameters and N^* is the total number of 'observations' (N for ML and $N - p$ for REML, where N is the number of subjects).

small

Models with *large* AIC or SBC values indicate a better fit. However, it is important to note that the SBC criterion penalizes models more severely for the number of estimated parameters than does AIC. Hence the two criteria will not always agree on the choice of 'best' model. Since our objective is parsimonious modelling of the covariance structure, we will rely more on the SBC than the AIC criterion.

AIC and SBC values for the six covariance structures are shown in Table III. 'Unstructured', has the largest AIC, but AR(1)+RE, 'autoregressive with random effect for patient', has the largest SBC. Toeplitz ranks second in both AIC and SBC. The discrepancy between AIC and SBC for the UN structure reflects the penalty for the large number of parameters in the UN covariance matrix. Based on inspection of the correlation estimates in Tables I and III, the graphs of Figure 5, and the relative values of SBC, we conclude that AR(1)+RE, 'autoregressive with random effect for patient', is the best choice of covariance structure.

CROSS VALIDATION

Cross validation is generally used for variable selection in regression problems. Suppose that y is the variable to be predicted based on a set of k predictor variables, x_1, \dots, x_k . Consider n sets of observations on y, x_1, \dots, x_k .

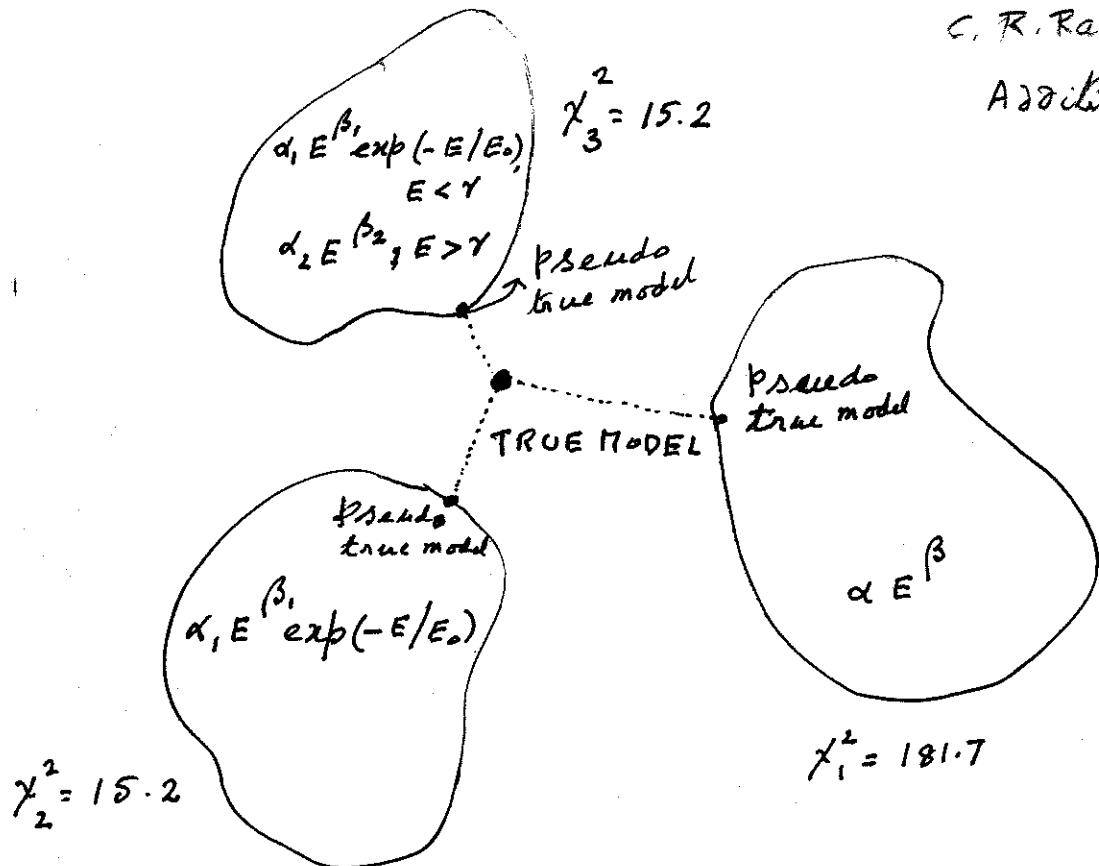
y	x_1	x_2	\dots	x_k
y_1	x_{11}	x_{21}	\dots	x_{k1}
\cdot	\cdot	\cdot	\cdot	\cdot
y_n	x_{1n}	x_{2n}	\dots	x_{kn}

Divide the observations into two random sets of sizes n_1 and n_2 (usually with $n_1 \ll n_2$). Using the n_2 observations, which we may call the training set, we fit the regression or prediction functions $f_1(x), f_2(x), \dots$ based on different subsets of variables. Use the different prediction functions to predict the y values in the n_1 observations not used, called the validating set, and for each function compute $S = \text{sum of squares of the differences between the observed and predicted values}$. Choose that function which gives the smallest value to S . The variables used to estimate the function are considered to be the best for predicting y . We can repeat the process a number of times by dividing the observations at random into training and validating sets and each time compute the sum of squares of differences for each subset. Finally the decision can be taken on the average of S values for each subset.

Ideally, one has to divide the observations into 3 sets of n_1, n_2, n_3 . Use n_2 as the training set. Use n_1 to select one or a few among alternative models. Finally compute the prediction error using n_3 observations, as an estimate of true error.

C. R. Rao

Additional notes



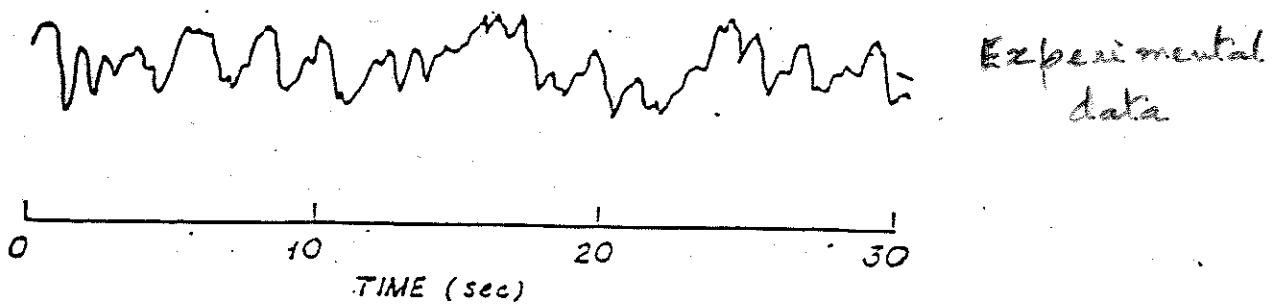
MODEL SELECTION CRITERIA (MSC)

- Choose the model
- Do cross validation for accuracy of prediction
- If MSC suggests more than one model we may have to find some sort of average model!

It would be interesting to apply these model choosing methods on the gamma ray burst data

Check χ^2_2 and χ^2_3 !!

An interesting example due to the famous mathematician Mark Kac (see his autobiography *Enigmas of Chance*, pp. 74-76.) shows how the graph of a deterministic function could mimic the tracing of a random mechanism. To test Smoluchowski's theory of Brownian motion of a little mirror suspended on a quartz fiber in a vessel containing air, Kappler conducted an ingenious experiment in 1931 to obtain photographic tracings of the motion of the mirror. One such a tracing of 30 seconds' duration is reproduced in the figure below.



Kac remarks that looking at the graph, "it is difficult to escape the feeling that one is in the presence of chance incarnate and the tracing could only have been produced by a random mechanism." Kappler's experiment might be interpreted as confirming Smoluchowski's theory that the mirror is hit at random by the molecules of air giving the graph of the displacement of the mirror the character of a stationary Gaussian process.

Kac shows that the same kind of tracing indistinguishable from Kappler's graph by any statistical analysis, can be produced by plotting the function

$$\alpha \frac{\cos \lambda_1 t + \cos \lambda_2 t + \dots + \cos \lambda_n t}{\sqrt{n}}$$

~~chaos~~
~~chance~~

for sufficiently large n , choosing a sequence of numbers $\lambda_1, \dots, \lambda_n$ and a scale factor α . Kac asks: So what is chance?

modeling real data

= mostly deterministic + stochastic
(chaos)

7

There is no need for these hypotheses to be true, or even to be at all like the truth; rather one thing is sufficient for them –that they should yield calculations which agree with the observations.

Osiander
in preface to Copernicus *De Revolutionibus*

Osiander was a Protestant Theologian. The issue became more heated in the following century in the dispute between Galileo and the catholic church. The position of the latter as stated by Cardinal Bellarmine in 1615 was that the church would raise no objections if Galileo stated his theory as a mathematical hypothesis, "invented and assumed in order to abbreviate and ease the calculations", provided he did not claim it to be a true description of the world.

Statistical Modeling**The Two Cultures**

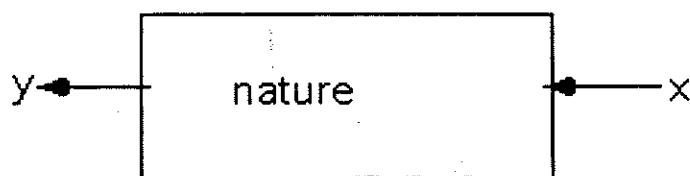
Statistics starts with data.

Think of the data as being generated by a black box .

A vector of input variables x (independent variables) go into one side.

Response variables y come out on the other side.

Inside the black box, nature functions to associate the input variables with the response variables, so the picture is like this:



The Data Modeling Culture

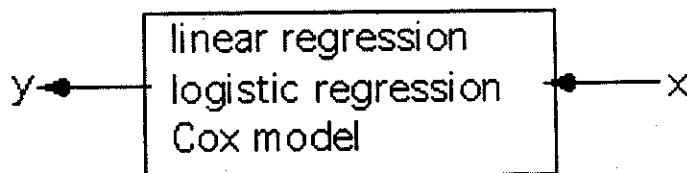
starts with assuming a stochastic data model for the inside of the black box.

A common data model is that data are generated by independent draws from:

*response variables=f(predictor variables,
random noise, parameters)*

Parameters are estimated from the data and the model then used for information and/or prediction.

The black box is filled in like this:



Model Validation: yes-no using goodness-of-fit tests and residual examination

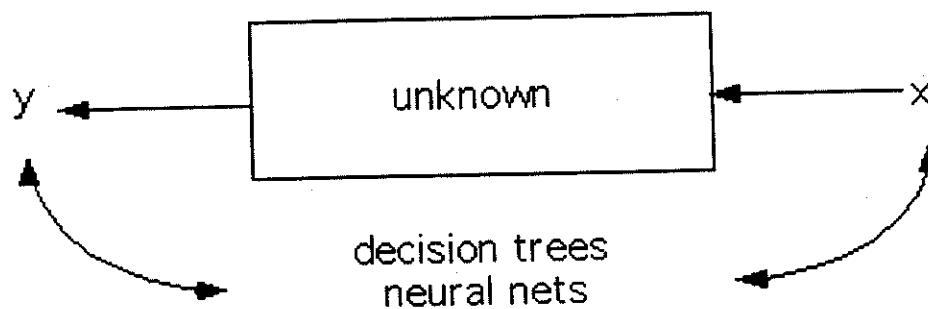
Estimated Culture Population: 98% of all statisticians

The Algorithmic Modeling Culture

The analysis in this culture considers the inside of the box complex and unknown.

The approach is to find a function $f(x)$ --an algorithm that operates on x to predict the responses y .

The black box looks like this



Model Validation: measured by predictive accuracy

Estimated Culture Population: 2% of statisticians--many in other fields

Two Major Goals In Analyzing The Data:

prediction: *to be able to predict what the responses are going to be to future input variables.*

regression:

classification:

accuracy:

Information *to extract some information about how nature is associating the response variables to the input variables.*

FRIESON (1937)
 (PROC. ROY. SOC A, VOL. 160)

Determinations of Velocity of light

year	Velocity km/sec	Fitted value	stated error
1874	299,990	299,987	± 300
1879	299,910	299,921	± 50
1882	299,850	299,867	± 30
1882	299,853	299,867	± 60
1902	299,901	299,903	± 84
1924	299,802	299,833	± 30
1926	299,796	299,804	± 4
1928	299,778	299,783	± 20
1932	299,774	299,771	± 11

$$V = 299,885 + 115 \sin \left\{ \frac{2\pi}{40} (T - 1901) \right\}$$

T in years

Ghewry de Bray (1934)
 Nature Vol 137

Tests of significance, when used accurately, are capable of rejecting or invalidating a hypotheses in so far as they are contradicted by data; but they are never capable of establishing them as certainly true.

R.A. Fisher

"Statistical tests", *Nature*, 136, 474

All hypotheses are wrong. One may be better than the other for answering a specific question.

Criterion: PERFORMANCE

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

**Spectral
Analysis**

The
Periodogram
The
Periodogram
and Regression
The
Periodogram
and the
Spectral
Density
Smoothing and
Tapering
Extensions

References

Time Series and Stochastic Processes.

John Fricks

Dept of Statistics
Penn State University
University Park, PA 16802



Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

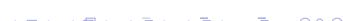
**Spectral
Analysis**

The
Periodogram
The
Periodogram
and Regression
The
Periodogram
and the
Spectral
Density
Smoothing and
Tapering
Extensions

References

Outline

- Poisson Process.
- Kalman Filter.
- Spectral analysis of stochastic processes.
 - Periodic trend.
 - Stationary processes and the spectral density.
 - Estimation of the Spectral Density.
 - Extensions for non-stationary processes.



Terminology

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain
Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering
Extensions

References

- A stochastic process is a collection of random variables $\{X_t\}$ indexed by a set T , i.e. $t \in T$. (Not necessarily independent!)
- If T consists of the integers (or a subset), the process is called a *Discrete Time Stochastic Process*.
- If T consists of the real numbers (or a subset), the process is called *Continuous Time Stochastic Process*.
- If T is \mathbb{R}^2 , then the process is called a *Random Field*.
- Similarly, these processes may take on values which are real or restricted to the integers and are called *continuous state space* or *discrete state space* accordingly.

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

Terminology

- Note that in the statistics literature, the term *Time Series* is generally restricted to discrete time, continuous state space stochastic processes.
- For the other combinations, techniques are generally characterized as *Statistical Inference for Stochastic Processes*.

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

Poisson Process

- The *Poisson process* is the canonical example of a continuous time, discrete state space stochastic process and more specifically a *counting process*.
- Counting process.
 - A counting process, $N(t)$, is a non-negative integer.
 - A counting process is an non-decreasing function of t .
 - For $s < t$, $N(t) - N(s)$ are the number of events in $(s, t]$.



Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

Poisson Process Definition

Time Series and Stochastic Processes.

John Fricks

Introduction

Time Domain Poisson Process State Space Model & Kalman Filter Filtering and the Likelihood Function

Spectral Analysis The Periodogram The Periodogram and Regression The Periodogram and the Spectral Density Smoothing and Tapering Extensions

References

Sample Path of a Poisson Process

Poisson Process Sample Path

t	N(t)
0	1
5	2
10	4
15	6
20	8
25	10
30	13
35	15
40	17
45	19
50	21

Navigation icons: back, forward, search, etc.

Time Series and Stochastic Processes.

John Fricks

Introduction

Time Domain Poisson Process State Space Model & Kalman Filter Filtering and the Likelihood Function

Spectral Analysis The Periodogram The Periodogram and Regression The Periodogram and the Spectral Density Smoothing and Tapering Extensions

References

Properties of Poisson Process

- **Independent Increments.** $N(t) - N(s)$ and $N(v) - N(u)$ are independent whenever $(s, t]$ and $(u, v]$ do not overlap.
- **Stationary Increments.** The distribution of $X(t) - X(s)$ depends only on the length of the interval, $t - s$.
- **Mean of a Poisson process,** $EN(t) = \lambda t$.
- **Variance of a Poisson process,** $\text{Var}N(t) = \lambda t$.

Navigation icons: back, forward, search, etc.

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain
Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

Spectral
Analysis
The
Periodogram
The
Periodogram
and Regression
The
Periodogram
and the
Spectral
Density
Smoothing and
Tapering
Extensions

References

Poisson Process Alternative Definition #1

$N(t)$ is a stochastic process with

- $N(0) = 0$.
- $N(t)$ has independent increments
- In a small interval of length Δ ,
 - $P(N(t + \Delta) - N(t) = 1)$ is approximately $\lambda\Delta$
 - $P(N(t + \Delta) - N(t) > 1)$ is negligible.



Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain
Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

Spectral
Analysis
The
Periodogram
The
Periodogram
and Regression
The
Periodogram
and the
Spectral
Density
Smoothing and
Tapering
Extensions

References

Kolmogorov Forward Equations

From this alternative definition #1, one may obtain the Master equations for $p_i(t) = P(X(t) = i)$. These are also known as the Kolmogorov forward equations.

$$p'_i(t) = \lambda p_{i-1}(t) - \lambda p_i(t) \quad i = 1, 2, \dots$$

with

$$p'_0(t) = -\lambda p_0(t)$$

where $p_0(0) = 1$.



Time Series and Stochastic Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space Model & Kalman Filter

Filtering and the Likelihood Function

Spectral Analysis

The Periodogram

The Periodogram and Regression

The Periodogram and the Spectral Density

Smoothing and Tapering

Extensions

References

Inference for a Poisson process.

Assume that one has data from a Poisson process at evenly spaced intervals.

$$N_1 = N(\Delta), N_2 = N(2\Delta), N_3 = N(3\Delta), \dots$$

- One may be tempted to use the last definition and consider your data as Bernoulli.
- Preferably, one may treat the data as a Poisson random sample with mean $\lambda\Delta$.
- The maximum likelihood estimator of λ in this case will be given as

$$\hat{\lambda} = \frac{\sum_{i=1}^n N(i\Delta) - N((i-1)\Delta)}{n\Delta}.$$

A set of small, light-blue navigation icons typically used in Beamer presentations, including symbols for back, forward, search, and table of contents.

Time Series and Stochastic Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space Model & Kalman Filter

Filtering and the Likelihood Function

Spectral Analysis

The Periodogram

The Periodogram and Regression

The Periodogram and the Spectral Density

Smoothing and Tapering

Extensions

References

Poisson Process Alternative Definition #2

$X(t)$ is a Poisson process if

- $X(0) = 0$.
- The interarrival times between the $i - 1$ st and i th event, T_i , are exponential and independent with mean $1/\lambda$.
- If data consists of nearly exact time of events, one may use the definition involving interarrival times and use the resulting random sample from an exponential distribution.
- Specifically in this case, the maximum likelihood estimator for λ is given by $\hat{\lambda} = \frac{n}{\sum_{i=1}^n T_i}$

A set of small, light-blue navigation icons typically used in Beamer presentations, including symbols for back, forward, search, and table of contents.

The figure is a scatter plot titled "X-ray photons". The vertical axis is labeled "N(t)" and ranges from 0 to 100 with major tick marks every 20 units. The horizontal axis is labeled "t" and ranges from 0 to 1500 with major tick marks every 500 units. The data points are represented by small black dots connected by a dashed line. The curve begins at the origin (0,0), remains at zero until approximately t = 100, then rises in a generally increasing, concave-downward fashion, reaching about 100 at t = 1500.

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

What is the intensity λ ?

The data.

```
[1] 25.0 33.7 37.8 47.3 50.5 98.4 99.4 107.9 141.1 161.4 164.3 210.4  
[13] 252.5 262.6 264.8 287.8 290.2 313.4 339.2 394.8 432.1 445.7 471.6 481.3  
[25] 584.7 632.3 644.5 664.2 666.5 695.4 696.8 700.6 710.6 712.0 726.1 737.9  
[37] 747.3 761.0 779.3 829.1 838.9 852.1 882.6 897.1 897.7 899.9 911.0 944.4  
[49] 961.3 975.4 985.5 1008.4 1011.0 1029.2 1032.1 1032.6 1046.0 1051.1 1051.9 1053.0  
[61] 1054.0 1078.0 1091.1 1091.7 1098.4 1101.1 1101.2 1125.7 1126.6 1136.9 1145.4 1152.2  
[73] 1152.4 1152.7 1167.4 1171.7 1196.7 1197.1 1253.8 1260.9 1267.8 1283.5 1333.4 1383.2  
[85] 1420.1 1447.0 1450.9 1463.7 1467.5 1471.6 1473.7 1488.2 1496.2 1502.5 1509.0 1516.0  
[97] 1516.9 1529.7 1534.1 1534.9 1550.1 1551.6 1586.5 1588.0 1590.0 1600.0 1600.8
```

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering
Extensions

References

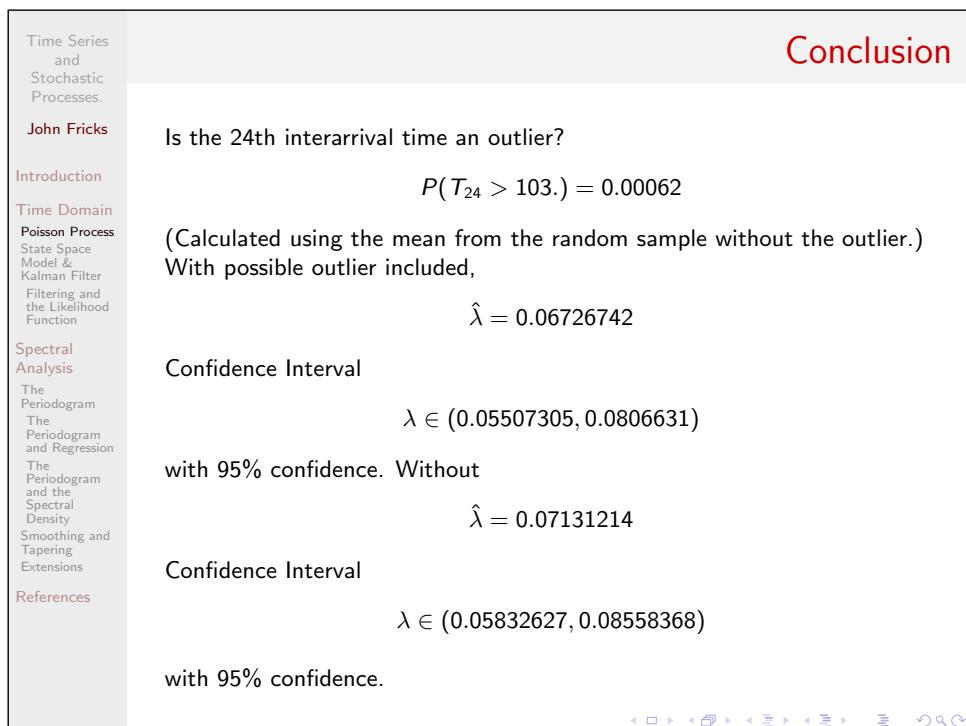
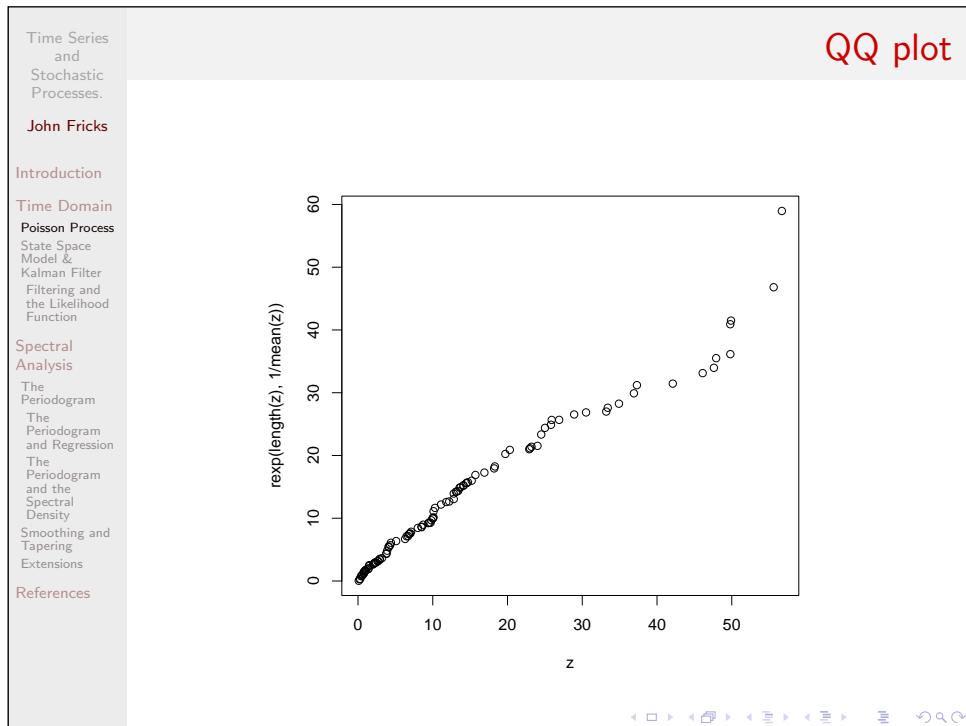
What is the intensity λ?

Difference the data to find the interarrival times.

[1]	8.7	4.1	9.5	3.2	47.9	1.0	8.5	33.2	20.3	2.9	46.1	42.1	10.1	2.2
[15]	23.0	2.4	23.2	25.8	55.6	37.3	13.6	25.9	9.7	103.4	47.6	12.2	19.7	2.3
[29]	28.9	1.4	3.8	10.0	1.4	14.1	11.8	9.4	13.7	18.3	49.8	9.8	13.2	30.5
[43]	14.5	0.6	2.2	11.1	33.4	16.9	14.1	10.1	22.9	2.6	18.2	2.9	0.5	13.4
[57]	5.1	0.8	1.1	1.0	24.0	13.1	0.6	6.7	2.7	0.1	24.5	0.9	10.3	8.5
[71]	6.8	0.2	0.3	14.7	4.3	25.0	0.4	56.7	7.1	6.9	15.7	49.9	49.8	36.9
[85]	26.9	3.9	12.8	3.8	4.1	2.1	14.5	8.0	6.3	6.5	7.0	0.9	12.8	4.4
[99]	0.8	15.2	1.5	34.9	1.5	2.0	10.0	0.8						

A Quantile-Quantile (QQ) plot comparing two data series. The x-axis is labeled y and ranges from 0 to 100. The y-axis is labeled $\exp(\text{length}(y), 1/\text{mean}(y))$ and ranges from 0 to 60. The data points, represented by open circles, show a strong positive linear trend, indicating a good fit between the two variables.

y	$\exp(\text{length}(y), 1/\text{mean}(y))$
0	0
5	5
10	10
15	15
20	20
25	25
30	30
35	35
40	40
45	45
50	50
55	55
60	60
70	70
80	80
90	90
100	100



Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain
Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

Spectral
Analysis
The
Periodogram
The
Periodogram
and Regression
The
Periodogram
and the
Spectral
Density
Smoothing and
Tapering
Extensions

References

Poisson Process Generalizations

- Renewal Processes
- Nonhomogeneous Poisson Process—intensity function is a deterministic function in time.
- Birth/Death processes—rates of jumps are state dependent.
- Cox Process—intensity function is a stochastic process.
- Marked Poisson process
- Poisson random field.

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain
Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

Spectral
Analysis
The
Periodogram
The
Periodogram
and Regression
The
Periodogram
and the
Spectral
Density
Smoothing and
Tapering
Extensions

References

Autoregressive (AR) Model

A basic time series model related to regression is the Autoregressive Model (AR)

$$x_t = \phi x_{t-1} + w_t$$

for $t = 1, \dots, n$ where ϕ is a constant and w_t is a sequence of independent and identically distributed normal random variables (often called a white noise sequence in this context). Note that the result of this model will be a single *dependent* series—a time series, x_1, \dots, x_t . This contrasts the regression model which relates two variables.

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

Vector Autoregressive Model

An obvious generalization of this model is a vector version

$$\mathbf{x}_t = \Phi \mathbf{x}_{t-1} + w_t$$

for $t = 1, \dots, n$ where $\mathbf{x}_t = (x_{t1}, \dots, x_{tp})'$ and w_t is a sequence of independent $p \times 1$ normal random vectors with covariance matrix Q . The matrix Φ is a $p \times p$ matrix.

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain
Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

The Linear State Space Model

Now, imagine that we cannot actually observe our system of interest \mathbf{x}_t which is a Vector Autoregressive Model. Instead we may observe a linear transformation of x_t with additional observational error. In other words, the complete model is as follows:

$$\mathbf{y}_t = A\mathbf{x}_t + v_t \quad , \quad \mathbf{x}_t = \Phi\mathbf{x}_{t-1} + w_t$$

where \mathbf{y}_t is a $q \times 1$ vector, A is a $q \times p$ matrix, and v_t is a sequence of independent normal random variables with mean zero and covariance matrix R . Also, the sequence v_t is independent of the sequence w_t . The equation on the left is generally called the *observation equation*, and the equation on the right is called the *system equation*.

Time Series and Stochastic Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space Model & Kalman Filter

Filtering and the Likelihood Function

Spectral Analysis

The Periodogram

The Periodogram and Regression

The Periodogram and the Spectral Density

Smoothing and Tapering

Extensions

References

What can we do with the state space model?

- Maximum Likelihood estimation of the parameters (including standard errors of our estimates).
- Bayesian estimation of parameters.
- Filtering—conditional distribution of the systems given our observations. We will, therefore, have a “guess” of our unseen system, x_t given our observations y_t .
- Prediction—predict the next observation given the observations up to the current time.

Time Series and Stochastic Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space Model & Kalman Filter

Filtering and the Likelihood Function

Spectral Analysis

The Periodogram

The Periodogram and Regression

The Periodogram and the Spectral Density

Smoothing and Tapering

Extensions

References

Filtering

Suppose you may observe y_1, \dots, y_n , but you are really interested in x_1, \dots, x_n and estimating parameters such as ϕ . While you cannot “know” x_t , you can have an optimal estimate of x_t . The goal will be to calculate

$$p(x_t | y_t, \dots, y_1)$$

For a Gaussian model this means that you’ll know $E(x_t | y_t, \dots, y_1)$ (your guess) and also the conditional variance of x_t .

Time Series and Stochastic Processes.

John Fricks

Introduction

Time Domain

Poisson Process State Space Model & Kalman Filter Filtering and the Likelihood Function

Spectral Analysis

The Periodogram The Periodogram and Regression The Periodogram and the Spectral Density Smoothing and Tapering Extensions

References

Steps for Filtering

Here's an outline of how that works—assume that you know all the parameters. Assume that you have the guess at the last time step, i.e. $p(x_{t-1}|y_{t-1}, \dots, y_1)$.

- ① **Predict** the next system observation based on what you have.

$$p(x_t|y_1, \dots, y_{t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{t-1}, \dots, y_1)dx_{t-1}$$

- ② Calculate the guess for the observation, y_t , based on this prediction.

$$p(y_t|y_{t-1}, \dots, y_1) = \int p(y_t|x_t, y_{t-1}, \dots, y_1)p(x_t|y_{t-1}, \dots, y_1)dx_t$$

- ③ Use Bayes rule to **update** the prediction for x_t with the current observation y_t

$$p(x_t|y_t, \dots, y_1) = \frac{p(x_t|y_{t-1}, \dots, y_1)p(y_t|x_t, y_{t-1}, \dots, y_1)}{p(y_t|y_{t-1}, \dots, y_1)}$$

Time Series and Stochastic Processes.

John Fricks

Introduction

Time Domain

Poisson Process State Space Model & Kalman Filter Filtering and the Likelihood Function

Spectral Analysis

The Periodogram The Periodogram and Regression The Periodogram and the Spectral Density Smoothing and Tapering Extensions

References

The Likelihood Function

- Remember that the likelihood function is simply the density of the data evaluated at the observations.

$$p(y_T, \dots, y_1) = \prod_{t=1}^T p(y_t|y_{t-1}, \dots, y_1)$$

Now, we have a likelihood to maximize to obtain parameters such as ϕ .

- When the errors are Gaussian, finding $p(x_t|y_t, \dots, y_1)$ for each t is known as calculating the Kalman filter. In general, this density is called the filter density. These calculations are reduced to matrix operations in the linear Gaussian case.

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

- Poisson Process
- State Space Model & Kalman Filter
- Filtering and the Likelihood Function

Spectral Analysis

- The Periodogram
- The Periodogram and Regression
- The Periodogram and the Spectral Density
- Smoothing and Tapering
- Extensions

References

Introduction

- Spectral analysis decomposes a sequence of numbers into its frequency components.
- What are we trying to estimate?
- The periodogram and its twin interpretations.
 - Finding periodic trends through regression. Non-constant mean.
 - Spectral decomposition of the autocovariance function. Constant mean.



Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

- Poisson Process
- State Space Model & Kalman Filter
- Filtering and the Likelihood Function

Spectral Analysis

- The Periodogram
- The Periodogram and Regression
- The Periodogram and the Spectral Density
- Smoothing and Tapering
- Extensions

References

Finding periodic trends through regression

- In this case, we are assuming a model of the following form:

$$x_t = \sum_{i=1}^p A_i \cos(2\pi\omega_i t + \phi_i) + e_t$$

where $Ee_t = 0$. Assuming we know ω_i , this is a non-linear regression problem.

- This is equivalent to a linear regression problem.

$$x_t = \sum_{i=1}^p (\beta_{1i} \cos(2\pi\omega_i t) + \beta_{2i} \sin(2\pi\omega_i t)) + e_t.$$

- We will see that the periodogram may be used to identify the ω_i .



Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

Discrete Fourier Transform of the Time Series

A Fourier transform of the data moves the data from the time domain to the frequency domain. For our time series, x_1, \dots, x_n , the discrete Fourier transform would be

$$d(\omega_j) = n^{-1/2} \sum_{t=1}^n x_t \exp(-2\pi i t \omega_j)$$

where $\omega_j = 0, 1/n, \dots, (n-1)/n$.

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

An Alternate Representation

Note that we can break up $d(\omega_j)$ into two parts

$$d(\omega_j) = n^{-1/2} \sum_{t=1}^n x_t \cos(2\pi i \omega_j t) - i n^{-1/2} \sum_{t=1}^n x_t \sin(2\pi i \omega_j t)$$

which we could write as as a cosine component and a sine component

$$d(\omega_j) = d_c(\omega_j) - i d_s(\omega_j)$$

Time Series and Stochastic Processes.
John Fricks
Introduction
Time Domain Poisson Process State Space Model & Kalman Filter Filtering and the Likelihood Function
Spectral Analysis
The Periodogram
The Periodogram and Regression
The Periodogram and the Spectral Density
Smoothing and Tapering Extensions
References

Fourier Basis

We may use an inverse Fourier transform to rewrite the data as

$$\begin{aligned}
 x_t &= n^{-1/2} \sum_{j=1}^n d(\omega_j) e^{2\pi i \omega_j t} \\
 &= n^{-1/2} \sum_{j=1}^n d(\omega_j) e^{2\pi i \omega_j t} \\
 &= a_0 + n^{-1/2} \sum_{j=1}^m d(\omega_j) e^{2\pi i \omega_j t} + n^{-1/2} \sum_{j=m+1}^n d(\omega_j) e^{2\pi i \omega_j t} \\
 &= a_0 + \sum_{j=1}^m \frac{2d_c(\omega_j)}{n^{-1/2}} \cos(2\pi i \omega_j t) + \sum_{j=1}^m \frac{2d_s(\omega_j)}{n^{-1/2}} \sin(2\pi i \omega_j t)
 \end{aligned}$$

where $m = \lfloor \frac{n}{2} \rfloor$



Time Series and Stochastic Processes.
John Fricks
Introduction
Time Domain Poisson Process State Space Model & Kalman Filter Filtering and the Likelihood Function
Spectral Analysis
The Periodogram
The Periodogram and Regression
The Periodogram and the Spectral Density
Smoothing and Tapering Extensions
References

Fourier Transform as Regression

- We can think of the Fourier transform as a regression of x_t on sines and cosines.

$$x_t = a_0 + \sum_{j=1}^m \frac{2d_c(\omega_j)}{n^{-1/2}} \cos(2\pi i \omega_j t) + \sum_{j=1}^m \frac{2d_s(\omega_j)}{n^{-1/2}} \sin(2\pi i \omega_j t)$$

- The coefficients of this regression are equal to $2/\sqrt{n}$ times the sine part and the cosine part of the Fourier transforms respectively.
- Also note that if our time series is Gaussian, $d_c(\omega_j)$ and $d_s(\omega_j)$ are Gaussian random variables.



Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

The Periodogram

- The periodogram is defined as
$$I(\omega_j) = |d(\omega_j)|^2 = d_c^2(\omega_j) + d_s^2(\omega_j)$$
- If there is no periodic trend in the data, then $Ed(\omega_j) = 0$, and the periodogram expresses the variance of x_t at frequency ω_j .
- If a periodic trend exists in the data, then $Ed(\omega_j)$ will be the contribution to the periodic trend at the frequency ω_j .

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

Example of Periodogram for Regression.

Pulsar Example(<http://xweb.nrl.navy.mil/timeseries/herx1.diskette>)

A time series plot showing a pulsar signal over 1500 units of time. The y-axis is labeled 'x' and ranges from 50 to 350. The x-axis is labeled 'Time' and ranges from 0 to 1500. The signal shows a strong periodic component with a period of approximately 100 units of time, superimposed on high-frequency noise.

Time Series and Stochastic Processes.

John Fricks

Introduction

Time Domain

- Poisson Process
- State Space Model & Kalman Filter
- Filtering and the Likelihood Function

Spectral Analysis

- The Periodogram
- The Periodogram and Regression**
- The Periodogram and the Spectral Density
- Smoothing and Tapering
- Extensions

References

Example of Periodogram for Regression.

$x[1:200]$

Time

Navigation icons: back, forward, search, etc.

Time Series and Stochastic Processes.

John Fricks

Introduction

Time Domain

- Poisson Process
- State Space Model & Kalman Filter
- Filtering and the Likelihood Function

Spectral Analysis

- The Periodogram
- The Periodogram and Regression**
- The Periodogram and the Spectral Density
- Smoothing and Tapering
- Extensions

References

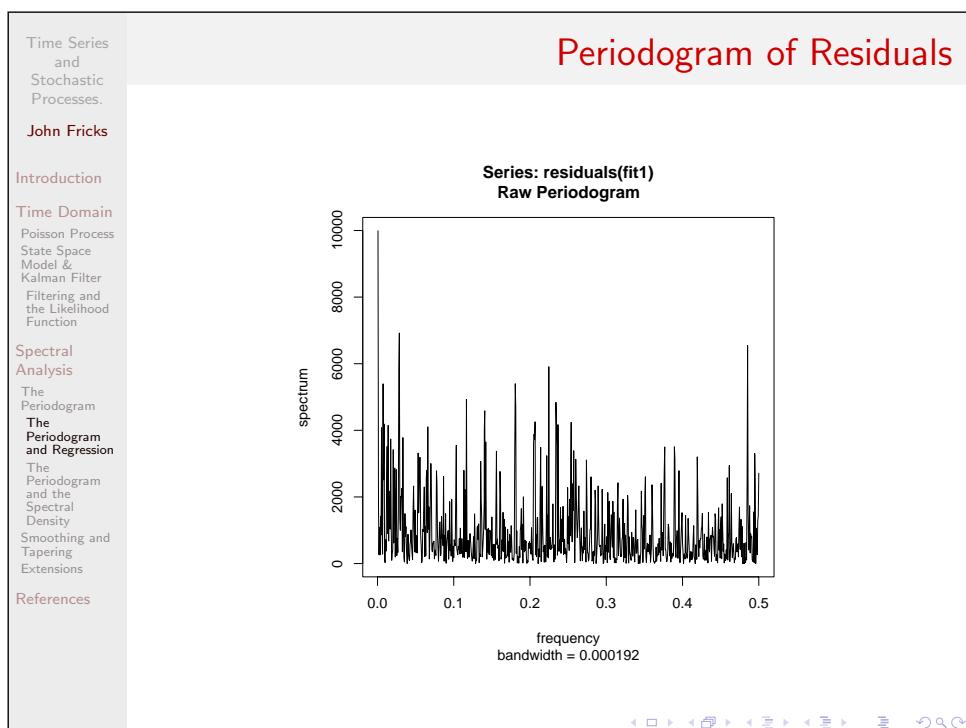
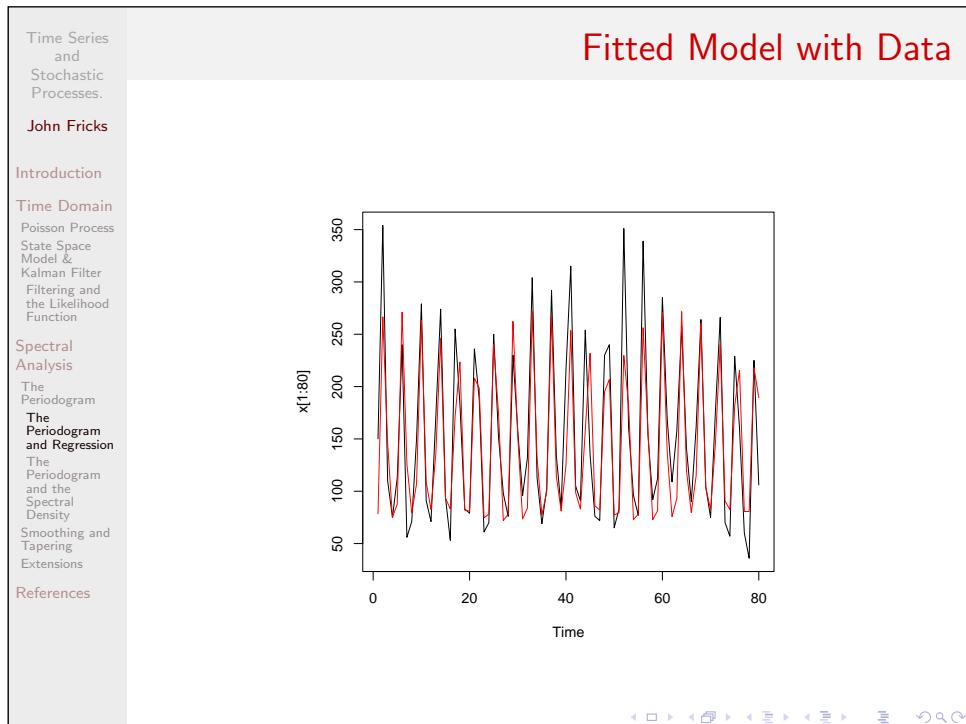
Periodogram of Pulsar Example

Series: x
Raw Periodogram

spectrum

frequency
bandwidth = 0.000192

Navigation icons: back, forward, search, etc.



The figure shows the ACF (Autocorrelation Function) plot for the residuals of a fitted model. The title of the plot is "Series fit1\$residuals". The y-axis is labeled "ACF" and ranges from 0.0 to 1.0 with increments of 0.2. The x-axis is labeled "Lag" and ranges from 0 to 200 with increments of 50. The data points are represented by vertical black bars. A solid blue horizontal line at approximately 0.05 represents the confidence interval, and a dashed blue horizontal line at approximately -0.05 represents the lower bound. The autocorrelation values are mostly within the bounds, indicating no significant autocorrelation.

The Spectral Density

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram
The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density
Smoothing and
Tapering
Extensions

References

Assume now that x_t is a stationary process with autocovariance function $\gamma(h)$ and expected value $E x_t = \mu$. The spectral density is the Fourier transform of the autocovariance function

$$f(\omega) = \sum_{h=-\infty}^{h=\infty} e^{-2\pi i \omega h} \gamma(h)$$

for $\omega \in (-0.5, 0.5)$. Note that this is a population quantity.
(i.e. This is a constant quantity defined by the model.)

Periodogram and the Autocovariance

Why is the periodogram an estimate for the spectral density?

Let m be the sample mean of our data.

$$\begin{aligned}
 I(\omega_j) &= |d(\omega_j)|^2 = n^{-1} \sum_{t=1}^n x_t e^{-2\pi i \omega_j t} \overline{\sum_{t=1}^n x_t e^{-2\pi i \omega_j t}} \\
 &= |d(\omega_j)|^2 = n^{-1} \sum_{t=1}^n \sum_{s=1}^n (x_t - m)(\overline{x_s} - m) e^{-2\pi i \omega_j(t-s)} \\
 &= n^{-1} \sum_{h=-(n-1)}^{(n-1)} \sum_{t=1}^{n-|h|} (x_{t+|h|} - m)(x_t - m) e^{-2\pi i \omega_j(h)} \\
 &= \sum_{h=-(n-1)}^{(n-1)} \hat{\gamma}(h) e^{-2\pi i \omega_j(h)} \approx f(\omega_j)
 \end{aligned}$$

(note that $h = t - s$.)

Smoothing

- Is the periodogram a **good** estimator for the spectral density? Not really!
- The periodogram, $I(\omega_1), \dots, I(\omega_m)$, attempt to estimate parameters $f(\omega_1), \dots, f(\omega_m)$. We have nearly the same number of parameters as we have data.
- Moreover, the number of parameters grow as a constant proportion of the data. Therefore, the periodogram is NOT a consistent estimator of the spectral density.

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

Moving Average

- A simple way to improve our estimates is to use a moving average smoothing technique

$$\hat{f}(\omega_j) = \frac{1}{2m+1} \sum_{k=-m}^m l(\omega_{j-k})$$

- We can also iterate this procedure of uniform weighting to be more weight on closer observations.

$$\hat{u}_t = \frac{1}{3} u_{t-1} + \frac{1}{3} u_t + \frac{1}{3} u_{t+1}$$

Then, we iterate.

$$\hat{u}_t = \frac{1}{3} \hat{u}_{t-1} + \frac{1}{3} \hat{u}_t + \frac{1}{3} \hat{u}_{t+1}$$

Then, substitute to obtain better weights.

A scatter plot showing the spectral density of an AR(3) process. The x-axis is labeled "freq" and ranges from 0.0 to 0.5. The y-axis is labeled "var" and ranges from 0 to 6000. The data points, represented by open circles, show a clear peak at approximately freq = 0.09. A solid black line represents a fitted curve, which follows the general shape of the data points, peaking at the same frequency and then decaying towards zero as freq increases.

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

Smoothing Summary



Tapering

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain
Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

Spectral
Analysis
The
Periodogram
The
Periodogram
and Regression
The
Periodogram
and the
Spectral
Density
Smoothing and
Tapering
Extensions
References

Why do we need to taper?

Our theoretical model $\dots, x_{-1}, x_0, x_1, \dots$ consists of a doubly infinite time series. We could think of our data, y_t as the following transformation of the model

$$y_t = h_t x_t$$

where $h_t = 1$ for $t = 1, \dots, n$ and zero otherwise. This has repercussions on the expectation of the periodogram of our data.

$$E[I_y(\omega_j)] = \int_{-0.5}^{0.5} W_n(\omega_j - \omega) f_x(\omega) d\omega$$

where $W_n(\omega) = |H_n(\omega)|^2$ and $H_n(\omega)$ is the Fourier transform of the sequence h_t .



Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain
Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

Spectral
Analysis
The
Periodogram
The
Periodogram
and Regression
The
Periodogram
and the
Spectral
Density
Smoothing and
Tapering
Extensions
References

The Taper

Specifically,

$$H_n(\omega) = \frac{1}{\sqrt{n}} \sum_{t=1}^n h_t e^{-2\pi i \omega t}$$

When we put in the h_t above, we obtain a spectral window of

$$W_n(\omega) = \frac{\sin^2(n\pi\omega)}{\sin^2(\pi\omega)}.$$

We set $W_n(0) = n$.



Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain
Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

Spectral
Analysis
The
Periodogram
The
Periodogram
and Regression
The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering
Extensions

References

Fejer window

There are problems with this spectral window, namely there is too much weight on neighboring frequencies (sidelobes).

Fejer window, n=480

Fejer window (log), n=480

Navigation icons: back, forward, search, etc.

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain
Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

Spectral
Analysis
The
Periodogram
The
Periodogram
and Regression
The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering
Extensions

References

Cosine Taper

One way to fix this is to use a Cosine taper. We select a transform h_t to be

$$h_t = 0.5 \left[1 + \cos \left(\frac{2\pi(t - \bar{t})}{n} \right) \right]$$

Fejer window, n=480, L=9

Fejer window(log), n=480, L=9

Full Tapering Window, n=480, L=9

Full Tapering Window(log), n=480, L=9

Navigation icons: back, forward, search, etc.

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

Full Tapering

Full Tapering, n=480, transformation in time domain

Full Tapering Window, n=480, L=9

Full Tapering Window(log), n=480, L=9

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space

Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

50 % Tapering

50% Tapering, n=480, transformation in time domain

50% Tapering Window, n=480, L=9

50% Tapering Window(log), n=480, L=9

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

- Poisson Process
- State Space Model & Kalman Filter
- Filtering and the Likelihood Function

Spectral Analysis

- The Periodogram
- The Periodogram and Regression
- The Periodogram and the Spectral Density

Smoothing and Tapering

- Extensions

References

Smoothing and Tapering

- Smoothing introduces bias, but reduces variance.
- Smoothing tries to solve the problem of too many “parameters”.
- Tapering decreases bias and introduces variance.
- Tapering attempts to diminish the influence of sidelobes that are introduced via the spectral window.

Navigation icons

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

- Poisson Process
- State Space Model & Kalman Filter
- Filtering and the Likelihood Function

Spectral Analysis

- The Periodogram
- The Periodogram and Regression
- The Periodogram and the Spectral Density

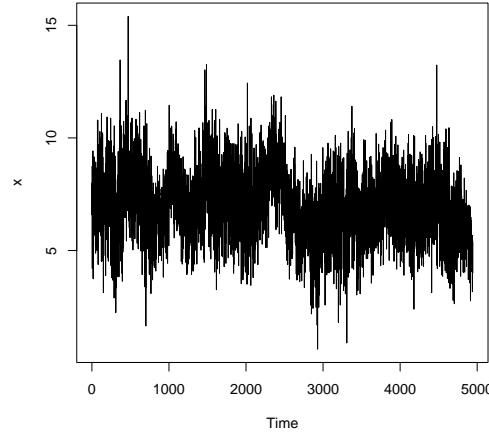
Smoothing and Tapering

- Extensions

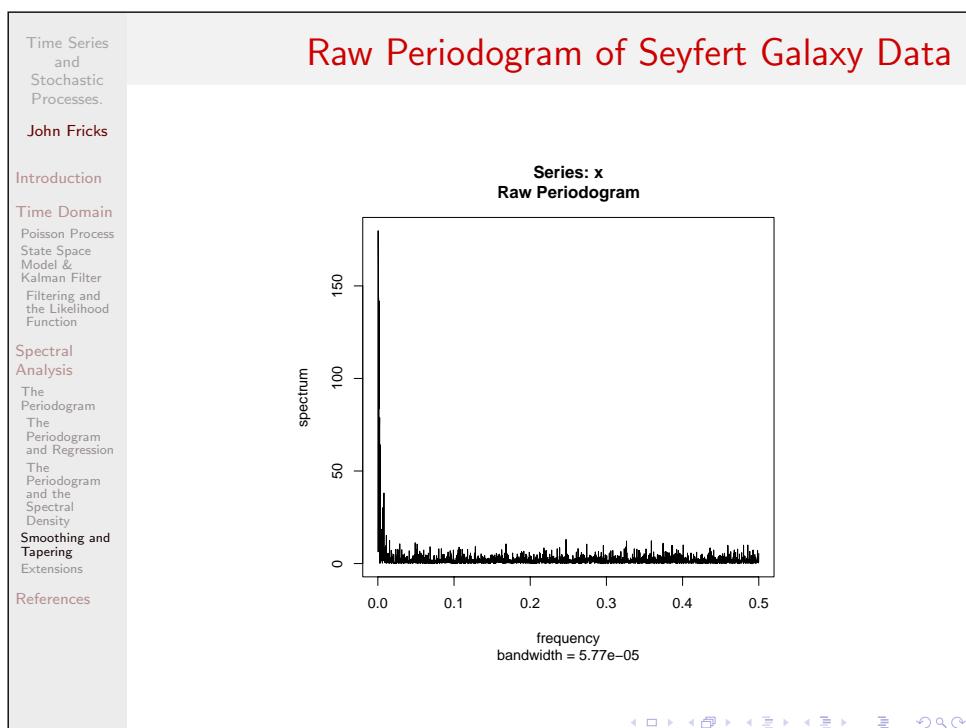
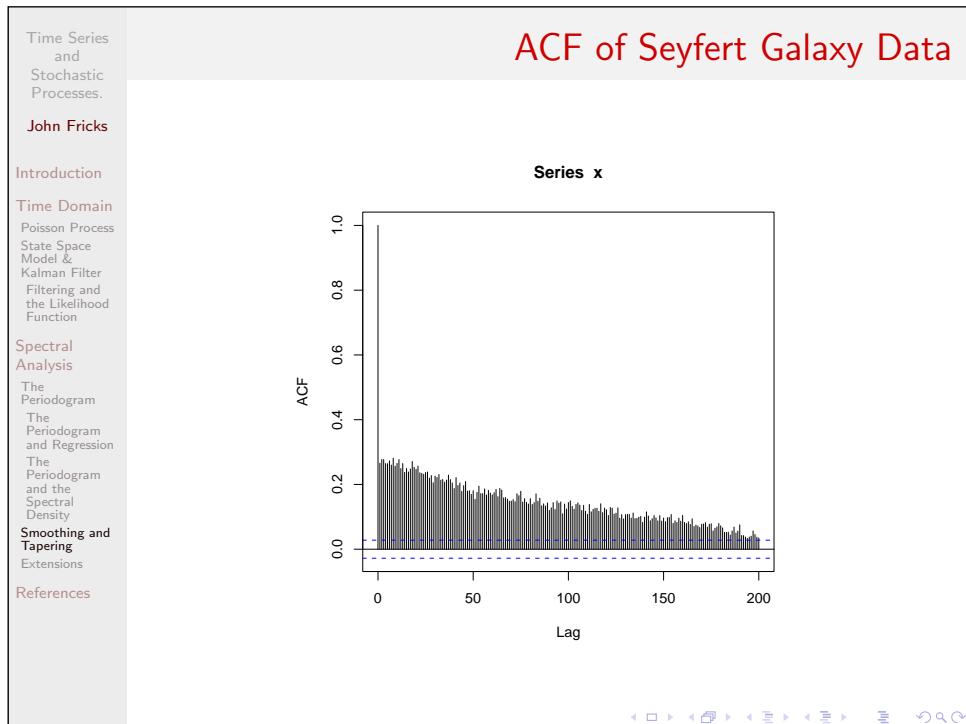
References

Example Estimating the Spectral Density

Fractal Time Variability in a Seyfert Galaxy.
[\(<http://xweb.nrl.navy.mil/timeseries/multi.diskette>\)](http://xweb.nrl.navy.mil/timeseries/multi.diskette)



Navigation icons



Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

Smoothed Periodogram of Seyfert Galaxy Data

Series: x
Smoothed Periodogram

A smoothed periodogram plot titled "Smoothed Periodogram" for "Series: x". The y-axis is labeled "spectrum" and ranges from 0 to 40 with major ticks at 0, 10, 20, 30, and 40. The x-axis is labeled "frequency" and ranges from 0.0 to 0.5 with major ticks every 0.1 units. The plot shows a very high spectrum value near zero frequency (approximately 45) which drops sharply to a baseline around 2-3 as frequency increases. A horizontal line at approximately y=2 represents the noise level. The plot includes text at the bottom indicating "bandwidth = 0.00408".

frequency
bandwidth = 0.00408

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

More Smoothed Periodogram of Seyfert Galaxy Data

Series: x

Smoothed Periodogram

A line graph titled "Smoothed Periodogram" showing the spectrum versus frequency. The y-axis is labeled "spectrum" and ranges from 0 to 25. The x-axis is labeled "frequency" and ranges from 0.0 to 0.5. A single data series is plotted, labeled "x". The spectrum is very high at low frequencies, peaking near 25 at frequency 0.0, and then drops sharply to a noisy baseline around 2. The plot includes a title "Series: x" and "Smoothed Periodogram" centered above the plot area. Below the plot, the text "frequency bandwidth = 0.00817" is displayed. The plot area is enclosed in a black rectangular frame.

frequency
bandwidth = 0.00817

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain

Poisson Process

State Space
Model &
Kalman Filter

Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram

The
Periodogram
and Regression

The
Periodogram
and the
Spectral
Density

Smoothing and
Tapering

Extensions

References

Oversmoothed Periodogram of Seyfert Galaxy Data

Series: x
Smoothed Periodogram

A line graph titled "Smoothed Periodogram" for "Series: x". The y-axis is labeled "spectrum" and ranges from 0 to 15 with major ticks at 5, 10, and 15. The x-axis is labeled "frequency" and ranges from 0.0 to 0.5 with major ticks every 0.1 units. The plot shows a single data series as a solid black line. It starts at a spectrum value of approximately 18 at frequency 0.0, drops sharply to about 2 at frequency 0.05, and then remains relatively flat with minor fluctuations between 2 and 4 for the rest of the frequency range.

frequency
bandwidth = 0.0122

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain
Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

Spectral
Analysis
The
Periodogram
The
Periodogram
and Regression
The
Periodogram
and the
Spectral
Density
Smoothing and
Tapering
Extensions
References

Dynamic Fourier Analysis

- What can be done for non-stationary data?
- One approach is to decompose our time series as a sum of a non-constant (deterministic) trend plus a stationary “noise” term:
$$x_t = \mu_t + y_t$$
- What if our data instead appears as a stationary model locally, but globally the model appears to shift? One approach is to divide the data into shorter sections (perhaps overlapping) and
- This approach is developed in Shumway and Stoffer. One essentially looks at how the spectral density changes over time.

Seismic Data

Time Frequency Plot for Earthquake Series

The figure is a 3D surface plot representing the power spectrum of an earthquake series over time and frequency. The vertical axis is labeled "Power" and ranges from 0.0 to 1.0. The horizontal axis is labeled "frequency" and ranges from 0.0 to 0.5. The depth axis is labeled "time" and ranges from 0 to 1500. The surface shows several sharp peaks, with the most prominent one occurring at a frequency of approximately 0.25 and a time of approximately 1000, reaching a power of about 0.8. There are also smaller peaks at higher frequencies and times.

Time Series and Stochastic Processes.

John Fricks

Introduction

Time Domain

- Poisson Process
- State Space Model & Kalman Filter
- Filtering and the Likelihood Function

Spectral Analysis

- The Periodogram
- The Periodogram and Regression
- The Periodogram and the Spectral Density
- Smoothing and Tapering
- Extensions

References

Time Frequency Plot for Explosion Series

Navigation icons: back, forward, search, etc.

Time Series and Stochastic Processes.

John Fricks

Introduction

Time Domain

- Poisson Process
- State Space Model & Kalman Filter
- Filtering and the Likelihood Function

Spectral Analysis

- The Periodogram
- The Periodogram and Regression
- The Periodogram and the Spectral Density
- Smoothing and Tapering
- Extensions

References

Wavelets

- We have been using Fourier components as a basis to represent stationary processes and seasonal trends.
- Since we are dealing with finite data, we must use a finite number of terms, and perhaps one could use an alternative basis.
- Wavelets are one option to accomplish this goal. They are particularly well suited to the same situation as dynamic Fourier analysis.

Navigation icons: back, forward, search, etc.

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain
Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram
The
Periodogram
and Regression
The
Periodogram
and the
Spectral
Density
Smoothing and
Tapering
Extensions

References

References

- General Time Series
 - Robert Shumway and David Stoffer. *Time Series Analysis and Its Applications*. Springer NY, 2006.
 - Peter Brockwell and Richard Davis. *Time Series: Theory and Methods, Second Ed.* Springer NY, 1991.
- Spectral Analysis
 - David Brillinger. *Time Series: Data Analysis and Theory*. SIAM, 2001.
 - Donald Percival and Andrew Walden. *Spectral Analysis for Physical Applications*. Cambridge University Press, 1993.
 - Donald Percival and Andrew Walden. *Wavelet Methods for Time Series Analysis*. Cambridge University Press, 2000.
 - Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

Time Series
and
Stochastic
Processes.

John Fricks

Introduction

Time Domain
Poisson Process
State Space
Model &
Kalman Filter
Filtering and
the Likelihood
Function

Spectral
Analysis

The
Periodogram
The
Periodogram
and Regression
The
Periodogram
and the
Spectral
Density
Smoothing and
Tapering
Extensions

References

References

- Introductory Stochastic Processes
 - Sheldon Ross. *Introduction to Probability Models*. Elsevier, 2006.
 - Paul Hoel, Sidney Port, and Charles Stone *Introduction to Stochastic Processes*. Waveland Press, Inc. 1986.
 - Samuel Karlin and Howard Taylor. *A First Course in Stochastic Processes*. Elsevier, 1975.
- Statistical Inference for Counting Processes
 - James Lindsey. *The Statistical Analysis of Stochastic Processes in Time*. Cambridge, 2004.
 - Alan Karr. *Point Processes and Their Statistical Inference*. Marcel Dekker, 1991.
 - Per Kragh Andersen, Richard Gill, Ornulf Borgan, and Niels Keiding. *Statistical Models Based on Counting Processes*. Springer, 1993.

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 $m = \int g(x) f(x) dx$. 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 $\text{Normal}(0,1)$ 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 $g(Y_1)w(Y_1), g(Y_2)(Y_1), \dots, g(Y_N)(Y_1)$ 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.

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

Our example uses a dataset from the Chandra Orion Ultradeep Project (COUP). 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 algorithm in [full conditional distributions and M-H algorithm \(pdf\)](#).

Programming an MCMC algorithm in R

We will need an editor for our program. For instance, we can use Wordpad (available under the Start button menu under Accessories). Ideally, a more 'intelligent' editor such as emacs (with ESS or emacs speaks statistics installed) should be used to edit R programs.

Please save code from [MCMC template in R](#)

into a file and open this file using the editor. Save this file as MCMCchpt.R .

Note that in this version of the code, all parameters are sampled except for k (which is fixed at our guessed change point).

To load the program from the file MCMCchpt.R we use the "source" command. (Reminder: It may be helpful to type: `setwd("V:/")` to set the default directory to the place where you can save your files)

```
source("MCMCchpt.R") # with appropriate pathname
```

We can now run the MCMC algorithm:

```
mchain <- mhsampler(NUNIT=1000,dat=Y) # call the function with appropriate arguments
```

MCMC output analysis

Now that we have output from our sampler, we can treat these samples as data from which we can estimate quantities of interest. For instance, to estimate the expectation of a marginal distribution for a particular parameter, we would simply average all draws for that parameter so to obtain an estimate of $E(\theta)$:

```
mean(mchain[1,]) # obtain mean of first row (thetas)
```

To get estimates for means for all parameters:

```
apply(mchain,1,mean) # compute means by row (for all parameters at once)
apply(mchain,1,median) # compute medians by row (for all parameters at once)
```

To obtain an estimate of the entire posterior distribution:

```
plot(density(mchain[1,]),main="smoothed density plot for theta posterior")
plot(density(mchain[2,]),main="smoothed density plot for lambda posterior")
hist(mchain[3,],main="histogram for k posterior")
```

To find the (posterior) probability that lambda is greater than 10

```
sum(mchain[2,>10])/length(mchain[2,])
```

Now comment the line that fixes k at our guess (add the # mark) :

```
# currk <- KGUESS
```

Rerun the sampler with k also sampled.

```
mchain <- mhsampler(NUMIT=1000,dat=Y)
```

With the new output, you can repeat the calculations above (finding means, plotting density estimates etc.)

You can also study how your estimate for the expectation of the posterior distribution for k changes with each iteration.

```
estvssamp(mchain[3,])
```

We would like to assess whether our Markov chain is moving around quickly enough to produce good estimates (this property is often called 'good mixing'). While this is in general difficult to do rigorously, estimates of the autocorrelation

in the samples is an informal but useful check. To obtain sample autocorrelations we use the acf plot function:

```
acf(mchain[1,],main="acf plot for theta")
acf(mchain[2,],main="acf plot for lambda")
acf(mchain[3,],main="acf plot for k")
acf(mchain[4,],main="acf plot for b1")
acf(mchain[5,],main="acf plot for b2")
```

If the samples are heavily autocorrelated we should rethink our sampling scheme or, at the very least, run the chain for much longer. Note that the autocorrelations are negligible for all parameters except k which is heavily autocorrelated. This is easily resolved for this example since the sampler is fast (we can run the chain much longer very easily). In problems where producing additional samples is more time consuming, such as complicated high dimensional problems, improving the sampler 'mixing' can be much more critical.

Why are there such strong autocorrelations for k? The acceptance rate for k proposals (printed out with each MCMC run) are well below 10% which suggests that k values are stagnant more than 90% of the time. A better proposal for the Metropolis-Hastings update of a parameter can help improve acceptance rates which often, in turn, reduces autocorrelations. Try another proposal for k and see how it affects autocorrelations. In complicated problems, carefully constructed proposals can have a major impact on the efficiency of the MCMC algorithm.

How do we choose starting values?

In general, any value we believe would be reasonable under the posterior distribution will suffice. You can experiment with different starting values. For instance: modify the starting value for k in the function (for instance, try setting k=10), "source" the function in R and run the sampler again as follows:

```
mchain2 <- mhsampler(NUMIT=1000,dat=Y)
```

You can study how your estimate for the expectation of the posterior distribution for k changes with each iteration.

```
estvssamp(mchain2[3,])
```

Assessing accuracy and determining chain length

There are two important issues to consider when we have draws from an MCMC algorithm: (1) how do we assess the accuracy of our estimates based on the sample (how do we compute Monte Carlo standard errors?) (2) how long do we run the chain before we feel confident that our results are reasonably accurate ?

Regarding (1): Computing standard errors for a Monte Carlo estimate for an i.i.d. (classical Monte Carlo) sampler is easy, as shown for the toy example on estimating $P(-1 < X < 0)$ when X is a $\text{Normal}(0,1)$ random variable. Simply obtain the sample standard deviation of the $g(x_i)$ values and divide by square root of n (the

number of samples). Since Markov chains produce dependent draws, computing precise Monte Carlo standard errors for such samplers is a very difficult problem in general. For (2): Draws produced by classical Monte Carlo methods (such as rejection sampling) produced draws from the correct distribution. The MCMC algorithm produces draws that are asymptotically from the correct distribution. All the draws we see after a finite number of iterations are therefore only approximately from the correct distribution. Determining how long we have to run the chain before we feel sufficiently confident that the MCMC algorithm has produced reasonably accurate draws from the distribution is therefore a very difficult problem. Most rigorous solutions are too specific or tailored towards relatively simple situations while more general approaches tend to be heuristic.

There are many ways to compute Monte Carlo standard errors. Two simple but reasonable ways of calculating it:

the consistent batch means (bm) and the iterated monotone sequence estimator (imse) method in R.

References: Flegal, J.M., Haran, M., and Jones, G.L. (2008) Markov chain Monte Carlo: Can we trust the third significant figure? *Statistical Science (in press)*, and Geyer, C.J. (1992) Practical Markov chain Monte Carlo, *Statistical Science*.

To compute MC s.error via batch means, download the bm function from the batchmeans.R file above and source the file into R. We can now calculate standard error estimates for each of the five parameter estimates:

```
bm(mchain[1,])
bm(mchain[2,])
bm(mchain[3,])
bm(mchain[4,])
bm(mchain[5,])
```

Are these standard errors acceptable ?

There is a vast literature on different proposals for dealing with the latter issue (how long to run the chain) but they are all heuristics at best. The links at the bottom of this page (see section titled "Some resources") provide references to learn more about suggested solutions. One method that is fairly simple, theoretically justified in some cases and seems to work reasonably well in practice is as follows: run the MCMC algorithm and periodically compute Monte Carlo standard errors. Once the Monte Carlo standard errors are below some (user-defined) threshold, stop the simulation.

Often MCMC users do not run their simulations long enough. For complicated problems run lengths in the millions (or more) are typically suggested (although this may not always be feasible). For our example run the MCMC algorithm again, this time for 100000 iterations (set NUMIT=100000).

```
mchain2 <- mhsampler(NUMIT=100000,dat=Y)
```

You can now obtain estimates of the posterior distribution of the parameters as before and compute the new Monte Carlo standard error. Note whether the estimates and corresponding MC standard error have changed with respect to the previous sampler.

Making changes to the model

If we were to change the prior distributions on some of the individual parameters, only relatively minor changes may be needed in the program. For instance if the Inverse Gamma prior on b1 and b2 were replaced by Gamma(0.01,100) priors on them, we would only have to change the lines in the code corresponding to the updates of b1 and b2 (we would need to perform a Metropolis-Hastings update of each parameter). The rest of the code would remain unchanged. Modifying the program to make it sample from the posterior for the modified model is a useful exercise. For the modified full conditionals see modified full conditional

An obvious modification to this model would be to allow for more than one change point. A very sophisticated model that may be useful in many change point problems is one where the number of change points is also treated as unknown. In this case the *number* of Poisson parameters (only two of them in our example: theta and lambda) is also unknown. The posterior distribution is then a mixture over distributions of varying dimensions (the dimensions change with the number of change points in the model). This requires an advanced version of the Metropolis-Hastings algorithm known as **reversible-jump Metropolis Hastings** due to Peter Green (*Biometrika*, 1995). Some related information is available at [the HSSS variable dimension MCMC workshop](#).

Some resources

The "[CODA](#)" and [BOA](#)

packages in R implement many well known output analysis techniques. Charlie Geyer's [MCMC package in R](#) is another free resource. There is also MCMC software from the popular [WINBugs project](#).

In addition to deciding how long to run the sampler and how to compute Monte Carlo standard error, there are many possibilities for choosing how to update the parameters and more sophisticated methods used to make the Markov chain move around the posterior distribution efficiently. The literature on such methods is vast. The following [references](#) are a useful starting point.

Acknowledgment:

The model is borrowed from Chapter 5 of "Bayes and Empirical Bayes Methods for Data Analysis" by Carlin and Louis (2000). The data example was provided by Konstantin Getman (Penn State University).

Tbin	Cts
0	11
1	3
2	5
3	9
4	3
5	4
6	5
7	5
8	5
9	5
10	13
11	18
12	27
13	8
14	4
15	10
16	8
17	3
18	12
19	10
20	10
21	3
22	9
23	8
24	5
25	9
26	4
27	6
28	1
29	5
30	14
31	7
32	9
33	10
34	8
35	13
36	8
37	11
38	11
39	10
40	11
41	13
42	10
43	3
44	8
45	5

```

## Markov chain Monte Carlo algorithm for a Bayesian (single) change point model
## read in the data
## chptdat = read.table("chpt.dat",header=T)
## Y = chptdat$Ener
## chptdat = read.table("coal.dat",header=T)
## Y = chptdat$Deaths
KGUESS = 10 # our guess for k based on exploratory data analysis
## Note: this function is not written in the most efficient way since its purpose is primarily
##       to illustrate the MCMC process

mhsampler = function(NUMIT=1000,dat=Y)
{
  n = length(dat)
  cat("n=",n,"\\n")
  ## set up
  ## NUMIT x 5 matrix to store Markov chain values
  ## each row corresponds to one of 5 parameters in order: theta,lambda,k,b1,b2
  ## each column corresponds to a single state of the Markov chain
  mchain = matrix(NA, 5, NUMIT)
  acc = 0 # count number of accepted proposals (for k only)

  ## starting values for Markov chain
  ## This is somewhat arbitrary but any method that produces reasonable values for each parameter
  ## For instance, we can use approximate prior means or approximate MLEs.

  kinit = floor(n/2) # approximately halfway between 1 and n
  mchain[,1] = c(1,1,kinit,1,1)

  for (i in 2:NUMIT)
  {
    ## most upto date state for each parameter
    currtheta = mchain[1,i-1]
    currlambda = mchain[2,i-1]
    currk = mchain[3,i-1]
    currb1 = mchain[4,i-1]
    currb2 = mchain[5,i-1]

    ## sample from full conditional distribution of theta (Gibbs update)
    currtheta = rgamma(1,shape=sum(Y[1:currk])+0.5, scale=currb1/(currk*currb1+1))

    ## sample from full conditional distribution of lambda (Gibbs update)
    currlambda = rgamma(1,shape=sum(Y[(currk+1):n])+0.5, scale=currb2/((n-currk)*currb2+1))

    ## sample from full conditional distribution of k (Metropolis-Hastings update)
    propk = sample(x=seq(2,n-1), size=1) # draw one sample at random from uniform{2,...,n-1}

    ## Metropolis accept-reject step (in log scale)
    logMHRatio = sum(Y[1:propk])*log(currtheta)+sum(Y[(propk+1):n])*log(currlambda)-propk*log(theta)

    logalpha = min(0,logMHRatio) # alpha = min(1,MHRatio)
    if (log(runif(1))<logalpha) # accept if unif(0,1)<alpha, i.e. accept with probability alpha
    {
      acc = acc + 1 # increment count of accepted proposals
      currk = propk
    }

    currk = KGUESS # if we do not sample k (k fixed)

    ## sample from full conditional distribution of b1 (Gibbs update): draw from Inverse Gamma
    currb1 = 1/rgamma(1,shape=0.5, scale=1/(currtheta+1))

    ## sample from full conditional distribution of b2 (Gibbs update): draw from Inverse Gamma
    currb2 = 1/rgamma(1,shape=0.5, scale=1/(currlambda+1))

    ## update chain with new values
    mchain[,i] = c(currtheta,currlambda,currk,currb1,currb2)
  }
}

```

```
cat("Markov chain algorithm ran for ",NUMIT,"iterations (acc.rate for k=",acc/(NUMIT-1),  
cat("Parameters are in order: theta, lambda, k, b1, b2\n")  
return(mchain)  
}
```

```

## consistent batch means and imse estimators of Monte Carlo standard errors
## author: Murali Haran

## An R function for computing consistent batch means estimate of standard error from:
## Citation: Galin L. Jones, Murali Haran, Brian S. Caffo, and Ronald Neath, "Fixed-Width Or

## input: vals, a vector of N values (from a Markov chain),bs=batch size
## default bs (batch size) is "sqroot"=> number of batches is the square root of the run length
## if bs is "cuberoott", number of batches is the cube root of the run length
## output: list consisting of estimate of expected value and the Monte Carlo standard error

## NOTE: YOU DO NOT NEED TO DOWNLOAD THIS FILE TO RUN BATCHMEANS IN R
## SIMPLY USE THE COMMAND BELOW FROM YOUR R COMMAND LINE
## source("http://www.stat.psu.edu/~mharan/batchmeans.R")

# new version: Sep.12, 2005
## Input: vals is a vector of values from a Markov chain produced by the Metropolis-Hastings
## bs provides the batch size, either "sqroot" for the square root of the sample size (recom)
## or "cuberoott" for cube root of the sample size
bm <- function(vals,bs="sqroot",warn=FALSE)
{
  N <- length(vals)
  if (N<1000)
  {
    if (warn) # if warning
      cat("WARNING: too few samples (less than 1000)\n")
    if (N<10)
      return(NA)
  }

  if (bs=="sqroot")
  {
    b <- floor(sqrt(N)) # batch size
    a <- floor(N/b) # number of batches
  }
  else
    if (bs=="cuberoott")
    {
      b <- floor(N^(1/3)) # batch size
      a <- floor(N/b) # number of batches
    }
  else # batch size provided
  {
    stopifnot(is.numeric(bs))
    b <- floor(bs) # batch size
    if (b > 1) # batch size valid
      a <- floor(N/b) # number of batches
    else
      stop("batch size invalid (bs=",bs," )")
  }

  Ys <- sapply(1:a,function(k) return(mean(vals[((k-1)*b+1):(k*b)])))

  muhat <- mean(Ys)
  sigmahatsq <- b*sum((Ys-muhat)^2)/(a-1)

  bmse <- sqrt(sigmahatsq/N)

  return(list(est=muhat,se=bmse))
}

## apply bm to each col of a matrix of MCMC samples
## input: mcmat is a matrix with each row corresponding to a sample from the multivariate di
## skip = vector of columns to skip
## output: matrix with number of rows=number of dimensions of distribution and 2 columns (es
bmmat=function(mcmat,skip=NA)
{
  if (!any(is.na(skip)))

```

```

{
  num=ncol(mcmat)-length(skip)
  mcmat=mcmat[-skip] # remove columns to be skipped
}
else # assume it is NA
  num=ncol(mcmat)

bmvals=matrix(NA,num,2,dimnames=list(paste("V",seq(1,num),sep=""),c("est","se")))) # first

bmres=apply(mcmat,2,bm)
for (i in 1:num)
{
  bmvals[i,]=c(bmres[[i]]$est,bmres[[i]]$se)
}
return(bmvals)
}

## Geyer's initial monotone positive sequence estimator (Statistical Science, 1992)
## input: Markov chain output (vector)
## output: monte carlo standard error estimate for chain
imse <- function(outp,asymvar=FALSE)
{
  chainAC <- acf(outp,type="covariance",plot = FALSE)$acf ## USE AUTOCOVARIANCES
  AClen <- length(chainAC)
  gammaAC <- chainAC[1:(AClen-1)]+chainAC[2:AClen]

  m <- 1
  currgamma <- gammaAC[1]
  k <- 1
  while ((k<length(gammaAC)) && (gammaAC[k+1]>0) && (gammaAC[k]>=gammaAC[k+1]))
    k <- k +1

  if (k==length(gammaAC)) # added up until the very last computed autocovariance
    cat("WARNING: may need to compute more autocovariances for imse\n")
  sigmasq <- -chainAC[1]+2*sum(gammaAC[1:k])

  if (asymvar) # return asymptotic variance
    return(sigmasq)

  mcse <- sqrt(sigmasq/length(outp))
  return(mcse)
}

imsemat=function(mcmat,skip=NA)
{
  if (!is.na(skip))
    num=ncol(mcmat)-length(skip)
  else
    num=ncol(mcmat)

  imsevals=matrix(NA,num,2,dimnames=list(paste("V",seq(1,num),sep=""),c("est","se")))) # fi

  mcmat=mcmat[-skip] # remove columns to be skipped
  imseres=apply(mcmat,2,imse)
  for (i in 1:num)
  {
    imsevals[i,]=c(mean(mcmat[,i]),imseres[i])
  }
  return(imsevals)
}

## plot how Monte Carlo estimates change with increase in sample size
## input: samp (sample vector) and g (where E(g(x)) is quantity of interest)
## output: plot of estimate over time (increasing sample size)
## e.g.: estvssamp(outp,plotname=expression(paste("E(", beta, ")")))
estvssamp = function(samp, g=mean, plotname="mean estimates")
{
  if (length(samp)<100)

```

Page 442

```
batchsize = 1
else
  batchsize = length(samp)%%100

est = c()
for (i in seq(batchsize,length(samp),by=batchsize))
{
  est = c(est, g(samp[1:i]))
}

# plot(seq(batchsize,length(samp),by=batchsize),est,main=paste("M.C. estimates vs. sample size"))
plot(seq(batchsize,length(samp),by=batchsize),est,main=plotname,type="l",xlab="sample size")

## estimate effective sample size (ESS) as described in Kass et al (1998) and Robert and Casella (1999)
## ESS=size of an iid sample with the same variance as the current sample
## ESS = T/kappa where kappa (the 'autocorrelation time' for the sample) = 1 + 2 sum of all correlations beyond lag 1
## Here we use a version analogous to IMSE where we cut off correlations beyond a certain lag
ess = function(outp,imselags=TRUE)
{
  if (imselags) # truncate number of lags based on imse approach
  {
    chainACov <- acf(outp,type="covariance",plot = FALSE)$acf ## USE AUTOCOVARIANCES
    ACovlen <- length(chainACov)
    gammaACov <- chainACov[1:(ACovlen-1)]+chainACov[2:ACovlen]

    m <- 1
    currgamma <- gammaACov[1]
    k <- 1
    while ((k<length(gammaACov)) && (gammaACov[k+1]>0) && (gammaACov[k]>=gammaACov[k+1]))
      k <- k +1
    cat("truncated after ",k," lags\n")
    if (k==length(gammaACov)) # added up until the very last computed autocovariance
      cat("WARNING: may need to compute more autocovariances/autocorrelations for ess\n")

    chainACorr = acf(outp,type="correlation",plot = FALSE)$acf ## USE AUTOCORRELATIONS
    if (k==1)
      ACtime = 1
    else
      ACtime <- 1 + 2*sum(chainACorr[2:k]) # add autocorrelations up to lag determined
  }
  else
  {
    chainACorr = acf(outp,type="correlation",plot = FALSE)$acf ## USE AUTOCORRELATIONS
    ACtime <- 1 + 2*sum(chainACorr[-c(1)])
  }

  return(length(outp)/ACtime)
}

## effective samples per second
espersec = function(outp,numsec,imselags=TRUE)
{
  essval = ess(outp,imselags)
  return(essval/numsec)
}
```

Penn State Astrostatistics MCMC tutorial

Murali Haran, Penn State Dept. of Statistics

A Bayesian change point model

Consider the following hierarchical changepoint model for the number of occurrences Y_i of some event during time interval i with change point k .

$$\begin{aligned} Y_i|k, \theta, \lambda &\sim \text{Poisson}(\theta) \text{ for } i = 1, \dots, k \\ Y_i|k, \theta, \lambda &\sim \text{Poisson}(\lambda) \text{ for } i = k + 1, \dots, n \end{aligned}$$

Assume the following prior distributions:

$$\begin{aligned} \theta|b_1 &\sim \text{Gamma}(0.5, b_1) & (\text{pdf}=g_1(\theta|b_1)) \\ \lambda|b_2 &\sim \text{Gamma}(0.5, b_2) & (\text{pdf}=g_2(\lambda|b_2)) \\ b_1 &\sim \text{IG}(0, 1) & (\text{pdf}=h_1(b_1)) \\ b_2 &\sim \text{IG}(0, 1) & (\text{pdf}=h_2(b_2)) \\ k &\sim \text{Uniform}(1, \dots, n) & (\text{pmf}=u(k)) \end{aligned}$$

k, θ, λ are conditionally independent and b_1, b_2 are independent.

Assume the Gamma density parameterization $\text{Gamma}(\alpha, \beta) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} e^{-x/\beta}$ and IG (Inverse Gamma) density parameterization $\text{IG}(\alpha, \beta) = \frac{e^{-1/\beta x}}{\Gamma(\alpha)\beta^\alpha x^{\alpha+1}}$

Inference for this model is therefore based on the 5-dimensional **posterior** distribution $f(k, \theta, \lambda, b_1, b_2 | \mathbf{Y})$ where $\mathbf{Y} = (Y_1, \dots, Y_n)$. The posterior distribution is obtained *up to a constant* (that is, the normalizing constant is unknown) by taking the product of all the conditional distributions. Thus we have

$$\begin{aligned} f(k, \theta, \lambda, b_1, b_2 | \mathbf{Y}) &\propto \prod_{i=1}^k f_1(Y_i|\theta, \lambda, k) \prod_{i=k+1}^n f_2(Y_i|\theta, \lambda, k) \\ &\quad \times g_1(\theta|b_1)g_2(\lambda|b_2)h_1(b_1)h_2(b_2)u(k) \\ &= \prod_{i=1}^k \frac{\theta^{Y_i} e^{-\theta}}{Y_i!} \prod_{i=k+1}^n \frac{\lambda^{Y_i} e^{-\lambda}}{Y_i!} \\ &\quad \times \frac{1}{\Gamma(0.5)b_1^{0.5}} \theta^{-0.5} e^{-\theta/b_1} \times \frac{1}{\Gamma(0.5)b_2^{0.5}} \lambda^{-0.5} e^{-\lambda/b_2} \\ &\quad \times \frac{e^{-1/b_1}}{b_1} \frac{e^{-1/b_2}}{b_2} \frac{1}{n} \end{aligned}$$

Penn State Astrostatistics MCMC tutorial

Murali Haran, Penn State Dept. of Statistics

Bayesian change point model: full conditional distributions

Our goal is to draw samples from the 5-dimensional **posterior** distribution $f(k, \theta, \lambda, b_1, b_2 | \mathbf{Y})$. The posterior distribution is

$$\begin{aligned} f(k, \theta, \lambda, b_1, b_2 | \mathbf{Y}) &\propto \prod_{i=1}^k \frac{\theta^{Y_i} e^{-\theta}}{Y_i!} \prod_{i=k+1}^n \frac{\lambda^{Y_i} e^{-\lambda}}{Y_i!} \\ &\times \frac{1}{\Gamma(0.5) b_1^{0.5}} \theta^{-0.5} e^{-\theta/b_1} \times \frac{1}{\Gamma(0.5) b_2^{0.5}} \lambda^{-0.5} e^{-\lambda/b_2} \\ &\times \frac{e^{-1/b_1}}{b_1} \frac{e^{-1/b_2}}{b_2} \times \frac{1}{n} \end{aligned} \quad (1)$$

Note: The reason we have a formula for what f is proportional to (hence \propto rather than $=$) instead of an exact description of the function is because the missing constant (the normalizing constant) can only be computed by integrating the above function. Fortunately, the Metropolis-Hastings algorithm does not require knowledge of this normalizing constant.

From (1) we can obtain full conditional distributions for each parameter by ignoring all terms that are constant with respect to the parameter. Sometimes these full conditional distributions are well known distributions such as the Gamma or Normal.

Full conditional for θ :

$$\begin{aligned} f(\theta | k, \lambda, b_1, b_2, \mathbf{Y}) &\propto \prod_{i=1}^k \frac{\theta^{Y_i} e^{-\theta}}{Y_i!} \times \frac{1}{\Gamma(0.5) b_1^{0.5}} \theta^{-0.5} e^{-\theta/b_1} \\ &\propto \theta^{\sum_{i=1}^k Y_i - 0.5} e^{-\theta(k+1/b_1)} \\ &\propto \text{Gamma}\left(\sum_{i=1}^k Y_i + 0.5, \frac{b_1}{kb_1 + 1}\right) \end{aligned}$$

Full conditional for λ :

$$\begin{aligned} f(\lambda | k, \theta, b_1, b_2, \mathbf{Y}) &\propto \prod_{i=k+1}^n \frac{\lambda^{Y_i} e^{-\lambda}}{Y_i!} \times \frac{1}{\Gamma(0.5) b_2^{0.5}} \lambda^{-0.5} e^{-\lambda/b_2} \\ &\propto \text{Gamma}\left(\sum_{i=k+1}^n Y_i + 0.5, \frac{b_2}{(n-k)b_2 + 1}\right) \end{aligned}$$

Full conditional for k :

$$\begin{aligned} f(k|\theta, \lambda, b_1, b_2, \mathbf{Y}) &\propto \prod_{i=1}^k \frac{\theta^{Y_i} e^{-\theta}}{Y_i!} \prod_{i=k+1}^n \frac{\lambda^{Y_i} e^{-\lambda}}{Y_i!} \\ &\propto \theta^{\sum_{i=1}^k Y_i} \lambda^{\sum_{i=k+1}^n Y_i} e^{-k\theta - (n-k)\lambda}. \end{aligned}$$

Full conditional for b_1 :

$$f(b_1|k, \theta, \lambda, b_2, \mathbf{Y}) \propto \frac{1}{b_1^{0.5}} e^{-\theta/b_1} \times \frac{e^{-1/b_1}}{b_1} \propto b_1^{-1.5} e^{-(1+\theta)/b_1} \propto IG(0.5, 1/(\theta+1))$$

Full conditional for b_2 :

$$f(b_2|k, \theta, \lambda, b_1, \mathbf{Y}) \propto \frac{1}{b_2^{0.5}} e^{-\lambda/b_2} \times \frac{e^{-1/b_2}}{b_2} \propto b_2^{-1.5} e^{-(1+\lambda)/b_2} \propto IG(0.5, 1/(\lambda+1))$$

We are now in a position to run the Metropolis-Hastings algorithm.

Note 1: $\theta, \lambda, b_1, b_2$ all have full conditional distributions that are well known and easy to sample from. We can therefore perform Gibbs updates on them where the draw is from their full conditional. However, the full conditional for k is not a standard distribution so we need to use the more general Metropolis-Hastings update instead of a Gibbs update.

Note 2: The Inverse Gamma density is said to be a **conjugate** prior in this case since it results in a posterior that is also Inverse Gamma and therefore trivial to sample. As such, this density is mathematically convenient (due to its conjugacy property) but does not necessarily result in a better MCMC sampler. Also, it has poorly behaved moments; it may be better to adopt another prior density (such as a Gamma) instead.

The Metropolis-Hastings algorithm:

1. Pick a starting value for the Markov chain, say $(\theta^0, \lambda^0, k^0, b_1^0, b_2^0) = (1, 1, 20, 1, 1)$.
 2. ‘Update’ each variable in turn:
 - (a) Sample $\theta^i \sim f(\theta|k, \lambda, b_1, b_2, \mathbf{Y})$ using the most upto date values of k, λ, b_1, b_2 (Gibbs update using the derived Gamma density).
 - (b) Sample $\lambda^i \sim f(\lambda|k, \theta, b_1, b_2, \mathbf{Y})$ using the most upto date values of k, θ, b_1, b_2 . (Gibbs update using the derived Gamma density).
 - (c) Sample $b_1^i \sim f(b_1|k, \theta, \lambda, b_2, \mathbf{Y})$ using the most upto date values of k, θ, λ, b_2 . (Gibbs update using the derived Gamma density).
 - (d) Sample $b_2^i \sim f(b_2|k, \theta, \lambda, b_1, \mathbf{Y})$ using the most upto date values of k, θ, λ, b_1 . (Gibbs update using the derived Gamma density).
 - (e) Sample $k \sim f(k|\theta, \lambda, b_1, b_2, \mathbf{Y})$ using the most upto date values of $k, \theta, \lambda, b_1, b_2$. This requires a Metropolis-Hastings update:
 - i. ‘Propose’ a new value for k , k^* according to a proposal distribution say $q(k|\theta, \lambda, b_1, b_2, \mathbf{Y})$. In our simple example we pick $q(k|\theta, \lambda, b_1, b_2, \mathbf{Y}) = \text{Unif}\{2, \dots, m-1\}$ where m is the length of the vector (time series) \mathbf{Y} .
 - ii. Compute the Metropolis-Hastings accept-reject ratio,
$$\alpha(k, k^*) = \min \left(\frac{f(k^*|\theta, \lambda, b_1, b_2, \mathbf{Y})q(k|\theta, \lambda, b_1, b_2, \mathbf{Y})}{f(k|\theta, \lambda, b_1, b_2, \mathbf{Y})q(k^*|\theta, \lambda, b_1, b_2, \mathbf{Y})}, 1 \right)$$
 - iii. Accept the new value k^* with probability $\alpha(k, k^*)$, otherwise ‘reject’ k^* , i.e., the next value of k remains the same as before.
 - (f) You now have a new Markov chain state $(\theta^1, \lambda^1, k^1, b_1^1, b_2^1)$
3. Return to step #2 N-1 times to produce a Markov chain of length N .

Penn State Astrostatistics MCMC tutorial

Murali Haran, Penn State Dept. of Statistics

Bayesian change point model with Gamma hyperpriors

Consider the following hierarchical changepoint model for the number of occurrences Y_i of some event during time interval i with change point k .

$$\begin{aligned} Y_i | k, \theta, \lambda &\sim \text{Poisson}(\theta) \text{ for } i = 1, \dots, k \\ Y_i | k, \theta, \lambda &\sim \text{Poisson}(\lambda) \text{ for } i = k + 1, \dots, n \end{aligned}$$

Assume the following prior distributions:

$$\begin{aligned} \theta | b_1 &\sim \text{Gamma}(0.5, b_1) & (\text{pdf} = g_1(\theta | b_1)) \\ \lambda | b_2 &\sim \text{Gamma}(0.5, b_2) & (\text{pdf} = g_2(\lambda | b_2)) \\ b_1 &\sim \text{Gamma}(c_1, d_1) & (\text{pdf} = h_1(b_1)) \\ b_2 &\sim \text{Gamma}(c_2, d_2) & (\text{pdf} = h_2(b_2)) \\ k &\sim \text{Uniform}(1, \dots, n) & (\text{pmf} = u(k)) \end{aligned}$$

where $c_1 = c_2 = 0.01$ and $d_1 = d_2 = 100$, k, θ, λ are conditionally independent and b_1, b_2 are independent.

Assume the Gamma density parameterization $\text{Gamma}(\alpha, \beta) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} e^{-x/\beta}$

Inference for this model is therefore based on the 5-dimensional **posterior** distribution $f(k, \theta, \lambda, b_1, b_2 | \mathbf{Y})$ where $\mathbf{Y} = (Y_1, \dots, Y_n)$. The posterior distribution is obtained *upto a constant* by taking the product of all the conditional distributions. Thus we have

$$\begin{aligned} f(k, \theta, \lambda, b_1, b_2 | \mathbf{Y}) &\propto \prod_{i=1}^k f_1(Y_i | \theta, \lambda, k) \prod_{i=k+1}^n f_2(Y_i | \theta, \lambda, k) \\ &\quad \times g_1(\theta | b_1) g_2(\lambda | b_2) h_1(b_1) h_2(b_2) u(k) \\ &= \prod_{i=1}^k \frac{\theta^{Y_i} e^{-\theta}}{Y_i!} \prod_{i=k+1}^n \frac{\lambda^{Y_i} e^{-\lambda}}{Y_i!} \\ &\quad \times \frac{1}{\Gamma(0.5)b_1^{0.5}} \theta^{-0.5} e^{-\theta/b_1} \times \frac{1}{\Gamma(0.5)b_2^{0.5}} \lambda^{-0.5} e^{-\lambda/b_2} \\ &\quad \times \frac{1}{\Gamma(c_1)d_1^{c_1}} b_1^{c_1-1} e^{-b_1/d_1} \frac{1}{\Gamma(c_2)d_2^{c_2}} b_2^{c_2-1} e^{-b_2/d_2} \times \frac{1}{n} \end{aligned}$$

If we are able to draw samples from this distribution, we can answer questions of interest.

Penn State Astrostatistics MCMC tutorial

Murali Haran, Penn State Dept. of Statistics

Bayesian change point model with Gamma hyperpriors: full conditionals

Our goal is to draw samples from the 5-dimensional **posterior** distribution $f(k, \theta, \lambda, b_1, b_2 | \mathbf{Y})$. The posterior distribution is

$$\begin{aligned} f(k, \theta, \lambda, b_1, b_2 | \mathbf{Y}) &\propto \prod_{i=1}^k \frac{\theta^{Y_i} e^{-\theta}}{Y_i!} \prod_{i=k+1}^n \frac{\lambda^{Y_i} e^{-\lambda}}{Y_i!} \\ &\times \frac{1}{\Gamma(0.5)b_1^{0.5}} \theta^{-0.5} e^{-\theta/b_1} \times \frac{1}{\Gamma(0.5)b_2^{0.5}} \lambda^{-0.5} e^{-\lambda/b_2} \\ &\times \frac{1}{\Gamma(c_1)d_1^{c_1}} b_1^{c_1-1} e^{-b_1/d_1} \frac{1}{\Gamma(c_2)d_2^{c_2}} b_2^{c_2-1} e^{-b_2/d_2} \times \frac{1}{n} \end{aligned} \quad (1)$$

From 1 we can obtain full conditional distributions for each parameter by ignoring all terms that are constant with respect to the parameter.

For θ :

$$f(\theta | k, \lambda, b_1, b_2, \mathbf{Y}) \propto \prod_{i=1}^k \frac{\theta^{Y_i} e^{-\theta}}{Y_i!} \times \frac{1}{\Gamma(0.5)b_1^{0.5}} \theta^{-0.5} e^{-\theta/b_1} \quad (2)$$

For λ :

$$f(\lambda | k, \theta, b_1, b_2, \mathbf{Y}) \propto \prod_{i=k+1}^n \frac{\lambda^{Y_i} e^{-\lambda}}{Y_i!} \times \frac{1}{\Gamma(0.5)b_2^{0.5}} \lambda^{-0.5} e^{-\lambda/b_2} \quad (3)$$

For k :

$$f(k | \theta, \lambda, b_1, b_2, \mathbf{Y}) \propto \prod_{i=1}^k \frac{\theta^{Y_i} e^{-\theta}}{Y_i!} \prod_{i=k+1}^n \frac{\lambda^{Y_i} e^{-\lambda}}{Y_i!} \quad (4)$$

For b_1 :

$$f(b_1 | k, \theta, \lambda, b_2, \mathbf{Y}) \propto \frac{1}{b_1^{0.5}} e^{-\theta/b_1} \times b_1^{c_1-1} e^{-b_1/d_1} \quad (5)$$

For b_2 :

$$f(b_2 | k, \theta, \lambda, b_1, \mathbf{Y}) \propto \frac{1}{b_2^{0.5}} e^{-\lambda/b_2} \times b_2^{c_2-1} e^{-b_2/d_2} \quad (6)$$

$f(b_1 | k, \theta, \lambda, b_2, \mathbf{Y})$ and $f(b_2 | k, \theta, \lambda, b_1, \mathbf{Y})$ are not well known densities. We can use a Metropolis-Hastings accept-reject step to sample from their full conditionals.

Spatial Models: A *Quick* Overview

Astrostatistics Summer School, 2008

Murali Haran

Department of Statistics
Penn State University

1

Spatial Data

- Beginning statistics: Data are assumed to be independent and identically distributed ('i.i.d.') Inference is based on theory that relies on this assumption.
- *Spatial data* contain information about both the attribute of interest as well as its location.
- There is a need for more realistic models that account for the fact that data are spatially dependent. What's more, the dependence may be present in all directions and the relationships may be highly complex.
- Typical modeling assumption: observations that are further apart are less dependent than observations that are located close to each other.

2

The importance of dependence

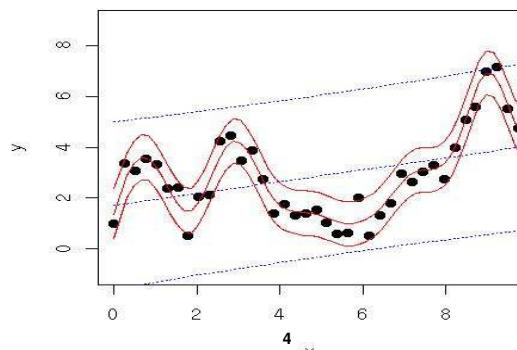
- Model will be a poor fit to the data, hence ignoring dependence can lead to poor estimates and poor prediction based on the estimated model.
- Not only do we have poor estimates and predictions, we will underestimate the variability of our estimates. (Variability of estimates is higher due to dependence.)
- Toy example: Consider the following simulated realization from a dependent process. For easy visualization, we consider a simple 1-D scenario:
 - Simulate $Y(s_i) = \beta s_i + \epsilon_i$ where $s_i \in (0, 1)$ and $i = 1, \dots, N$.
 - $(\epsilon_1, \dots, \epsilon_N)^T \sim \text{zero mean dependent process}$.

3

When ‘true model’ has dependent errors

Independent error model (blue, dotted): Poor fit though mean trend (β) is estimated reasonably well.

Dependent error model (red, solid), ϵ from a Gaussian process: much better fit.

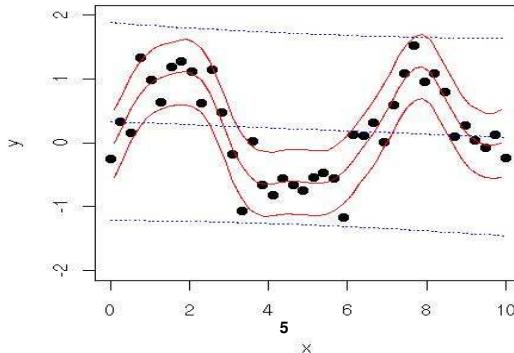


When ‘true model’ has complicated mean

Truth: $Y(s_i) = \sin(s_i) + \epsilon(s_i)$, with $\epsilon(s_i)$ independent.

Linear model with independent errors (blue, dotted): Poor fit.

Linear model with dependent (Gaussian process) errors (red, solid): much better fit *even though it is the ‘wrong’ model!*



Some reasons to use spatial models

- Can lead to superior estimators (e.g. low mean squared error).
- Spatial dependence can protect against misspecification of mean structure (hence, gaussian process are often used in machine learning, emulating output from complex computer models etc.)
- Statistically sound framework for interpolation.
- Ignoring dependence may underestimate variability.
- Sometimes learning about spatial dependence is of interest in its own right, e.g. finding clusters, regions of influence/dependence.

Useful ideas for non-spatial data

Although we will be talking about methods/models in the context of spatial data, some methods discussed here may be useful in non-spatial scenarios, for instance:

- Gaussian processes: Useful for modeling complex relationships of various kinds — particularly in machine learning (including classification), emulation of complex computer experiments (nonparametric curve fitting).
- Markov random fields: Time series, Graphical models, Semiparametric regression, Varying coefficient models etc.
- Notion of distance may arise in non-spatial data.

7

Some goals of spatial modeling

Scientists are often interested in one or more of the following goals:

- Modeling of trends and correlation structures, finding clusters.
- Estimation of the model parameters.
- Hypothesis Testing (or comparison of competing models).
- Prediction of observations at unobserved times or locations.
- Experimental design: Location of experimental units for optimal inference.

8

Types of Spatial Data

There are three main categories of spatial data (though it is not always obvious how to classify data into these categories):

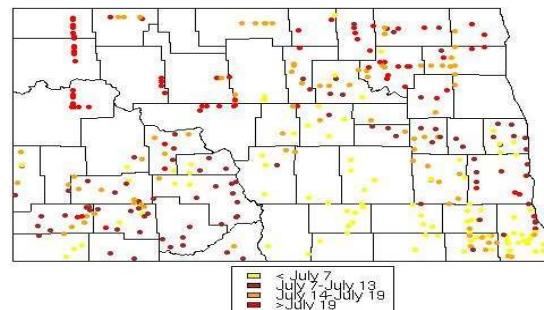
- Geostatistical data: When a spatial process that varies continuously is observed only at points.
- Lattice (areal) data: When a spatial process is observed at countably many (often finitely many) locations. Usually this arises due to aggregation of some sort, e.g. averages over a pixel.
- Spatial point processes: When a spatial process is observed at points and the locations themselves are of interest. Typical research questions are: Is the pattern random or does it exhibit clustering?

9

Geostatistical (point-referenced) data: Examples

(1) Concentrations of PM2.5 (pollutants) across the U.S.

(2) Wheat flowering dates by location (below):

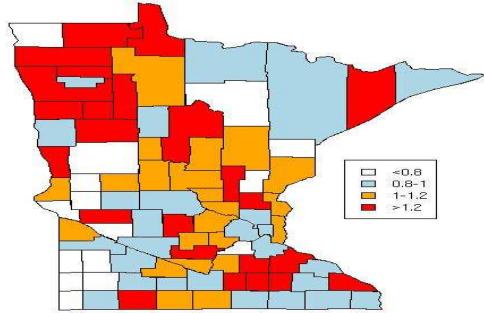


Courtesy Plant Pathology, PSU and North Dakota State.

10

Areal/Lattice Data: Examples

- (1) Pixel values from remote sensing e.g. forest cover in PA.
- (2) Event rates by county (e.g. below).



Courtesy MN Cancer Surveillance System, Dept. of Health

11

Spatial (linear) model for geostatistics and lattice data

Although geostatistical models and areal/lattice data models are usually talked about separately, they can be viewed in a unified framework.

- Spatial process at location \mathbf{s} is $Z(\mathbf{s}) = \mu(\mathbf{s}) + w(\mathbf{s})$ where:
 - $\mu(\mathbf{s})$ is the mean. Often $\mu(\mathbf{s}) = X(\mathbf{s})\beta$, $X(\mathbf{s})$ are covariates at \mathbf{s} and β is a vector of coefficients.
- Model dependence among spatial random variables by imposing it on the errors (the $w(\mathbf{s})$'s).
- For n locations, $\mathbf{s}_1, \dots, \mathbf{s}_n$, $\mathbf{w} = (w(\mathbf{s}_1), \dots, w(\mathbf{s}_n))^T$ can be jointly modeled via a zero mean Gaussian process (GP), for geostatistics, or Gaussian Markov random field (GMRF), for areal/lattice data.

12

Gaussian Processes

- Gaussian Process (GP): Let Θ be the parameters for covariance matrix $\Sigma(\Theta)$. Then:

$$\mathbf{w}|\Theta \sim N(0, \Sigma(\Theta)).$$

This implies:

$$\mathbf{Z}|\Theta, \beta \sim N(\mathbf{X}\beta, \Sigma(\Theta))$$

- We have used the simplest multivariate distribution (the multivariate normal). We will specify $\Sigma(\Theta)$ so it reflects spatial dependence.
- Need to ensure that $\Sigma(\Theta)$ is positive definite for this distribution to be valid, so we assume some valid parametric forms for specifying the covariance.

13

Gaussian Processes: Example

- Consider the popular **exponential** covariance function.
- Let $\Sigma(\Theta) = \kappa I + \psi H(\phi)$ where I is the $N \times N$ identity matrix. Note that $\Theta = (\kappa, \psi, \phi)$ and $\kappa, \psi, \phi > 0$.
- The i, j th element of the matrix H ,
$$H(\|\mathbf{s}_i - \mathbf{s}_j\|; \phi)_{ij} = \exp(-\phi \|\mathbf{s}_i - \mathbf{s}_j\|).$$
- Note: covariance between i, j th random variables depends only on distance between \mathbf{s}_i and \mathbf{s}_j , and does not depend on the locations themselves (implying *stationarity*) and only depends on the magnitude of the distance, not on direction (implying *isotropy*).
- Extremely flexible models, relaxing these conditions, can be easily obtained though fitting them can be more difficult.

14

Gaussian Processes: Inference

- The model completely specifies the likelihood, $\mathcal{L}(\mathbf{Z}|\Theta, \beta)$.
- This means we can do likelihood-based inference:
 - If we observe \mathbf{Z} , can find maximum likelihood estimates of Θ, β by maximizing $\mathcal{L}(\mathbf{Z}; \Theta, \beta)$ with respect to Θ, β .
 - Using the MLEs of Θ, β , and conditioning on the observed values \mathbf{Z} , we can easily estimate the value of this process at other locations ('kriging' with Gaussian processes.)
- If we place priors on Θ, β , we can do Bayesian inference:
 - Simulate from the posterior distribution, $\pi(\Theta, \beta | \mathbf{Z})$ via Markov chain Monte Carlo (tutorial tomorrow!)
 - Using sampled values of Θ, β , conditioning on \mathbf{Z} , can easily simulate value of this process at other locations.
 - Bayesian version incorporates variability due to uncertainty about Θ, β .

15

Gaussian Processes: Computing

- For likelihood based inference: R's `geoR` package by Ribeiro and Diggle.
- For Bayesian inference:
 - R's `spBayes` package by Finley, Banerjee and Carlin.
 - `WINBUGS` software by Spiegelhalter, Thomas and Best.
- Very flexible packages: can fit many versions of the linear Gaussian spatial model. Also reasonably well documented.
- Warning: With large datasets (>1000 data points), matrix operations (of order $O(N^3)$) become very slow. Either need to be clever with coding or modeling. Above software will not work.

16

Modeling areal/lattice data

- Recall that we are specifying dependence on the spatial data \mathbf{Z} via \mathbf{w} where $Z(\mathbf{s}) = \mu(\mathbf{s}) + w(\mathbf{s})$.
- We could stick to geostatistical modeling (using, say, distances between centroids of subregions.)
- This is often not reasonable. For e.g. centroids of subregions may lie outside the subregion.
- Modeling the process through its adjacencies may make more sense.
- Adjacent (neighboring) regions/pixels are thought to be more strongly related than those further away so, again, concerned with incorporating dependence into modeling.

17

Areal/lattice data: conditionally specified models

- Dependence in such cases can be imposed by a **conditionally specified modeled**. Idea is as follows:
What is the distribution of the random variable at this location *given* that I know the values of the random variable at neighboring locations?
- Caution: Need to ensure that conditional specification of distribution results in valid joint distribution. Theory relies on Hammersley-Clifford theorem and Gibbs distributions (cf. Besag, 1974).
- Best to use well studied conditionally specified models. For example, we could use Gaussian Markov random field models.

18

Areal/lattice data: conditionally specified models

- We model the conditional distribution of $Z(\mathbf{s}_i) | Z(\mathbf{s}_{-i})$ where $Z(\mathbf{s}_{-i})$ denotes all $Z(\mathbf{s}_j)$ except $Z(\mathbf{s}_i)$.
- Markov property: $Z(\mathbf{s}_i) | Z(\mathbf{s}_{-i})$ is the same as $Z(\mathbf{s}_i) | Z(\mathbf{s}_{j \sim i})$ where $j \sim i$ indicates that \mathbf{s}_j is a neighbor of \mathbf{s}_i . The distribution of $Z(\mathbf{s}_i)$ is *conditionally independent* of all the other values, *given* its neighboring values.
- This is therefore a local specification, although the model indirectly implies a global specification, i.e., a joint distribution (all the $Z(\mathbf{s})$'s will still be dependent on one another.)

19

Gaussian Markov random field (contd.)

- If we assume all conditional distributions are Normal (with appropriate conditions on variance parameters), resulting distribution is a multivariate normal.
- See any standard references on GMRFs for (slightly messy) details.
- If we let Θ be the parameters for the *precision matrix* $Q(\Theta)$. Then:

$$\mathbf{Z} | \Theta, \beta \sim N(\mathbf{X}\beta, Q^{-1}(\Theta))$$

- Since we have a likelihood, we can (as before) find an MLE or place priors on the parameters and do Bayesian inference and estimation.

20

Models for areal/lattice data: computing

- GMRF models (and more generally, conditionally specified models) have an important advantage: The matrices involved tend to be quite sparse and hence can yield considerable computational advantages over a Gaussian Process specification.
- GeoDa package at
<https://www.geoda.uiuc.edu/> (free) by Luc Anselin
- R's spdep package by Roger Bivand et al.
- Bayesian inference: WINBUGS includes GeoBUGS which is useful for fitting such models.

21

Spatial Point Processes: Introduction

Have so far discussed the first two major categories of spatial data. The third category is also equally important and perhaps of particular interest to astronomers.

- **Spatial point process:** The *locations* where the process is observed are random variables, process itself may not be defined; if defined, it is a **marked spatial point process**.
- **Observation window:** the area where points of the pattern can possibly be observed. The observation window specification is vitally important since absence of points in a region where they could potentially occur is also valuable information whereas absence of points outside of an observation window does not tell us anything.

22

Spatial Point Process: Example 1

Many problems can be formulated as spatial point process problems. Consider a study of tree species biodiversity (from Møller and Waagepetersen):

- Information available:
 - Locations of (potentially hundreds of thousands) of trees belonging to potentially thousands of species species.
 - Covariate information such as altitude, norm of altitude gradient etc.
- Some questions of interest:
 - Is the pattern completely random ?
 - If not completely random, can an explanatory point process model be fit to it?
 - How is the point pattern related to the covariates ?

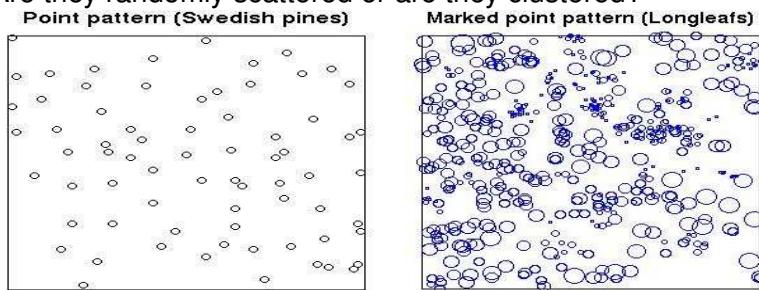
23

Spatial Point Process Data: Example 2

Locations of pine saplings in a Swedish forest.

Location and diameter of Longleaf pines (marked point process).

Are they randomly scattered or are they clustered?



(from Baddeley and Turner R package, 2006)

24

Questions related to spatial randomness of process

Some examples:

- Is there regular spacing between locations where process was observed or do locations show a tendency to cluster together? Need to fit clustering models and perhaps do some hypothesis testing.
- Does the probability of observing the event vary according to some factors? (Need to relate predictors to observations in a regression type setting.)
- Can we estimate the overall count from only partial observations? Need to fit a model to observations and make estimates/predictions based on fitted model.

25

Questions related to spatial randomness (contd)

Assume multiple (sometimes competing) models for the process. For instance, when studying point patterns of observations:

- Perhaps non-homogenous environmental conditions (associated with locations) are related to the presence/absence.
- Maybe the pattern arose by virtue of how the process spreads (e.g. clustering of 'offspring' near 'parents')?
- Note that **hypothesis testing alone is inadequate** for most of these questions. Can try to resolve these by fitting appropriate models where intensity of the process is modeled according to one of the models above.

26

Spatial Point Processes: Notes

- There appear to be many important problems where spatial point process modeling may be the most appropriate approach.
- However, the complexity of the theory along with computational difficulties have made it much less ‘friendly’ to applications than geostatistical models or areal models.
- Recent methodological developments and software such as the R library `spatstat` (A.Baddeley and Turner) are slowly opening up greater possibilities for practical modeling and analyses.

27

Classical Approaches

- Relatively small spatial point patterns.
- Assumption of stationarity is central and non-parametric methods based on summary statistics play a major role.
- Lack of software that works for classes of problems (software has been tailored to specific problems).
- In recent years, fast computing resources and better algorithms have allowed for analyses of larger point pattern data sets.

28

Some definitions for spatial point processes

- A spatial point process is a stochastic process, a realization of which consists of a countable set of points $\{\mathbf{s}_1, \dots, \mathbf{s}_n\}$ in a bounded region $S \in \mathbb{R}^2$
- The points \mathbf{s}_i are called **events**.
- For a region $A \in S$, $N(A) = \#\{\mathbf{s}_i \in A\}$.
- The **intensity measure** $\Lambda(A) = E(N(A))$ for any $A \in S$.
- If measure $\Lambda(A)$ has a density with respect to Lebesgue measure (we will typically assume this holds), then it can be written as:

$$\Lambda(A) = \int_A \lambda(\mathbf{s}) d\mathbf{s} \text{ for all } A \in S.$$

$\lambda(\mathbf{s})$ is called the **intensity function**.

29

Some definitions for spatial point processes (contd.)

- The process is **stationary** if for any integer k and regions A_i , $i = 1, \dots, k$, the joint distribution of $N(A_1), \dots, N(A_k)$ is translation-invariant, i.e., the joint distribution of $N(A_1), \dots, N(A_k)$ = joint distribution of $N(A_1 + \mathbf{y}), \dots, N(A_k + \mathbf{y})$ for arbitrary \mathbf{y} .
- The process is **isotropic** if for any integer k and regions A_i , $i = 1, \dots, k$, the joint distribution of $N(A_1), \dots, N(A_k)$ is invariant to rotation through an arbitrary angle, i.e., there is no directional effect.

30

Spatial point process modeling

Spatial point process models can be specified by :

- A deterministic intensity function (analogous to generalized linear model framework)
- A random intensity function (analogous to random effects models).
- Two classes of models:
 - Poisson Processes \approx provide models for no interaction patterns.
 - Cox processes \approx provide models for aggregated point patterns.
- Poisson process: Fundamental point process model — basis for exploratory tools and constructing more advanced point process models.

31

Homogeneous Poisson Process

Poisson process on \mathbf{X} defined on S with intensity measure Λ and intensity function λ , satisfies for any bounded region $B \in S$ with $\Lambda(B) > 0$:

- ① $N(B) \sim \text{Poisson}(\Lambda(B))$.
 - ② Conditional on $N(B)$, the points (event locations) $\mathbf{X}_B = \{X_1, \dots, X_{N(B)}\}$ in the bounded region are (i.i.d.) and each uniformly distributed in the region B .
- **Homogeneous Poisson process:** The intensity function, $\lambda(\mathbf{s})$ is constant for all $\mathbf{s} \in S$.
 - Poisson process is a model for complete spatial randomness since \mathbf{X}_A and \mathbf{X}_B are independent for all $A, B \in S$ that are disjoint.

32

Poisson Process (contd.)

- The intensity $\lambda(\mathbf{s})$ specifies the mean number of events per unit area as a function of location \mathbf{s} .
- Intensity is sometimes called the ‘density’ in other fields such as ecology (this term would be confused with a probability density, which is why it is not used in statistics).
- It is important as a null model and as a simple model from which to build other models.
- Homogeneous Poisson process is model for complete spatial randomness against which spatial point patterns are compared.

33

Poisson Process (contd)

Some notes:

- ① Stationarity $\Rightarrow \lambda(\mathbf{s})$ is constant $\Rightarrow \mathbf{X}$ is isotropic.
- ② **Random thinning** of a point process is obtained by deleting the events in series of mutually independent Bernoulli trials. Random thinning of Poisson process results in another Poisson process.
- Independence properties of Poisson process makes it unrealistic for most applications. However, it is mathematically tractable and hence easy to use/study.
- For modeling, usually consider log model of intensity function (to preserve non-negativity of intensity):

$$\log \lambda(\mathbf{s}) = z(\mathbf{s})\beta^T$$

34

Intensity of Poisson point process

Let $d\mathbf{s}$ denote a small region containing location \mathbf{s} .

- First-order intensity function of a spatial point process:

$$\lambda(\mathbf{s}) = \lim_{d\mathbf{s} \rightarrow 0} \frac{E(N(d\mathbf{s}))}{|d\mathbf{s}|}.$$

- Second-order intensity function of a spatial point process:

$$\lambda^{(2)}(\mathbf{s}_1, \mathbf{s}_2) = \lim_{d\mathbf{s}_1 \rightarrow 0} \lim_{d\mathbf{s}_2 \rightarrow 0} \frac{E\{N(d\mathbf{s}_1)N(d\mathbf{s}_2)\}}{|d\mathbf{s}_1||d\mathbf{s}_2|}.$$

- Covariance density of a spatial point process

$$\gamma(\mathbf{s}_1, \mathbf{s}_2) = \lambda^{(2)}(\mathbf{s}_1, \mathbf{s}_2) - \lambda(\mathbf{s}_1)\lambda(\mathbf{s}_2).$$

35

Intensity of Poisson point process (contd.)

Assuming stationarity and isotropy:

- Constant intensity: If $\mathbf{s} \in A$, $\lambda(\mathbf{s}) = \lambda = E(N(A))/|A|$, constant for all A .
- Second order intensity depends only on distance between locations $\mathbf{s}_1, \mathbf{s}_2$: $\lambda^{(2)}(\mathbf{s}_1, \mathbf{s}_2) = \lambda^{(2)}(\|\mathbf{s}_1 - \mathbf{s}_2\|)$.
- $\gamma(d) = \lambda^{(2)}(d) - \lambda^2$, where $d = \|\mathbf{s}_1 - \mathbf{s}_2\|$.

Hard to interpret $\lambda^{(2)}$. Instead, consider the *reduced second moment function*, the K-function:

$$K(d) = 2\pi \frac{1}{\lambda^2} \int_0^d \lambda^{(2)}(r) dr.$$

36

Intensity of Poisson point process (contd.)

Still assuming stationarity and isotropy:

$$K(d) = \frac{1}{\lambda} E(\text{number of events within distance } d \text{ of an arbitrary event}).$$

- Easier to interpret than second-order intensity and by dividing by λ , eliminate dependence on the intensity.
 - If process is clustered: Each event is likely to be surrounded by more events from the same cluster. $K(d)$ will therefore be *relatively large* for small values of d .
 - If process is randomly distributed in space: Each event is likely to be surrounded by empty space. For small values of d , $K(d)$ will be *relatively small*.
- Can obtain an intuitive estimator for $K(d)$ for a given data set.

37

Ripley's K Function

Let λ be the intensity of the process.

- Effective method for seeing whether the process is completely random in space.

$$K(d) = \frac{\text{Mean number of events within distance } d \text{ of an event}}{\lambda}$$

- This can be estimated by

$$\hat{K}(d) = \frac{\sum_{i \neq j} w_{ij} I(d_{ij} \leq d)}{\hat{\lambda}}$$

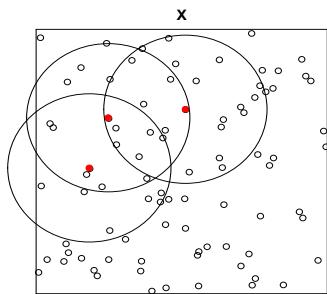
where $\hat{\lambda} = N/|A|$ with $|A|$ as the total area of the observation window and N is the observed count.

- Note: K can also be viewed as an integral of the two point correlation function as used by astronomers (cf. Martinez and Saar, 2002).

38

Estimating Ripley's K Function

Three circles of radius $d = 0.2$ each have been drawn with centers located at 3 locations where the process was observed. Note that they may overlap and also part of the circle may be outside the observation window. Circles are drawn for every point, number of points within each circle is counted.



39

Ripley's K Function (contd.)

- What are the weights (w_{ij} s) ?
- Just a way to account for edge effects: For events close to the edge of the observation window, we cannot observe the events within radius d .
- When we are estimating the $K(d)$ corresponding to a circle centered at location of an event at \mathbf{s}_i , and we are looking at an event at location \mathbf{s}_j , the weight w_{ij} is the reciprocal of the portion of the circle of radius d that is inside the region. If circle is completely contained in the region, w_{ij} is 1; the smaller the portion contained in the region, the larger the weight w_{ij} assigned (to 'correct' for the fact that the count was only for an area smaller than πd^2).

40

Ripley's K Function (contd.)

- Under complete spatial randomness (homogeneous spatial Poisson point process):

$$E(K(d)) = \pi d^2.$$
- Easy to see why (simple proof):
 - ① Location of events in a Poisson process are independent so occurrence of one event does not affect other events.
 - ② Since $E(\text{number of events in a unit area}) = \lambda$, $E(\text{number of events in area within radius } d) = \lambda \pi d^2$.
 - ③ $E(K(d)) = \frac{1}{\lambda} \lambda \pi d^2 = \pi d^2.$
- Once we have obtained $\hat{K}(d)$, we can plot $\hat{K}(d)$ versus d .
- Compare it to the plot we would have obtained under complete spatial randomness.

41

Inhomogeneous Poisson processes

Useful for modeling spatial process that varies in intensity over space. An inhomogeneous Poisson process with intensity λ satisfies:

- Number of events $N(A)$ in an observation window A is Poisson with mean

$$\Lambda(A) = \int_A \lambda(\mathbf{s}) d\mathbf{s},$$

equivalently, $P(N(A) = N) = \frac{1}{N!} e^{-\Lambda(A)} (\Lambda(A))^N$.

- Conditional on $N(A)$, event locations are independently sampled from a probability density function proportional to $\lambda(\mathbf{s})$.

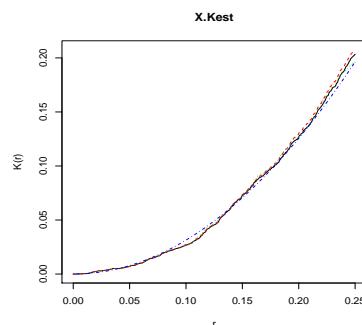
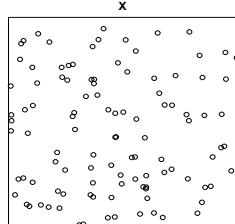
42

Ripley's K for homogeneous Poisson Process

Process was simulated with intensity function $\lambda(x, y) = 100$.

homogeneous Poisson Process

Ripley's K



blue=K function under complete spatial randomness

black (and red and green) are various versions of estimates of the K function

43

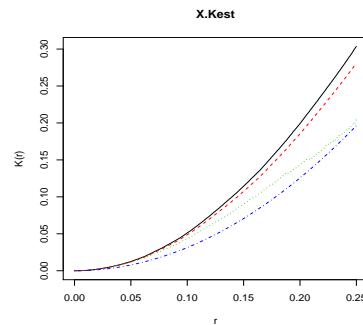
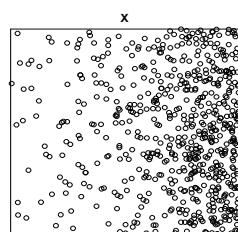
Ripley's K for inhomogeneous Poisson Process (Eg.1)

Process was simulated with intensity function

$\lambda(x, y) = 100 \exp(3x)$.

Inhomogeneous Poisson Process

Ripley's K



blue=K function under complete spatial randomness

black (and red and green) are various versions of estimates of the K function

44

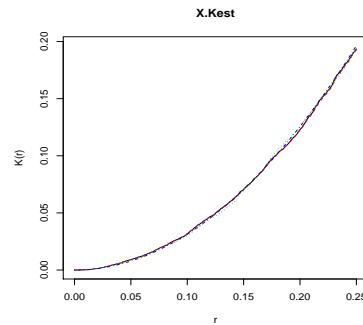
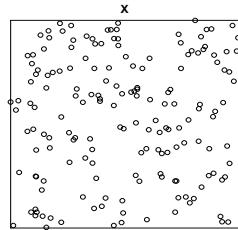
Ripley's K for inhomogeneous Poisson Process (Eq.2)

Process was simulated with intensity function

$$\lambda(x, y) = 100 \exp(y).$$

Inhomogeneous Poisson Process

Ripley's K



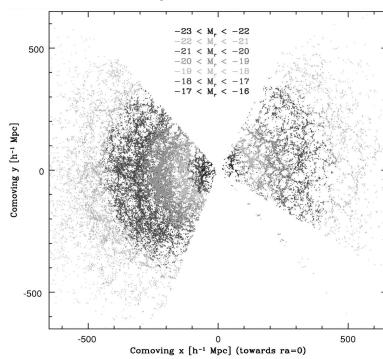
blue=K function under complete spatial randomness

black (and red and green) are various versions of estimates of
the K function

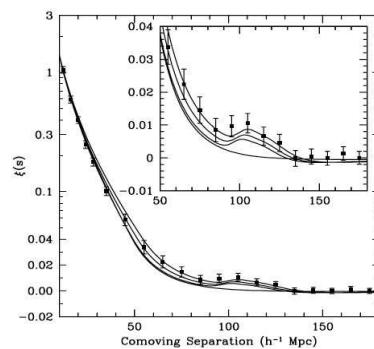
45

Example: Galaxy clustering (Sloan Digital Sky Survey)

Galaxy distribution



Two-point correlation function



Distribution of 67,676 galaxies in two slices of the sky showing strong anisotropic clustering (Tegmark et al. 2004).

Bottom: Two-point correlation function showing the faint feature around 100 megaparsec scales revealing cosmological Baryonic Acoustic Oscillations (Eisenstein et al. 2005).

Ripley's K function: transformation

- As can be seen from Eg.2, even for strong departures from complete spatial randomness, difference between Ripley's K and its expectation under complete spatial randomness can be small.
- Plot of K function may not suffice. Instead, consider a linearizing transformation:

$$L(d) = \sqrt{K(d)/\pi} - d.$$

- Complete spatial randomness: $E(L(d)) = 0$.
- Clustering: $E(L(d)) > 0$.
- Regular spacing: $E(L(d)) < 0$.

47

Ripley's K function: robustness to thinning

- As pointed out before, random thinning of a Poisson process results in a Poisson process.
- Also, random thinning reduces the intensity and the number of events within a distance d of a location by the same multiplicative factor.
- Since $K(d)$ is the ratio of the number of events within a distance d and the intensity of the process, it is robust to incomplete ascertainment (random thinning).
- Hence, $K(d)$ does not change as long as missing cases are missing at random (missingness does not depend on location).

48

Inference for Poisson process

We would like to be able to perform statistical inference for a point process. By definition, Poisson process on \mathbf{X} defined on S with intensity measure Λ and intensity function λ satisfies for any bounded region $B \in S$ with $\Lambda(B) > 0$:

- $N(B) \sim \text{Poisson}(\Lambda(B))$, i.e.

$$f(N(B)|\Lambda(B)) = \frac{\exp(-\Lambda(B))\Lambda(B)^{N(B)}}{N(B)!}$$

- Conditional on $N(B)$, the points (event locations)

$\mathbf{X}_B = \{X_1, \dots, X_{N(B)}\}$ in the bounded region are (i.i.d.) and each uniformly distributed in the region B :

$$f(X_1, \dots, X_{N(B)}|N(B)) = \prod_{i=1}^{N(B)} f(X_i|N(B)) = \prod_{i=1}^{N(B)} \frac{\lambda(X_i)}{\int_B \lambda(\mathbf{s}) d\mathbf{s}}$$

49

Inference for Poisson process (contd.)

- The joint distribution is then:

$$\begin{aligned} f(X_1, \dots, X_{N(B)}, N(B)) &= \frac{\exp(-\Lambda(B))\Lambda(B)^{N(B)}}{N(B)!} \prod_{i=1}^{N(B)} \frac{\lambda(X_i)}{\int_B \lambda(\mathbf{s}) d\mathbf{s}} \\ &= \frac{\exp(-\Lambda(B))\Lambda(B)^{N(B)}}{N(B)!} \prod_{i=1}^{N(B)} \frac{\lambda(X_i)}{\Lambda(B)} = \frac{\exp(-\Lambda(B))}{N(B)!} \prod_{i=1}^{N(B)} \lambda(X_i). \end{aligned}$$

- For instance, this means that for a region $F \in S$ and a point process \mathbf{X} :

$$P(\mathbf{X} \in F, N = n) = \int_S \mathbf{1}(\mathbf{X} \in F) \frac{\exp(-\Lambda(B))}{n!} \prod_{i=1}^n \lambda(X_i) d\mathbf{X}$$

and $P(\mathbf{X} \in F) = \sum_{n=0}^{\infty} P(\mathbf{X} \in F, N = n)$.

50

Space-varying covariates: modulated Poisson process

- As before, denote covariates at a location \mathbf{s} by $\mathbf{X}(\mathbf{s})$.
Impact of spatially varying covariates on a spatial point pattern may be modeled through the intensity function

$$\lambda(\mathbf{s}) = \exp(\beta X(\mathbf{s}))$$
- Inhomogeneous Poisson process with this intensity is a *modulated Poisson process*.
- Examples of $\mathbf{X}(\mathbf{s})$: spatially varying environmental variables such as elevation, precipitation etc., known functions of the spatial coordinates or distances to known environmental features (e.g. distance to nearest road).
- Important question: How is \mathbf{X} related to the spatial point process intensity, i.e., what is β ?
51

Parameter estimation for modulated Poisson process

Maximum likelihood estimation using observed \mathbf{X} on a region S :

- The likelihood for the simple linear model is (from before):

$$\mathcal{L}(\mathbf{X}, N; \beta) = \frac{\exp(\Lambda(S))}{N(B)!} \prod_{i=1}^{N(B)} \lambda(X_i).$$

$$\mathcal{L}(\mathbf{X}, N; \beta) = \frac{\exp(-\int_S \exp(\mathbf{X}(\mathbf{s})\beta))}{N(B)!} \prod_{i=1}^{N(B)} \exp(\beta X_i).$$

- MLE for β : Find $\hat{\beta}$ that maximizes likelihood. This may be difficult, need to use Newton-Raphson or other optimization algorithm.
- Note that an assumption above is that covariates are available everywhere.

Modulated Poisson process with missing covariates

- Impractical to assume covariates are observed for every observed event and all locations in observation window.
- Need to turn to other approaches. Natural approach is to estimate covariate information based on observed covariate information (cf. Rathbun, 1996).
- Use kriging (a form of spatial interpolation, falling under ‘Geostatistics’) to predict the values of the covariates at locations of observed events and at unsampled locations.
- Substitute predicted values of the covariates into the likelihood.
- Maximize this approximate likelihood to obtain coefficient estimates, $\tilde{\beta}$.

53

Cox Process

The Cox process or the *doubly stochastic Poisson process* (Cox, 1955) is a more flexible and realistic class of models than the Poisson process model.

- Natural extension of a Poisson process: Consider the intensity function of the Poisson process as a realization of a random field. We assume $\Lambda(A) = \int_A \lambda(\mathbf{s}) d\mathbf{s}$.
 - Stage 1: $N(A)|\Lambda \sim \text{Poisson}(\Lambda(A))$.
 - Stage 2: $\lambda(\mathbf{s})|\Theta \sim f(\cdot; \Theta)$ so that λ is stochastic, a nonnegative random field parametrized by Θ .
- Simple case: If $\lambda(\mathbf{s})$ is deterministic, \mathbf{X} is a Poisson process with intensity $\lambda(\mathbf{s})$.

54

Markov Point Processes

- Point patterns may require a flexible description that allows for the points to interact.
- Markov point processes are models for point processes with interacting points (attractive or repulsive behavior can be modeled).
- ‘Markovian’ in that intensity of an event at some location \mathbf{s} , given the realization of the process in the remainder of the region, depends only on information about events within some distance of \mathbf{s} .
- Origins in statistical physics, used for modeling large interacting particle systems.

55

Inference for spatial point process models

- Maximum likelihood for all but the simplest spatial point process model is analytically intractable. Maximum pseudolikelihood (MPL) is a useful approximation to maximum likelihood.
- For some models, can use Newton-Raphson or some variant but often need (Markov chain) Monte Carlo maximum likelihood (MCML), also referred to as simulated maximum likelihood (SML).
- No ‘automatic’ methods exist for fitting such models.
- Simulating from a point process model is often easy but inference (estimating a point process model based on observations) is usually more difficult. Challenging to fit flexible new models.

56

Spatial point processes: computing

- **R command:** `spatstat` function `ppm` fits models that include spatial trend, interpoint interaction, and dependence on covariates, generally using MPL.
- MPL often works well in practice (Baddeley, 2005). Caveat: MPL can work very poorly in some cases, particularly when there is strong dependence.
- MPLE can be used to get a guess for MLE before doing something more elaborate like Markov chain Maximum Likelihood (cf. C.J.Geyer's chapter in "MCMC in Practice", 1996 for a gentle introduction.)
- There is not much in the way of computing resources for fitting Bayesian models, even though they are becoming increasingly common.

57

Summary: spatial data types and associated models

General spatial process: $\{Z(\mathbf{s}) : \mathbf{s} \in D\}$, D is set of locations.

- **Geostatistics:** D is a fixed subset of \mathbb{R}^2 (or \mathbb{R}^3 in 3D case).
 $Z(\mathbf{s})$ is a random variable at each location $\mathbf{s} \in D$.
 Usual (basic) model: [Gaussian process](#).
- **Areal/lattice data:** $D = \{\mathbf{s}_1, \dots, \mathbf{s}_N\}$ is a fixed regular or irregular lattice, on \mathbb{R}^2 (or \mathbb{R}^3).
 $Z(\mathbf{s})$ is a random variable at each location $\mathbf{s} \in D$.
 Usual (basic) model: [Gaussian Markov random field](#).
- **Spatial point process:** $D = \{\mathbf{s}_1, \dots, \mathbf{s}_N\}$ is a random collection of points on the plane.
 Ordinary point process: $Z(\mathbf{s})$ does not exist. For marked point process, $Z(\mathbf{s})$ is a random variable as well. Usual (basic) models: [Poisson process](#), [Cox process](#).

58

References: Geostatistics and Lattice Processes

Geostatistics and Lattice/Areal Data:

- Schabenberger and Gotway (2005) "Statistical Methods for Spatial Data Analysis". A fairly comprehensive easy to read book on spatial models for (in order of emphasis): geostatistics, lattice data and point processes.
- Cressie (1994) "Statistics for Spatial Data". This is a comprehensive guide to classical spatial statistics, but it is considerably more technical than the other two references listed here.
- S. Banerjee, B.P. Carlin and A.E. Gelfand "Hierarchical Modeling and Analysis for Spatial Data". This is a textbook on Bayesian models for spatial data.

59

References: Spatial Point Processes

Spatial Point Processes:

- Møller and Waagepetersen's monograph "Modern statistics for spatial point processes" (2007) To appear in Scandinavian Journal of Statistics.
- Baddeley and Turner's R `spatstat` package.
- Baddeley et al. "Case Studies in Spatial Point Process Modeling" (2005).
- P.J.Diggle's online lecture notes:
[http://www.maths.lancs.ac.uk/~diggle
/spatial/epi/notes.ps](http://www.maths.lancs.ac.uk/~diggle/spatial/epi/notes.ps)

60

References: Spatial Point Processes

- P.J.Diggle "Statistical Analysis of Spatial Point Patterns" (2003)
- "Modern statistics for spatial point processes" by J.Møller and R.P.Waagepeterson (2004).
- V.J.Martinez and E.Sarr "Statistics of the Galaxy Distribution."

Notes about the references:

- ① Several of these references also cover spatiotemporal (space-time) process, that may also be of significant interest.
- ② Acknowledgement: A lot of the material and examples in this tutorial were drawn from several of the listed references.

Time Series Analysis

Eric Feigelson

Penn State University

<http://astrostatistics.psu.edu>

Outline

- ① Time series in astronomy
- ② Time domain methods: Nonparametric
- ③ 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

Poisson processes:

Individual photon/particle events in high-energy
astronomy

Time domain methods: Nonparametric

Autocorrelation function

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)} \quad \text{where} \quad \hat{\gamma}(h) = \frac{\sum_{i=1}^{n-h} (x_{t+h} - \bar{x})(x_t - \bar{x})}{n}$$

This sample ACF is an estimator of the correlation between the x_t and x_{t-h} in an evenly-spaced time series lags.

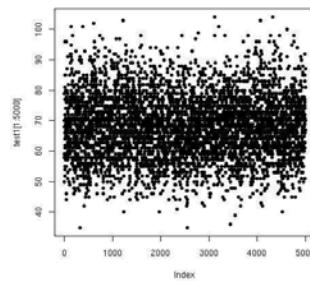
The partial autocorrelation function (PACF) estimates the correlation with the linear effect of the intermediate observations, $x_{t-1}, \dots, x_{t-h+1}$, removed. Calculate with the Durbin-Levinson algorithm based on an autoregressive model.

Note that the error on the mean, or any other parameter, of an autocorrelated time series is different from the usual value:

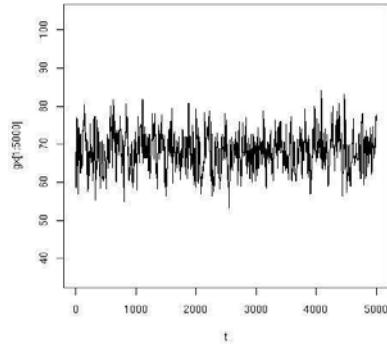
$$Var(\bar{x}) = \frac{\sigma^2}{N} [1 + \sum_{i=1}^{N-1} (1 - \frac{i}{N})\rho(i)]$$

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' (QPOs) from inhomogeneities in the disk and/or beating between the neutron star rotation and disk orbits. We plot below the first 5000 of 65,536 count rates from Ginga satellite observations. Superficially, it looks like white noise.

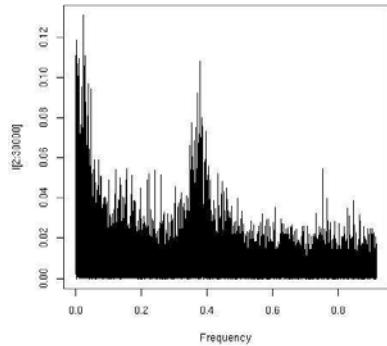


Nonparametric estimation: Kernel smoothing
Normal kernel, bandwidth = 7 bins



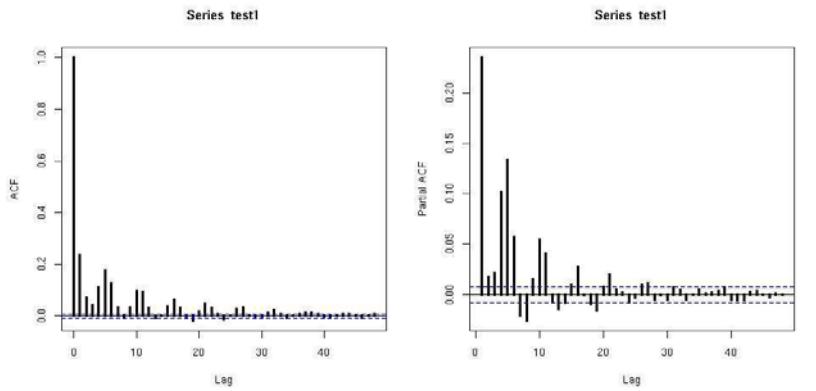
Smoothing is not very useful in this case, but it does reveal some correlated behavior.

Fourier transform (FFT)



The Fourier power spectrum reveals three components: power at low frequency (1/f-type 'red noise'), the QPO around freq=0.4, and white noise. Most astronomical studies of XRB time series and the QPO phenomenon are based on FFT analysis.

Autocorrelation functions



`acf(GX, lwd=3)`

`pacf(GX, lwd=3)`

The ACF and PACF provide quantitative measure of the short-term correlation, and show the periodic behavior.

Time domain models: ARMA models

Autoregressive moving average model

Very common model in human and engineering sciences, designed for stationary, Gaussian processes. Easily fit by maximum-likelihood. Disadvantage: parameter values are difficult to interpret physically.

AR(p) model $x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + w_t$

MA(q) model $x_t = w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q}$

(Note: model diverges if any $|\theta_i| > 1$)

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. ARIMA (I = integrated) models some types of non-stationary behaviors.

Time domain models: 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 $w_t = N(\nu, \tau)$,

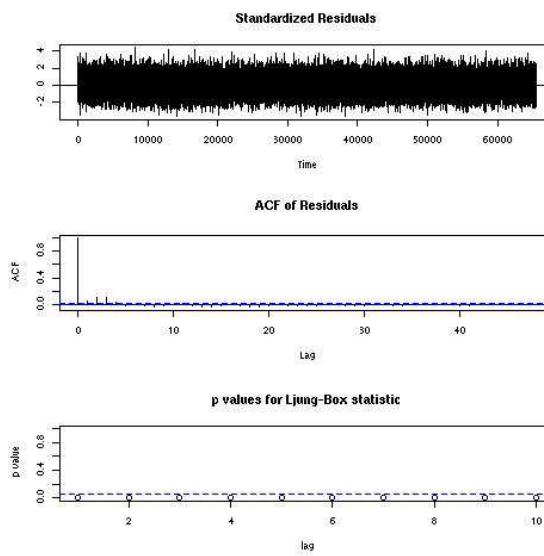
$$y_t = Ax_t + v_t \quad x_t = \phi_1 x_{t-1} + w_t$$

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 can be fit by maximum likelihood methods, or a Bayesian framework by assuming priors for the parameters. The likelihood function is easily calculated and updated via Kalman filtering.

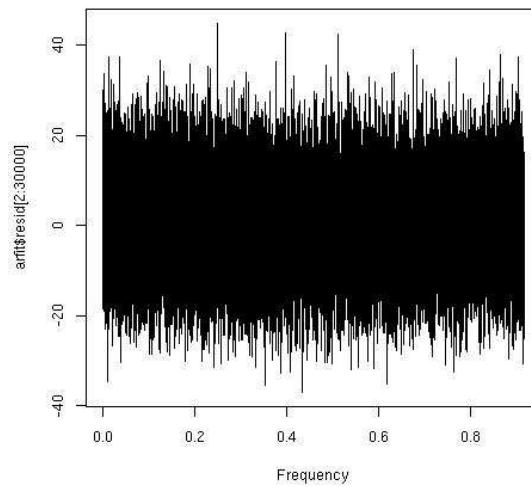
GX 5+1 autoregressive modeling

```
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      σ² = 102      log L = -244446.5
AIC = 488911.1 (use AIC for model selection)
```



Although the scatter is reduced by a factor of 30, the chosen model is not adequate: the model is divergent and the Ljung-Box test shows significant correlation in the residuals.



Nonetheless, the FFT power spectrum of the ARIMA residuals shows that most of the red noise and QSO structure is removed by the model.

Other time domain models

- Extended ARMA models: VAR (vector autoregressive), SARIMA (S = seasonal for periodic behavior), ARFIMA (F = fractional for long-memory behavior), 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

References

C. Chatfield, *The Analysis of Time Series: An Introduction*, 6th Ed., 2003

R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications (with R examples)*, 2nd Ed., 2006

G. Kitagawa & W. Gersch, *Smoothness Priors Analysis of Time Series*, 1996

J. K. Lindsey, *Statistical Analysis of Stochastic Processes in Time*, 2004

S. M. Ross, *Stochastic Processes*, 2nd ed, 1996

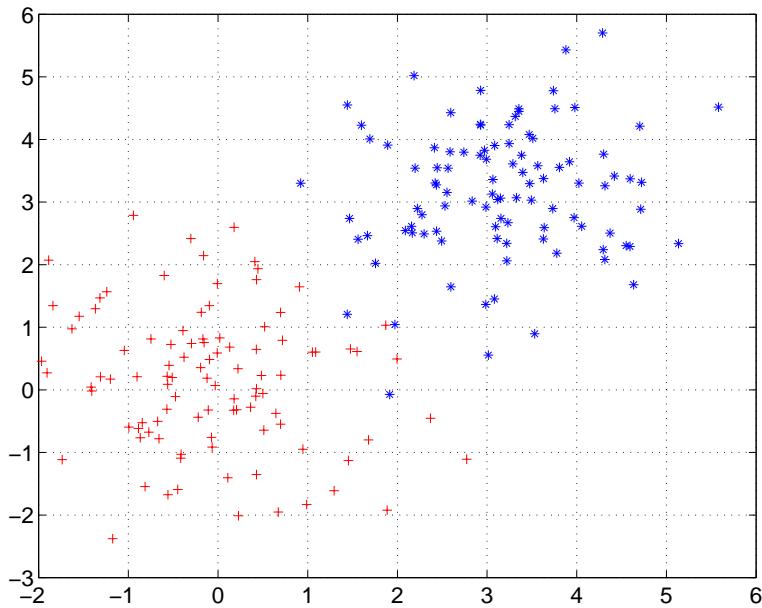
Cluster Analysis

Jia Li
Department of Statistics
Penn State University

Summer School in Statistics for Astronomers IV
June 9-14, 2008

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.

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

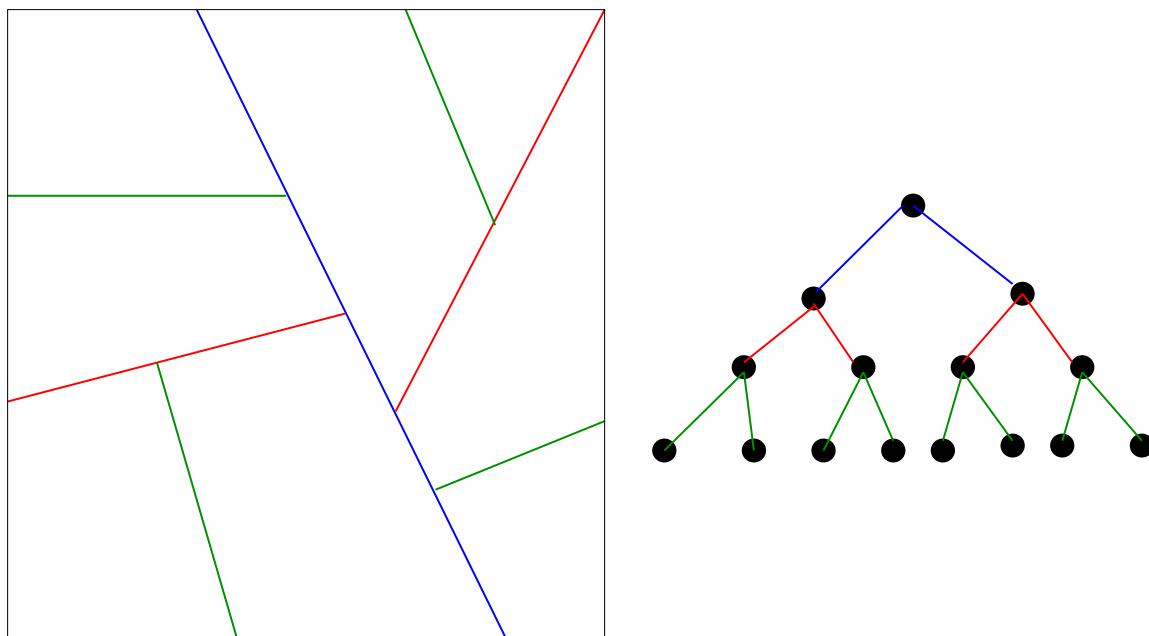
- Randomly pick up the prototypes to start the k-means iteration.
- Different initial prototypes may lead to different local optimal solutions given by k-means.
- Try different sets of initial prototypes, compare the objective function at the end to choose the best solution.
- When randomly select initial 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 ϵ has a small norm and a random direction.
 4. $k \leftarrow k + \tilde{k}$; $l \leftarrow l + 1$.
 5. Use $\{z_1^{(l)}, z_2^{(l)}, \dots, z_{\tilde{k}}^{(l)}\}$ 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.

K-center Clustering

- Let A be a set of n objects.
- Partition A into K sets C_1, C_2, \dots, C_K .
- *Cluster size of C_k* : the least value D for which all points in C_k are:
 1. within distance D of each other, or
 2. within distance $D/2$ of some point called the cluster center.
- Let the cluster size of C_k be D_k .
- The *cluster size* of partition S is

$$D = \max_{k=1,\dots,K} D_k .$$

- Goal: Given K , $\min_S D(S)$.

Comparison with k-means

- Assume the distance between vectors is the squared Euclidean distance.
- K-means:

$$\min_S \sum_{k=1}^K \sum_{i:x_i \in C_k} (x_i - \mu_k)^T (x_i - \mu_k)$$

where μ_k is the centroid for cluster C_k . In particular,

$$\mu_k = \frac{1}{N_k} \sum_{i:x_i \in C_k} x_i .$$

- K-center:

$$\min_S \max_{k=1,\dots,K} \max_{i:x_i \in C_k} (x_i - \mu_k)^T (x_i - \mu_k) .$$

where μ_k is called the “centroid”, but may not be the mean vector.

- Another formulation of k-center:

$$\min_S \max_{k=1,\dots,K} \max_{i,j:x_i,x_j \in C_k} L(x_i, x_j) .$$

$L(x_i, x_j)$ denotes any distance between a pair of objects.

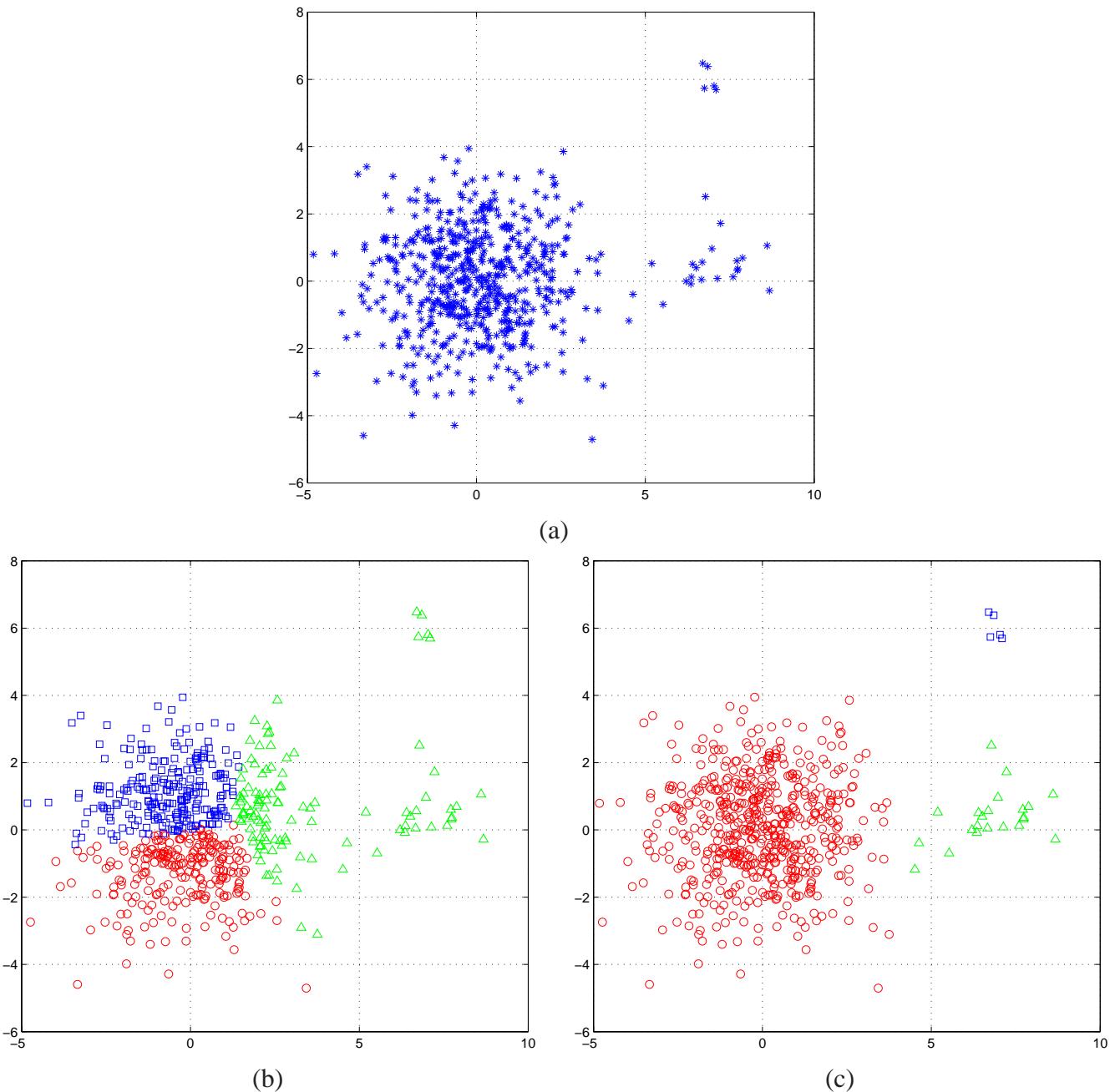


Figure 1: Comparison of k-means and k-center. (a): Original unclustered data. (b): Clustering by k-means. (c): Clustering by k-center. K-means focuses on average distance. K-center focuses on worst scenario.

Greedy Algorithm

- Choose a subset H from S consisting K points that are farthest apart from each other.
- Each point $h_k \in H$ represents one cluster C_k .
- Point x_i is partitioned into cluster C_k if

$$L(x_i, h_k) = \min_{k'=1, \dots, K} L(x_i, h_{k'}) .$$

- Only need pairwise distance $L(x_i, x_j)$ for any $x_i, x_j \in S$. Hence, x_i can be a non-vector representation of the objects.
- The greedy algorithm achieves an approximation factor of 2 as long as the distance measure L satisfies the triangle inequality. That is, if

$$D^* = \min_S \max_{k=1, \dots, K} \max_{i,j: x_i, x_j \in C_k} L(x_i, x_j)$$

then the greedy algorithm guarantees that

$$D \leq 2D^* .$$

- The relation holds if the cluster size is defined in the sense of centralized clustering.

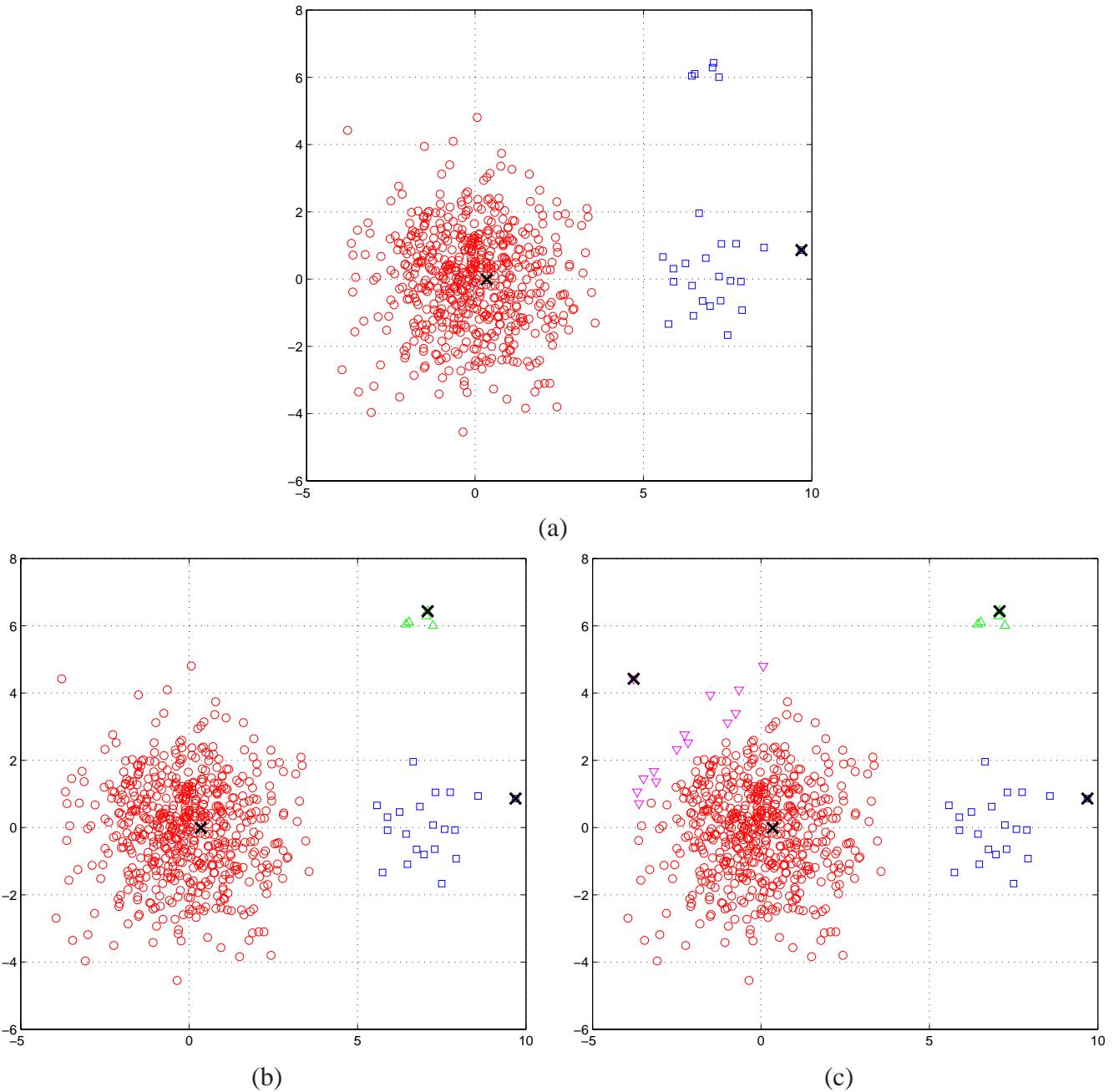


Figure 2: K-center clustering step by step. (a)-(c): 2-4 clusters.

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.

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:

$$\begin{aligned} D(t, k) = & \frac{n_r}{n_r + n_s} D(r, k) + \frac{n_s}{n_r + n_s} D(s, k) \\ & - \frac{n_r n_s}{n_r + n_s} D(r, s) \end{aligned}$$

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:*

$$\begin{aligned} D(t, k) = & \frac{n_r + n_k}{n_r + n_s + n_k} D(r, k) \\ & + \frac{n_s + n_k}{n_r + n_s + n_k} D(s, k) \\ & - \frac{n_k}{n_r + n_s + n_k} D(r, s) \end{aligned}$$

Merge the two clusters for which the change in the variance of the clustering is minimized. The vari-

ance 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.

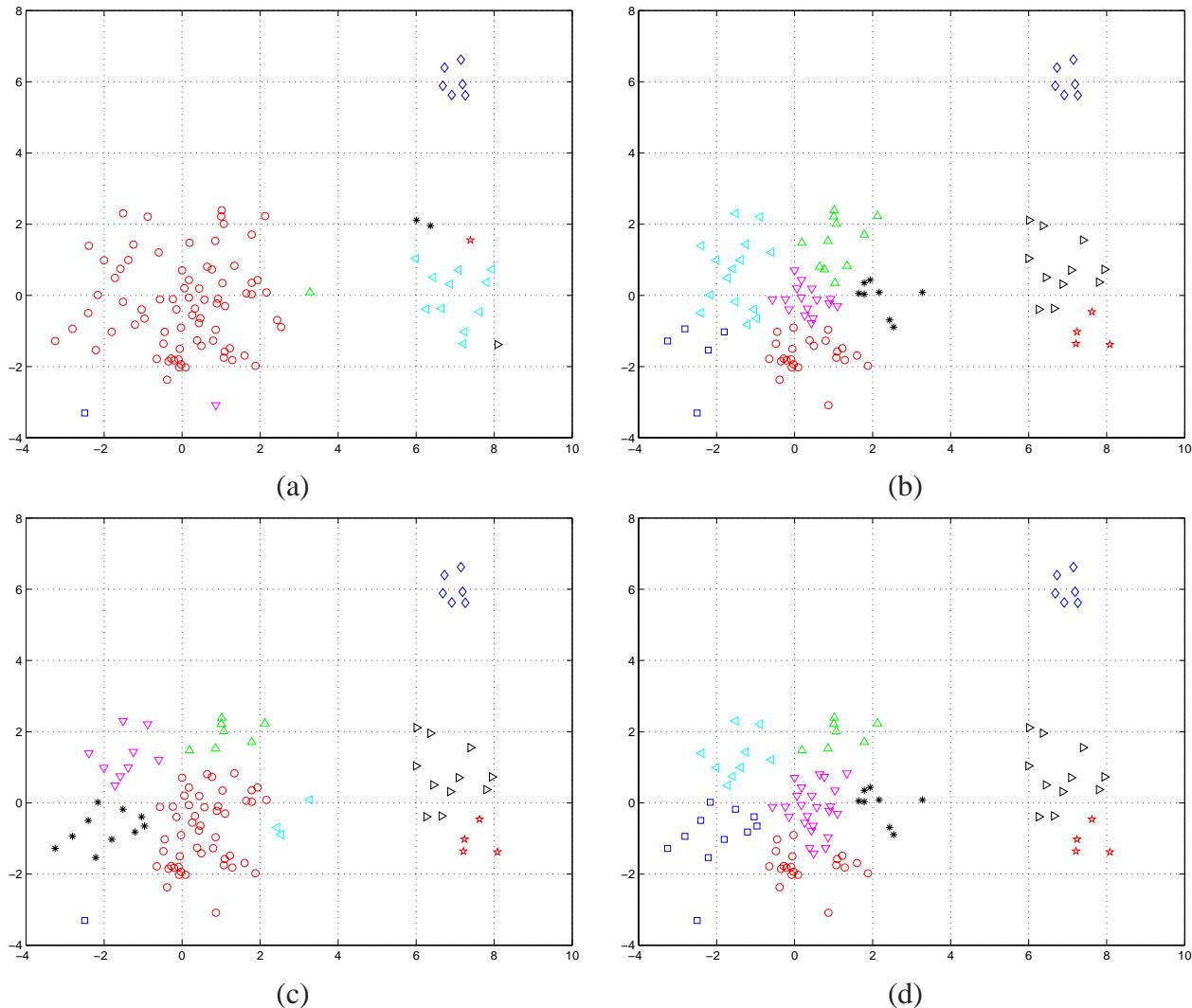


Figure 3: Agglomerate clustering of a data set (100 points) into 9 clusters. (a): Single-link, (b): Complete-link, (c): Average linkage, (d) Wards clustering

Hipparcos Data

- Clustering based on $\log L$ and BV .

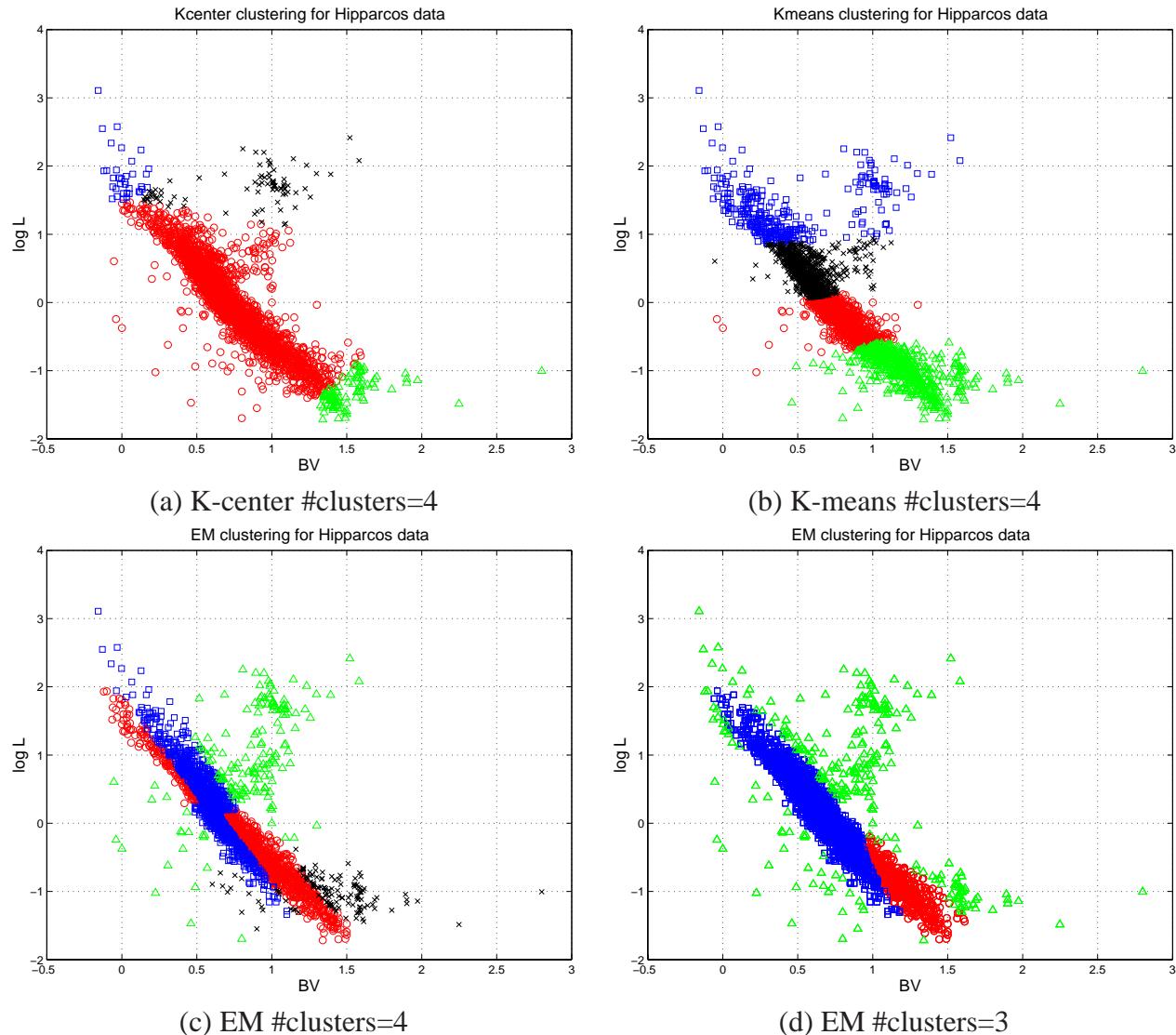


Figure 4: Clustering of the Hipparcos data

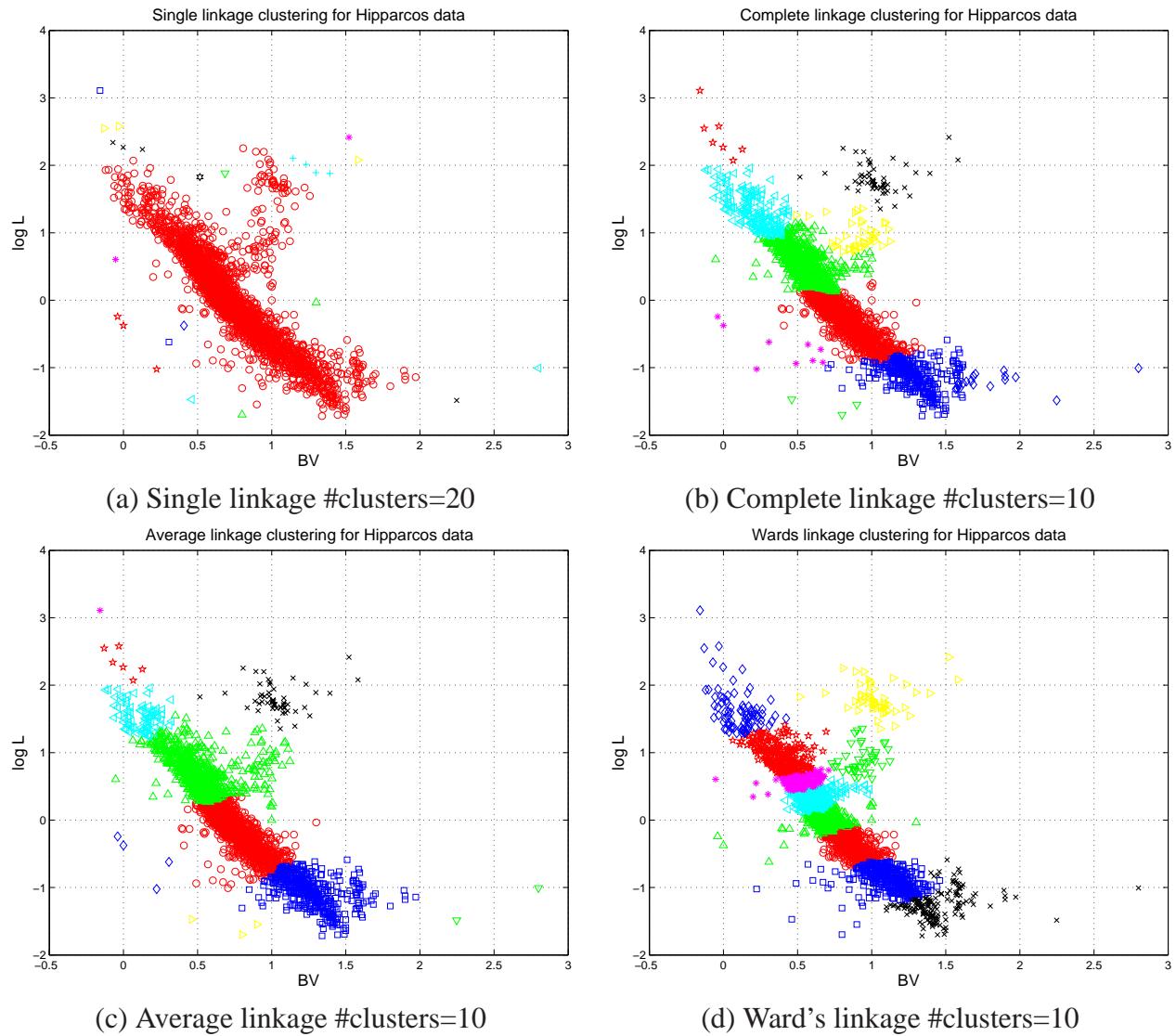


Figure 5: Clustering of the Hipparcos data

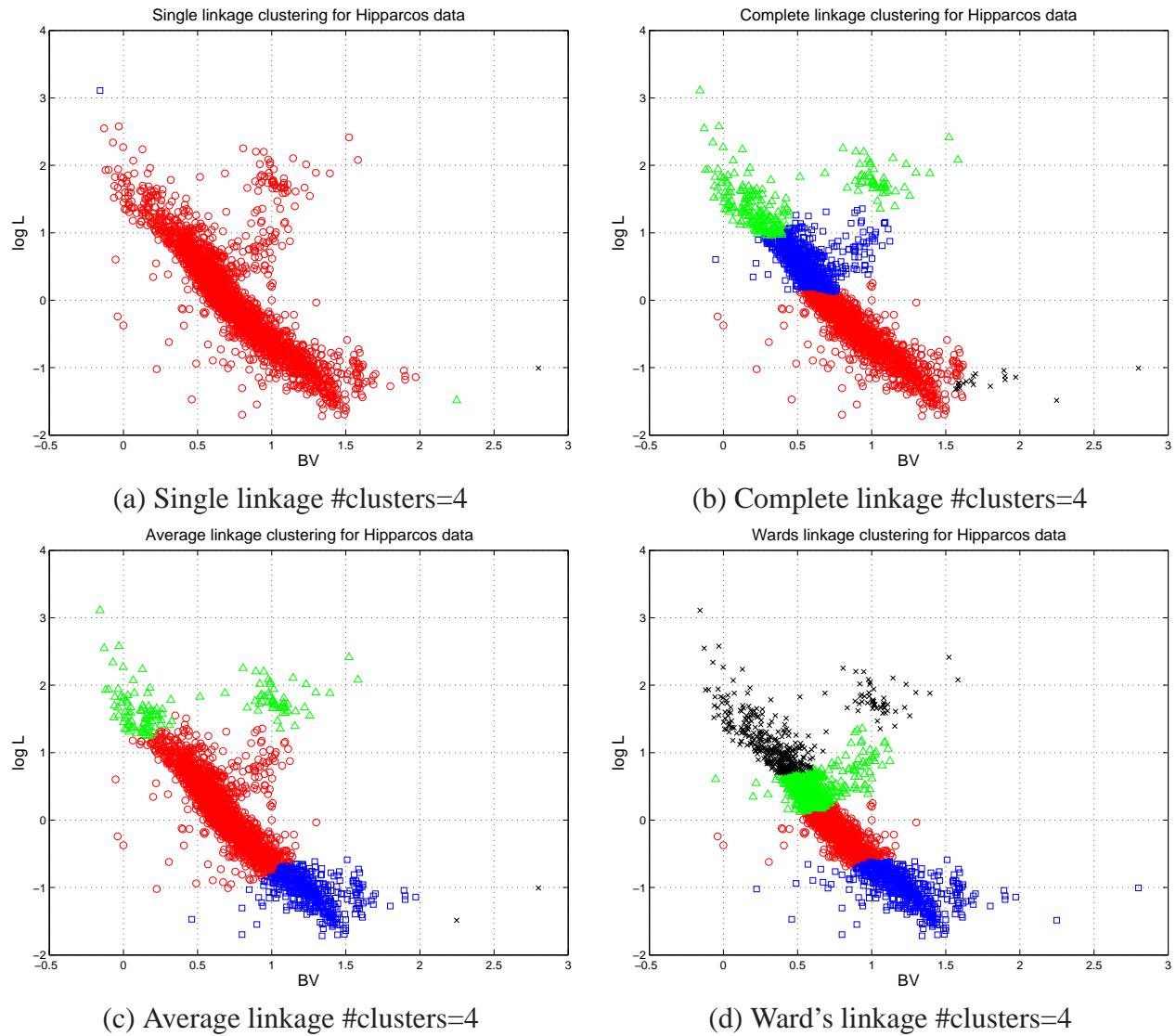


Figure 6: Clustering of the Hipparcos data

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.
- 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 μ_k , Σ_k . Denote the data by X , $X \in \mathcal{R}^d$. The density of component k is

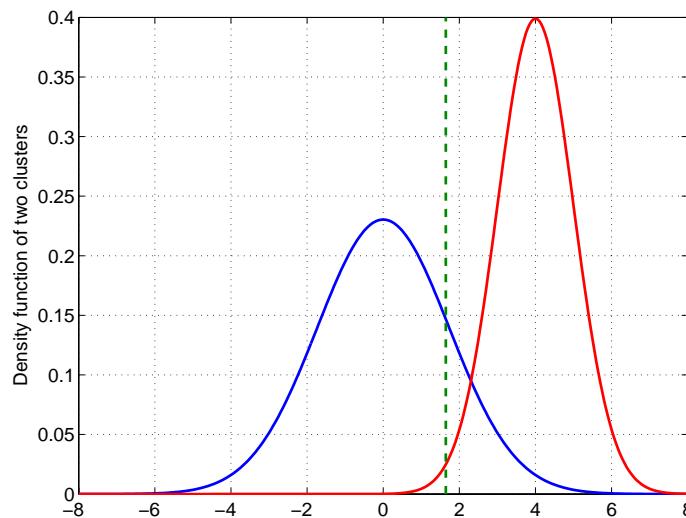
$$\begin{aligned} f_k(x) &= \phi(x \mid \mu_k, \Sigma_k) \\ &= \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} \exp\left(\frac{-(x - \mu_k)^t \Sigma_k^{-1} (x - \mu_k)}{2}\right). \end{aligned}$$

- 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 $f(x \mid \theta)$, with parameters $\theta \in \Omega$, on x . The distribution of y , $g(y \mid \theta)$ is

$$g(y \mid \theta) = \int_{\mathcal{X}(y)} f(\mathbf{x} \mid \theta) dx .$$

- The EM algorithm aims at finding a θ that maximizes $g(y \mid \theta)$ given an observed y .
- Introduce the function

$$Q(\theta' \mid \theta) = E(\log f(x \mid \theta') \mid y, \theta) ,$$

28

that is, the expected value of $\log f(x \mid \theta')$ according to the conditional distribution of x given y and parameter θ . The expectation is assumed to exist for all pairs (θ', θ) . In particular, it is assumed that $f(x \mid \theta) > 0$ for $\theta \in \Omega$.

- **EM Iteration:**

- E-step: Compute $Q(\theta \mid \theta^{(p)})$.
- M-step: Choose $\theta^{(p+1)}$ to be a value of $\theta \in \Omega$ that maximizes $Q(\theta \mid \theta^{(p)})$.

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, θ , includes: a_k, μ_k, Σ_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:

$$\begin{aligned}
 Q(\theta' | \theta) &= E \left[\log \prod_{i=1}^n a'_{y_i} \phi(x_i | \mu'_{y_i}, \Sigma'_{y_i}) \mid \mathbf{x}, \theta \right] \\
 &= E \left[\sum_{i=1}^n (\log(a'_{y_i}) + \log \phi(x_i | \mu'_{y_i}, \Sigma'_{y_i})) \mid \mathbf{x}, \theta \right] \\
 &= \sum_{i=1}^n E [\log(a'_{y_i}) + \log \phi(x_i | \mu'_{y_i}, \Sigma'_{y_i}) \mid x_i, \theta] .
 \end{aligned}$$

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

$$\begin{aligned} & E \left[\log(a'_{y_i}) + \log \phi(x_i \mid \mu'_{y_i}, \Sigma'_{y_i}) \mid x_i, \theta \right] \\ &= \sum_{k=1}^K p_{i,k} \log a'_k + \sum_{k=1}^K p_{i,k} \log \phi(x_i \mid \mu'_k, \Sigma'_k) . \end{aligned}$$

- Note that we cannot see the direct effect of θ in the above equation, but $p_{i,k}$ are computed using θ , i.e, the current parameters. θ' includes the updated parameters.
- We then have:

$$\begin{aligned} Q(\theta'|\theta) &= \sum_{i=1}^n \sum_{k=1}^K p_{i,k} \log a'_k + \\ &\quad \sum_{i=1}^n \sum_{k=1}^K p_{i,k} \log \phi(x_i \mid \mu'_k, \Sigma'_k) \end{aligned}$$

- Note that the prior probabilities a'_k and the parameters of the Gaussian components μ'_k, Σ'_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 μ_k and Σ_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 $i = 1, \dots, n, k = 1, \dots, K$.

$$p_{i,k} = \frac{a_k^{(p)} \phi(x_i \mid \mu_k^{(p)}, \Sigma_k^{(p)})}{\sum_{k=1}^K a_k^{(p)} \phi(x_i \mid \mu_k^{(p)}, \Sigma_k^{(p)})}.$$

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.

Comment: for mixtures of other distributions, the EM algorithm is very similar. The E-step involves computing the posterior probabilities. Only the particular distribution ϕ needs to be changed. The M-step always involves parameter optimization. Formulas differ according to distributions.

Computation Issues

- If a different Σ_k is allowed for each component, the likelihood function is not bounded. Global optimum is meaningless. (Don't overdo it!)
- How to initialize? Example:
 - Apply k-means first.
 - Initialize μ_k and Σ_k using all the samples classified to cluster k .
 - Initialize a_k by the proportion of data assigned to cluster k by k-means.
- In practice, we may want to reduce model complexity by putting constraints on the parameters. For instance, assume equal priors, identical covariance matrices for all the components.

Examples

- The heart disease data set is taken from the UCI machine learning database repository.
- There are 297 cases (samples) in the data set, of which 137 have heart diseases. Each sample contains 13 quantitative variables, including cholesterol, max heart rate, etc.
- We remove the mean of each variable and normalize it to yield unit variance.
- data are projected onto the plane spanned by the two most dominant principal component directions.
- A two-component Gaussian mixture is fit.

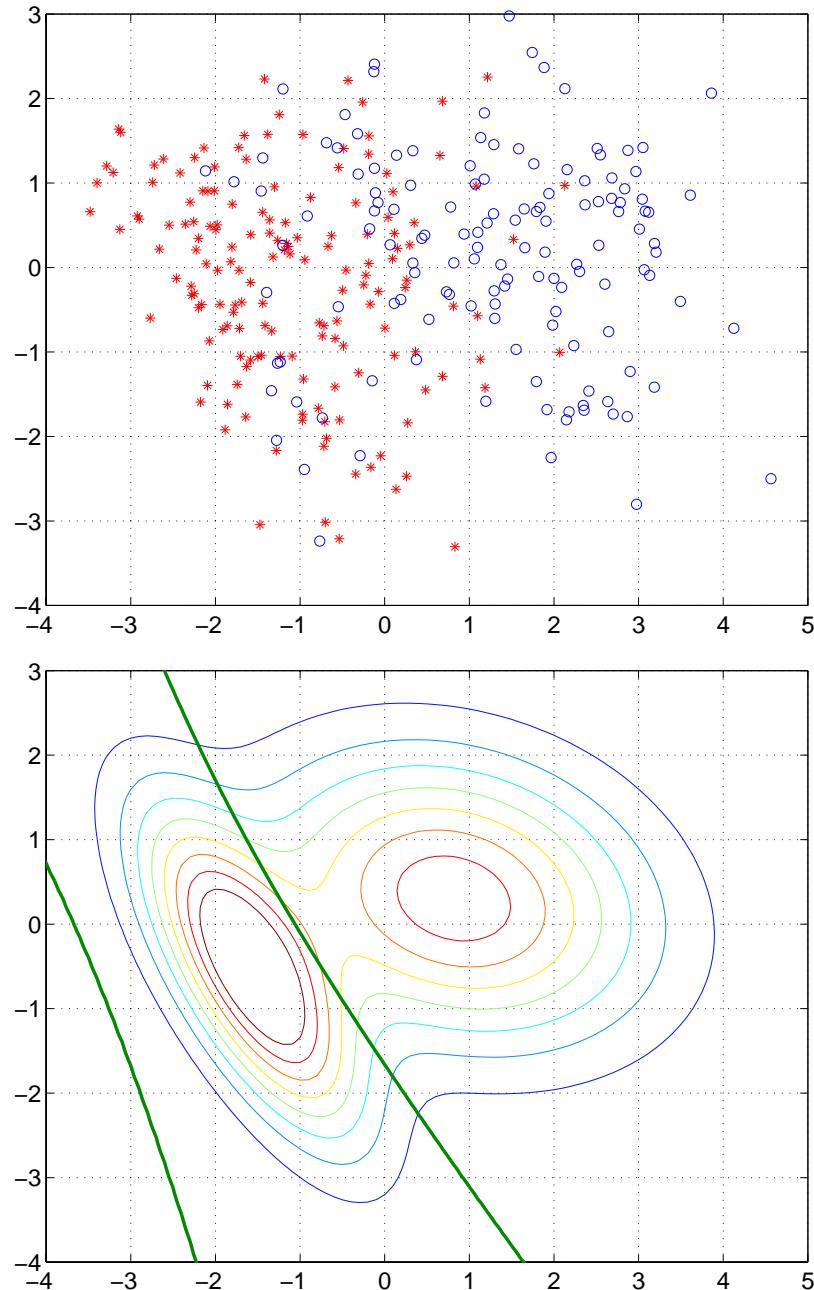


Figure 7: The heart disease data set and the estimated cluster densities. Top: The scatter plot of the data. Bottom: The contour plot of the pdf estimated using a single-layer mixture of two normals. The thick lines are the boundaries between the two clusters based on the estimated pdfs of individual clusters.