Lecture Notes for M761: Topics in Applied Statistics

Elizabeth Ann Housworth

Contents

1 Multi-Parameter Models and their Optimization **1**

1.1 Mixed Models 1

1.2 Structural Equation Models 8

1.3 Mixture Models 13

1.4 Combinatorial Models: Phylogeny Estimation 17

1.5 Derivative Free Search Methods in One Dimension 18

1.6 Powell's Method for Multidimensional Searches 22

1.7 The Simplex Method for Multidimensional Searches 26

1.8 The Expectation-Maximation Algorithm 28

1.9 Combinatorial Search Techniques: Simulated Annealing 32

1.10 Review 33

1.11 Projects 34

2 Techniques and Uses for Simulation **35**

2.1 Asymptotic Theory and its Limitations 35

2.2 Non-Parametric Bootstrapping 37

2.3 Better Bootstrapping Methods 43

2.4 Bootstrapping Examples and Counter-Examples 46

2.5 Parametric Bootstrapping 48

2.6 Projects 57

3 Bayesian Methods **59**

3.1 Bayesian Philosophy and Theory 59

3.2 Metropolis and Metropolis-Hastings Algorithms . . . 62
3.3 Nuisance Parameters and Conserved Synteny Example . . . 68
3.4 Combinatorial Example . . . 72
3.5 Projects . . . 74

4 Contingency Tables **75**
4.1 Sampling Schemes and Contingency Tables . . . 75
4.2 The χ^2 Statistic . . . 76
4.3 To Condition or Not to Condition? . . . 77
4.4 The Use of χ^2 as a Measure of Independence . . . 80
4.5 Random Effects MetaAnalyses . . . 81
4.6 Projects . . . 89

5 Data Mining **91**
5.1 Corrections for Conducting Too Many Tests . . . 91
5.2 Benjamini and Hochberg's False Discovery Rate . . . 92
5.3 Storey's Positive False Discovery Rate . . . 93
5.4 Projects . . . 94

A R Software **95**
A.1 Obtaining, Installing, and Using R . . . 95
A.2 Basic R Commands and Sample R Code . . . 96

B Useful Facts from Linear Algebra and Statistics **103**

C R Routines and Code for the Figures **105**

Chapter 1

Multi-Parameter Models and their Optimization

We spend the first few sections discussing some common multi-parameter models and the remainder of the chapter on numerical methods to find the optimum parameters for those (and other multi-parameter) models.

♢ 1.1 Mixed Models

Mixed models allow for both fixed and random effects in ANOVA applications. Typical applications are similar to the first two examples below. Such mixed models are handled well by most software packages today. Other examples such as applications to pedigrees have recently been added to SAS and some packages exist for R. Applications to phylogenetics are not routinely handled yet in standard statistical software packages. Mixed models provide a rich class of multi-parameter models requiring numerical optimization procedures.

Example 1 The following data provide an example of the common form of the mixed model. The original data come from Milliken and Johnson (1984) and the data are used as an example in the SAS documentation . Most software packages can handle mixed models of this type appropriately. That data involvees the rating of three machines by their operators. Machines are a fixed effect - there are 3 specific machines that exist and the company cares about. The person operating the machine is a random effect. A sample of 6 people from a large population of possible machine operators were chosen to participate in this experiment. But, although these 6 people are theoretically repeatable, the company is not interested in them per se, but only to the extent that they represent a larger population of people who will operate the machines.

The model for these data is:

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

where μ is the overall mean ranking, α_i is the fixed effect on the ranking of the i^{th} machine and no effect for the machines corresponds to $\alpha_1 = \alpha_2 = \alpha_3 = 0$, β_j is the random effect on the ranking due to the j^{th} person where the β_j is modeled as $N(0, \tau^2)$ and no random individual effect corresponds to $\tau^2 = 0$, $(\alpha\beta)_{ij}$ is the random interaction effect between the i^{th} machine and the j^{th} person modeled as $N(0, \eta^2)$, and ϵ_{ijk} is the residual for the k^{th} time person j operated machine i which is modeled as $N(0, \sigma^2)$.

Thus, y_{ijk} is normal with mean $\mu + \alpha_i$ and variance $\tau^2 + \eta^2 + \sigma^2$. The covariance structure determined by the random effects has $\text{COV}(y_{i_1jk_1}, y_{i_2jk_2}) = \tau^2$ when $i_1 \neq i_2$ and $\text{COV}(y_{ijk_1}, y_{ijk_2}) = \tau^2 + \eta^2$ for $k_1 \neq k_2$. This structure determines the multivariate distribution of the data and provides a likelihood functions whose optimum parameters can be determined. Software can work out this structure

from the information provided by the model.

machine 1			machine 2			machine 3		
person	rating		person	rating		person	rating	
1	52	y_{111}	1	64	y_{211}	1	67.5	y_{311}
2	51.8	y_{121}	2	59.7	y_{221}	1	67.2	y_{312}
2	52.8	y_{122}	2	60	y_{222}	1	66.9	y_{313}
3	60	y_{131}	2	59	y_{223}	2	61.7	y_{321}
4	51.1	y_{141}	3	68.6	y_{231}	2	62.3	y_{322}
5	50.9	y_{151}	4	63.2	y_{241}	3	70.8	y_{331}
5	51.8	y_{152}	4	62.8	y_{242}	3	70.6	y_{332}
5	51.4	y_{153}	4	62.2	y_{243}	3	71	y_{333}
6	46.4	y_{161}	5	64.8	y_{251}	4	66.2	y_{341}
6	44.8	y_{162}	5	65	y_{252}	4	64	y_{342}
			6	44.2	y_{261}	5	72.1	y_{351}
			6	43	y_{262}	5	72	y_{352}
						5	71.1	y_{353}
						6	61.4	y_{361}
						6	60.5	y_{362}

The variance-covariance structure for a subset of the data is given below:

$$\begin{pmatrix} y_{111} \\ y_{121} \\ y_{122} \\ y_{211} \\ y_{221} \end{pmatrix} \sim N\left(\begin{pmatrix} \mu+\alpha_1 \\ \mu+\alpha_1 \\ \mu+\alpha_1 \\ \mu+\alpha_2 \\ \mu+\alpha_2 \end{pmatrix}, \begin{pmatrix} \tau^2+\eta^2+\sigma^2 & 0 & 0 & \tau^2 & 0 \\ 0 & \tau^2+\eta^2+\sigma^2 & \tau^2+\eta^2 & 0 & \tau^2 \\ 0 & \tau^2+\eta^2 & \tau^2+\eta^2+\sigma^2 & 0 & \tau^2 \\ \tau^2 & 0 & 0 & \tau^2+\eta^2+\sigma^2 & 0 \\ 0 & \tau^2 & \tau^2 & 0 & \tau^2+\eta^2+\sigma^2 \end{pmatrix} \right)$$

The general form for the variance-covariance structure is $\sigma^2\mathbf{I} + \tau^2\mathbf{J} + \eta^2\mathbf{K}$ where $\mathbf{I}$ is the identity matrix, $\mathbf{J}$ is an incidence matrix containing 1's in row i and column j if the i^{th} and j^{th} ratings come from the same person and 0's otherwise, and $\mathbf{K}$ is an incidence matrix containing 1's in row i and column j if the i^{th} and j^{th} ratings come from the same person and the same machine and 0's otherwise. ■

There are different methods for calculating the significance of the random effects. Linear estimates based on least squares arguments used to be the standard fair in statistical computing software. More recently, maximum likelihood and restricted maximum likelihood (REML, where the fixed effects, including the overall mean, are removed first and often less biased variance estimates result) have been included in many software packages. For instance, PROC MIXED is SAS's advanced mixed model package, SPSS has an add-on package (Advanced Models) to handle ML and REML estimates, and there are free packages (lmm) you can install in R to find ML and REML estimates to examples such as this one. The key to the ability to use these packages lies in the fact that the structure of the random effects above is simple.

The next example has a more complicated covariance structure between the random effects.

Example 2 Suppose we have three drug regimens: regimen 1, regimen 2, and controls. Suppose 10 patients are assigned to each regimen and their antibody response is measured at 24 hours, 48 hours, and 72 hours past treatment. The drug regimens are fixed effects, the patient responses at each time are random effects. However, the responses in a given patient ought to be correlated over the three times. We need to choose among the various models for this correlation structure. Based on what we are led to believe by previous studies, let us assume a simple lag 1 autocorrelation structure works well. Then the model is

$$y_{ijk} = \mu + \alpha_i + \beta_{jk} + \epsilon_{ijk}$$

where μ is the overall mean, α_i is the fixed effect of the i^{th} drug regimen, β_{jk} is the random effect of the j^{th} person at the k^{th} time (these effects are NOT independent anymore), and ϵ_{ijk} are independent, identically distributed residual errors. The variance structure for the random effects for each individual follows as AR(1) model:

$$V = \tau^2 \begin{pmatrix} 1 & \rho & \rho^2 \\ \rho & 1 & \rho \\ \rho^2 & \rho & 1 \end{pmatrix}$$

The responses between individuals are, of course, independent, so the overall variance-covariance structure for the random effects has copies of V down the diagonal. ■

The ability to handle examples such as the above with complex covariance structure between the random effects (in this case, the response of each patient over the times involved in the study) are included in advanced mixed model procedures in most software although care has to be taken to ensure that the structure is programmed correctly into the model. Many possible covariance structures are included in these programs, but not all possible structures are included. The next set of examples do not involve covariance structures pre-programmed into the SAS mixed model procedure, for instance.

Example 3 (Mixed models in genetics and phylogenetics.) In genetic studies involving related individuals used, for instance, in trying to locate disease genes or to identify superior livestock for breeding, there is often a known relationship matrix describing the variance-covariance structure of the random genetic effects. For example, a parent provides half of his or her genetic material to his or her child, so the genetic correlation between parent and child is 50%. Similarly, siblings share, on average, 50% of their genetic material (25% from each parent is expected to be shared.) So the correlation between siblings is 50%. Half-siblings, on the other hand, are expected to share 25% of their genetic material through their common parent so their genetic correlation is 25%. Such correlations can be worked out for any genetic relationship, including inbreeding relationships where, say, first cousins marry.

Let $\mathbf{G}$ be the relationship matrix of genetic correlations for the individuals in a study. A model for one trait, y_i, is

$$y_i = \mu + \alpha_i + \epsilon_i$$

where α_i is the random genetic component to the trait and ϵ_i is the residual error. For instance, height is hereditary but nutrition and other environmental factors also play a role.

The model appears over-parameterized. However, one can estimate both α_i and ϵ_i because they have different variance-covariance structures. Note $\alpha \sim N(\mathbf{0}, \tau^2\mathbf{G})$ and $\epsilon \sim N(\mathbf{0}, \sigma^2\mathbf{I})$ assuming that you do not want or need to model potential covariance structure in the residuals due to shared environmental effects among siblings. At any rate, the covariance structures in α and ϵ are different, so you can tease the effects apart.

The model for the data is easily written as $\mathbf{y} \sim N(\mu\mathbf{1}, \tau^2\mathbf{G} + \sigma^2\mathbf{I})$.

There is one peculiarity about such models (which also occurs in other common applications of the mixed model). There is a natural requirement that $\tau^2\mathbf{G} + \sigma^2\mathbf{I}$ be a positive semidefinite matrix. However, the sum can be positive while one of the components is negative. That is, it is possible for the most likely estimates for τ^2 or σ^2 to be less than zero.

A pedigree can be drawn using circles (for women) and squares (for men).

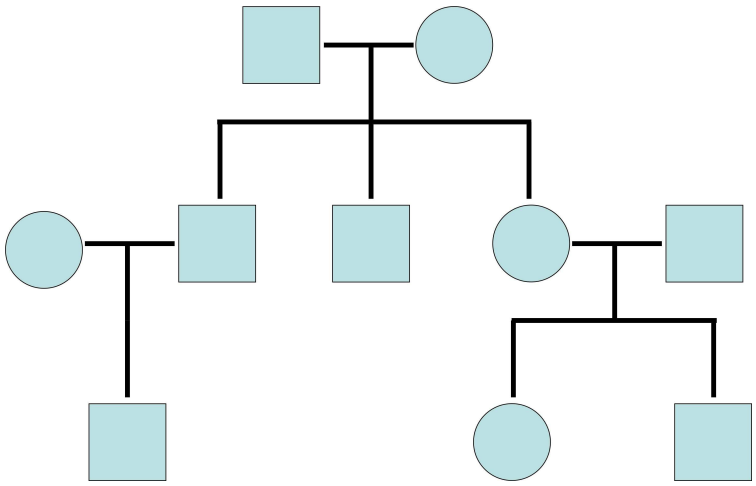


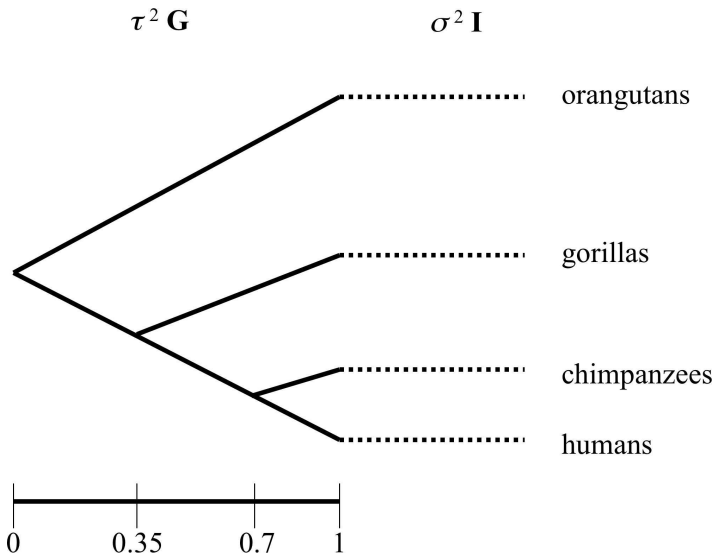
Figure 1.1: A simple pedigree involving a set of grandparents, their two sons and one daughter, spouses for a son and a daughter, and three grandchildren (two boys and a girl).

The relationship matrix for these individuals (in order left-to-right, top-to-bottom) is

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & .5 & .5 & .5 & 0 & .25 & .25 & .25 \\ 0 & 1 & 0 & .5 & .5 & .5 & 0 & .25 & .25 & .25 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & .5 & 0 & 0 \\ .5 & .5 & 0 & 1 & .5 & .5 & 0 & .5 & .25 & .25 \\ .5 & .5 & 0 & .5 & 1 & .5 & 0 & .25 & .25 & .25 \\ .5 & .5 & 0 & .5 & .5 & 1 & 0 & .25 & .5 & .5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & .5 & .5 \\ .25 & .25 & .5 & .5 & .25 & .25 & 0 & 1 & .125 & .125 \\ .25 & .25 & 0 & .25 & .25 & .5 & .5 & .125 & 1 & .5 \\ .25 & .25 & 0 & .25 & .25 & .5 & .5 & .125 & .5 & 1 \end{pmatrix}$$

SAS, for instance, now has a separate genetics package in which one can specify which individuals are parents to other individuals in the data set. The program then computes the genetic relationship matrix and uses it in the subsequent analyses.

The same model occurs in phylogenetics only the relationship matrix **G** is given by the proportion of evolutionary time shared by two species (when the model is that the trait evolution corresponds to a Brownian motion on the phylogeny). The model is used to determine the proportion of a trait measured on species in existence today that is due to the fact that the species shared evolutionary history versus the proportion of the trait that is due to species-specific effects and independent of the phylogeny. There is no standard statistical computing package of which I am aware that will allow the input of an arbitrary relationship matrix such as one needs for the phylogenetic model. A picture of a phylogeny with independent species-specific extensions is given below:



$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0.35 & 0.35 \\ 0 & 0.35 & 1 & 0.7 \\ 0 & 0.35 & 0.7 & 1 \end{pmatrix}$$

Figure 1.2: A graphical representation of the mixed model in phylogenetics.

Example 4 A more realistic pedigree example includes multiple traits which have both genetic correlations and environmental correlations. Such an example requires many more parameters and more notation. Let **G** be the $n \times n$ relationship matrix describing the genetic correlations between individuals and let **I** be the $n \times n$ identity matrix (we are assuming that the environmental residuals for the individuals are independent of the other individuals). For two traits y_1 and y_2 measured on these n individuals, let

$$\mathbf{A} = \begin{pmatrix} \sigma^2_{a,1} & \sigma_{a,12} \\ \sigma_{a,12} & \sigma^2_{a,2} \end{pmatrix} \quad \text{and} \quad \mathbf{E} = \begin{pmatrix} \sigma^2_{e,1} & \sigma_{e,12} \\ \sigma_{e,12} & \sigma^2_{e,2} \end{pmatrix}$$

be the covariance matrices describing the correlations between the genetic components of the two traits and the environmental components of the two traits. We assume that the genetic components are independent of the environmental components and vice versa. Then the multivariate normal

distribution describing the two traits can be written as

$$\begin{pmatrix} y_{1,1} \\ y_{1,2} \\ \vdots \\ y_{1,n} \\ y_{2,1} \\ y_{2,2} \\ \vdots \\ y_{2,n} \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_1\mathbf{1} \\ \mu_2\mathbf{1} \end{pmatrix}, (\mathbf{A}\otimes\mathbf{G} + \mathbf{E}\otimes\mathbf{I})\right)$$

where $\mathbf{1}$ is a vector of n ones and $\mathbf{A}\otimes\mathbf{G}$ and $\mathbf{E}\otimes\mathbf{I}$ are the Kronecker products:

$$\mathbf{A}\otimes\mathbf{G} = \begin{pmatrix} \sigma^2_{a,1}\mathbf{G} & \sigma_{a,12}\mathbf{G} \\ \sigma_{a,12}\mathbf{G} & \sigma^2_{a,2}\mathbf{G} \end{pmatrix} \quad \text{and} \quad \mathbf{E}\otimes\mathbf{I} = \begin{pmatrix} \sigma^2_{e,1}\mathbf{I} & \sigma_{e,12}\mathbf{I} \\ \sigma_{e,12}\mathbf{I} & \sigma^2_{e,2}\mathbf{I} \end{pmatrix}.$$

We now have 8 parameters to estimate. Further, with more traits there is a greater chance than at least one parameters will be "out of bounds," that is, that some variance will be estimated as negative or that the absolute value of some covariance will be larger than the square root of the product of the variances of the two components. ■

The naive analysis of pedigree data could require inverting extremely large matrices. In animal studies, for instances, thousands of animals can be involved. In studying human genes and diseases, hundreds of subjects are often involved. Ways around inverting such large matrices become very important [7, 8]. The bivariate mixed model has also been utilized in the analysis of traits between species [3].

References and Readings

[1] R P Freckleton, Harvey P H, and Pagel M. Phylogenetic analysis and comparative data. *American Naturalist*, 160:712–726, 2002.

[2] SAS Institute. *SAS OnlineDoc 9.1.3*. http://support.sas.com/onlinedoc/913/docMainpage.jsp, 2005.

[3] M Lynch. Methods for the analysis of comparative data in evolutionary biology. *Evolution*, 45:1065–1080, 1991.

[4] M Lynch and Walsh B. *Genetics and Analysis of Quantitative Traits*. Sinauer, Sunderland, Massachusetts, 1998.

[5] G A Milliken and Johnson D E. *Analysis of Messy Data, Volume I: Designed Experiments*. Lifetime Learning Publications, Belmont, CA, 1984.

[6] M Pagel. Inferring the historical patterns of biological evolution. *Nature*, 401:877–884, 1999.

[7] E A Thompson and Shaw R G. Pedigree analysis for quantitative traits: variance components without matrix inversion. *Biometrics*, 46:399–413, 1990.

[8] E A Thompson and Shaw R G. Estimating polygenic models for multivariate data on large pedigrees. *Genetics*, 131:971–978, 1992.

Conceptual Exercises for Section 1.1

1. Animal pedigrees sometimes involve inbreeding - the mating of relatives. If a brother and sister are mated and two offspring are produced, what is the genetic variance-covariance matrix for these 4 individuals? If a father and a daughter are mated and two offspring are produced, what is the genetic variance-covariance matrix for these 4 individuals?
2. In phylogenetics, the observed species trait value might come from averaging over a sample of individuals chosen from that species. Assume that the individuals are independent of each other. Can you draw a diagram of a mixed model for a single trait from several species (say, humans, chimpanzees, gorillas, and orangutans) that incorporates this sampling process? What is the corresponding mathematical model?

♢ 1.2 Structural Equation Models

Structural equation models (SEMs) are another rich class of models involving many parameters. The formal mathematical model due to Keesling, Wiley, and Jöreskog be written as:

$$\begin{aligned}
\boldsymbol{\eta}_{(m\times 1)} &= \mathbf{B}_{(m\times m)}\boldsymbol{\eta}_{(m\times 1)} + \boldsymbol{\Gamma}_{(m\times n)}\boldsymbol{\xi}_{(n\times 1)} + \boldsymbol{\zeta}_{(m\times 1)} \\
\mathbf{y}_{(p\times 1)} &= \boldsymbol{\Lambda}_{\mathbf{y}(p\times m)}\boldsymbol{\eta}_{(m\times 1)} + \boldsymbol{\epsilon}_{(p\times 1)} \\
\mathbf{x}_{(q\times 1)} &= \boldsymbol{\Lambda}_{\mathbf{x}(q\times n)}\boldsymbol{\xi}_{(n\times 1)} + \boldsymbol{\delta}_{(q\times 1)}
\end{aligned}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the observed variables, $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ are the unobserved, latent variables used to explain the observed variables, and $\boldsymbol{\zeta}, \boldsymbol{\delta}, \boldsymbol{\epsilon}$ are (unobserved) residuals. There is a general assumption that the individuals measured are independent of each other and that the residuals are normally distributed. Some structure within the residual components is allowed. For instance, the residuals for the responses, $\boldsymbol{\epsilon}$ might have a covariance structure if the responses constitute repeated measures.

Structural equation modeling subsumes much of multiple and multivariate regression models, regression with measurement error in the predictor variables, factor analysis, and path analysis.

A structural equation model written in purely mathematical terms is nearly impenetrable. However, such models can be written graphically, which provides a much more intuitive feeling for them. In the graph of the model, unobservable latent variables are placed in ovals while the observed variables are placed in rectangles. Arrows lead from "cause" to "effect." Since latent variables are not uniquely determined (they are only determined up to a scaling factor), one arrow leading from each latent variable must be given a scale (usually the scale 1).

Structural equation models are often used for "confirmatory factor analysis." That is, from previous studies involving the same variables, researchers have done factor analyses and have determined and named factors that seem to be involved in the explanation of the observed variables. The factors themselves may not be directly observable. For instance, intelligence is a concept that many people believe exists, but there is no good single measure of intelligence. IQ scores, SAT scores, and the like all involve intelligence but are not necessarily direct measures of intelligence. A factor analysis might take into consideration test scores, measures of success, measures of other kinds of reasoning skills, etc... and determine that some linear combination of these variables make up a factor called intelligence. Thus, the basic structure of the model is determined by the analysis of previous data sets. The purpose of structural equation modeling is to determine if the model suggested by the previous analyses works well with new data. The specific values of the parameters in the model, however, are determined by the new data.

Example 1 This example involves data from Wheaton *et al.* 1988 which is the classic example of SEMs discussed in the tutorials for the CALIS procedure in SAS [4], LISREL software [2], and the AMOS software addition to SPSS [1], among other places.

The model involves measured variables on 932 individuals in rural Illinois. The person's education and socioeconomic index were measured in 1967. The person's feelings of powerlessness and anomia

(self-worth, hopefulness, and belonging in society) were measured on a scale in 1967 and again in 1971. The model involves the unobservable latent variables Socio-Economic Status (SES) and Alienation (in 1967 and in 1971). Note that SES is somewhat like intelligence - it is a function of several factors. As a graduate student, you may think of yourself as being poor and your actual salary may be lower than that of a janitor. But you have made a decision to defer getting a higher salaried job in order to obtain an advanced degree which will likely help you achieve either a higher salary and/or favorable job conditions later and that decision might have been influenced by the socio-economic status in which you were raised. So SES is a function of income and education, at least. The model in this example has another latent variable, alienation, which influences a person's score on the anomia and powerlessness scales. Latent variables with arrows only pointing out from them are exogenous whereas ones with any arrows pointing into them are endogenous. That is, the assumption for exogenous latent variables are that they are random but independent of other factors in the model. Endogenous variables are not independent of other factors in the model.

A graphical representation of the model is given as:

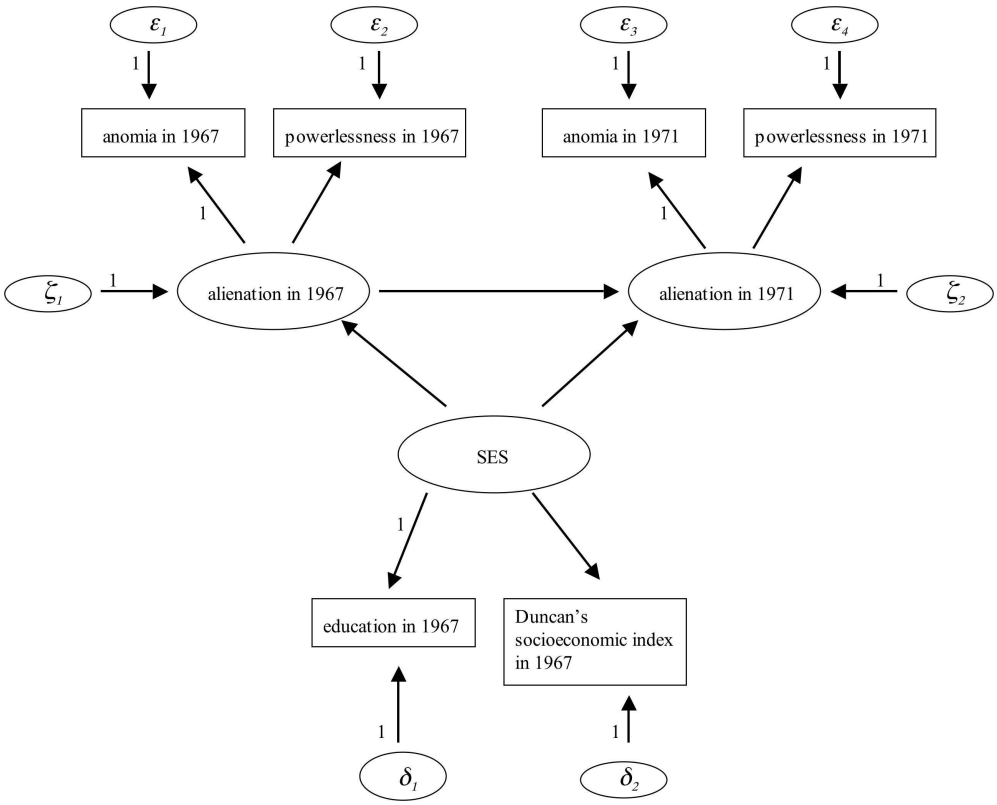


Figure 1.3: A graphical representation of a model of anomia and powerlessness in Illinois residents

The formal equations for this model have $p = 4$, $q = 2$, $m = 2$, $n = 1$ and

$$\boldsymbol{\eta} = \begin{pmatrix} \text{alienation 1967} \\ \text{alienation 1971} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ c & 0 \end{pmatrix} \begin{pmatrix} \text{alienation 1967} \\ \text{alienation 1971} \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix} (\text{ SES }) + \begin{pmatrix} \zeta_1 \\ \zeta_2 \end{pmatrix}$$

$$\begin{aligned}
\mathbf{y} &= \begin{pmatrix} \text{anomia 1967} \\ \text{powerlessness 1967} \\ \text{anomia 1971} \\ \text{powerlessness 1971} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ d & 0 \\ 0 & 1 \\ 0 & e \end{pmatrix} \begin{pmatrix} \text{alienation 1967} \\ \text{alienation 1971} \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \end{pmatrix} \\
\mathbf{x} &= \begin{pmatrix} \text{education 1967} \\ \text{socio-economic index 1967} \end{pmatrix} = \begin{pmatrix} 1 \\ f \end{pmatrix} \begin{pmatrix} \text{SES} \end{pmatrix} + \begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix}
\end{aligned}$$

The model has 6 regression coefficients $a - f$ and 9 variances (for the 8 residuals and the latent variable SES) to be calculated. Note that these parameters determine the theoretical variance-covariance matrix for the observed y and x variables. For instance,

$$\begin{aligned}
\text{Var}(\text{anomia 1967}) &= \text{Var}(\epsilon_1) + \text{Var}(\text{alienation 1967}) \\
&= \text{Var}(\epsilon_1) + a^2\text{Var}(\text{SES}) + \text{Var}(\zeta_1)
\end{aligned}$$

Note that

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{x} \\ \boldsymbol{\eta} \\ \boldsymbol{\xi} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}_\mathbf{y} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}_\mathbf{x} \\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \boldsymbol{\Gamma} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{x} \\ \boldsymbol{\eta} \\ \boldsymbol{\xi} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\epsilon} \\ \boldsymbol{\delta} \\ \boldsymbol{\zeta} \\ \boldsymbol{\xi} \end{pmatrix}$$

so that the theoretical variance-covariance matrix is given by:

$$\boldsymbol{\Sigma} = \mathbf{J}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{P}\left((\mathbf{I} - \mathbf{A})^{-1}\right)'\mathbf{J}'$$

where

$\mathbf{J}_{((p+q)\times(p+q+m+n))} = \left[\mathbf{I}_{((p+q)\times(p+q))}\mathbf{0}_{((p+q)\times(m+n))}\right]$ is a selection matrix

$$\mathbf{A}_{(p+q+m+n)\times(p+q+m+n)} = \begin{pmatrix} \mathbf{0}_{p\times p} & \mathbf{0}_{p\times q} & \boldsymbol{\Lambda}_{\mathbf{y},p\times m} & \mathbf{0}_{p\times n} \\ \mathbf{0}_{q\times p} & \mathbf{0}_{q\times q} & \mathbf{0}_{q\times m} & \boldsymbol{\Lambda}_{\mathbf{x},q\times n} \\ \mathbf{0}_{m\times p} & \mathbf{0}_{m\times q} & \mathbf{B}_{m\times m} & \boldsymbol{\Gamma}_{m\times n} \\ \mathbf{0}_{n\times p} & \mathbf{0}_{n\times q} & \mathbf{0}_{n\times m} & \mathbf{0}_{n\times n} \end{pmatrix}$$

is determined by the unknown regression coefficients and

$$\mathbf{P}_{(p+q+m+n)\times(p+q+m+n)} = \begin{pmatrix} \text{E}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}')_{p\times p} & & & \\ & \text{E}(\boldsymbol{\delta}\boldsymbol{\delta}')_{q\times q} & & \\ & & \text{E}(\boldsymbol{\zeta}\boldsymbol{\zeta}')_{m\times m} & \\ & & & \text{E}(\boldsymbol{\xi}\boldsymbol{\xi}')_{n\times n} \end{pmatrix}$$

is a block diagonal matrix of the unknown variances.

These regression coefficients and variance parameters are estimated using the variance-covariance matrix for the observed y and x variables. There are a total of 6 observed variables so there are 6 variances and $\binom{6}{2} = 15$ covariances observed versus 6 regression coefficients plus 9 variance parameters to be estimated.

The unknown parameters are found by minimizing the objective function of the observed variance-covariance matrix of the y and x variables, $\mathbf{S}$, and the unknown theoretical matrix $\mathbf{\Sigma}$ which depends on the 15 parameters:

$$F(\mathbf{S}, \mathbf{\Sigma}) = \log|\mathbf{\Sigma}| + \text{Tr}(\mathbf{S}\mathbf{\Sigma}^{-1})$$

The objective function comes from the negative log likelihood function with extraneous variables that are constant from model to model discarded. That is, if $\mathbf{z}_i$ is the vector of observed data for the i^{th} individual transformed to have mean zero (so that there are $n-1$ transformed individuals),

$$\begin{aligned}
-2\prod_{i=1}^{n-1}\log f(\mathbf{z}_i|\mathbf{\Sigma}) &= -2\prod_{i=1}^{n-1}\log\left(\frac{1}{\sqrt{2\pi|\mathbf{\Sigma}|}}e^{-\frac{1}{2}\mathbf{z}_i\mathbf{\Sigma}^{-1}\mathbf{z}_i'}\right) \\
&= (n-1)\log(2\pi) + (n-1)\log|\mathbf{\Sigma}| + \sum_{i=1}^{n-1}\mathbf{z}_i\mathbf{\Sigma}^{-1}\mathbf{z}_i' \\
\text{(the trace of a number is just the number)} &= (n-1)\log(2\pi) + (n-1)\log|\mathbf{\Sigma}| + \text{Tr}(\sum_{i=1}^{n-1}\mathbf{z}_i\mathbf{\Sigma}^{-1}\mathbf{z}_i') \\
(\text{trace}(AB) = \text{trace}(BA)) &= (n-1)\log(2\pi) + (n-1)\log|\mathbf{\Sigma}| + \text{Tr}(\sum_{i=1}^{n-1}\mathbf{z}_i\mathbf{z}_i'\mathbf{\Sigma}^{-1}) \\
\text{(definition of the sample variance matrix)} &= (n-1)\log(2\pi) + (n-1)\log|\mathbf{\Sigma}| + (n-1)\text{Tr}(\mathbf{S}\mathbf{\Sigma}^{-1})
\end{aligned}$$

■

After the most likely parameter values are determined, researchers interested in confirmatory factor analysis check to see if the model provides a good fit to the data. There are many goodness-of-fit tests. A common one is the Chi-Square goodness of fit test based on the likelihood ratio test of the model versus the null hypothesis that the best variance-covariance structure is the one determined by the data itself. The degrees of freedom for this test is the difference in the number of parameters in the two models. We have already noted that the SEM model requires 15 parameters and the null model has 21 parameters. So the Chi-Square goodness of fit test will have 6 degrees of freedom in this case.

References and Readings

[1] Information Technology Services at The University of Texas at Austin. *Structural Equation Modeling using AMOS: An Introduction*. http://www.utexas.edu/its/rc/tutorials/stat/amos/, 2002.

[2] Information Technology Services at The University of Texas at Austin. *LISREL 8 and PRELIS2: Getting Started*. http://www.utexas.edu/its/rc/tutorials/stat/lisrel/lisrel8.html, 2005.

[3] D Freedman. *Statistical models: Theory and practice*. Cambridge University Press, New York, 2005.

[4] SAS Institute. *SAS OnlineDoc 9.1.3*. http://support.sas.com/onlinedoc/913/docMainpage.jsp, 2005.

[5] K G Jöreskog. A general method for estimating a linear structural equation system. In A S Goldberger and O D Duncan, editors, *Structural Equation Models in the Social Sciences*. Academic Press, New York, 1973.

[6] J W Keesling. *Maximum Likelihood Approaches to Causal Analysis*. PhD thesis, University of Chicago, 1972.

[7] D E Wiley. The identification problem for structural equation models with unmeasured variables. In A S Goldberger and O D Duncan, editors, *Structural Equation Models in the Social Sciences*. Academic Press, New York, 1973.

Conceptual Exercises for Section 1.2

1. Draw a diagram for the following application of SEM to the multivariable regression: A researcher wants to predict two responses of economic success (raw income and a socio-economic index measure which takes into account savings, home ownership, etc...) from six explanatory variables (education, years of experience, job type, home ownership in the parents, level of religious observance, marital status) measured without error.

   What is the mathematical model that corresponds to your diagram for this problem?

2. Draw a diagram for the following application of SEM to the issue of measurement error in regression: A researcher wants to predict age based on the wear of teeth. For each specimen, a measure of age is provided. The wear on the teeth involves tedious observational counts. For each specimen, 3 student researchers measure the wear according to instructions but their reported values differ somewhat.

   What is the mathematical model that corresponds to your diagram for this problem?

3. In the example of the Wheaton study given, the measures of anomia and powerlessness constitute repeated measures since they were taken on each individual in both 1967 and 1971. What additional connections would be drawn on the graphical representation of the model that would allow for a correlation between the anomia measures in the two years and between the powerlessness measures in the two years? How does that affect the mathematical model? How many additional parameters does it add to the model?

4. Do the regularity conditions needed for the likelihood ratio in the Chi-Square goodness-of-fit test to be approximately Chi-Square distributed for large sample sizes hold for all SEMs?

♢ 1.3 Mixture Models

Mixture models are, as their name implies, models for the distribution of data that comes from 2 or more distinct, elementary, distributions. For instance, if boys have a normal distribution with a certain mean and variance, girls have a normal distribution with a different mean and variance, and you sample randomly from both sexes without recording the sex of the individual, the resulting distribution will likely be bimodal and will not be normally distributed but a mixture of two normal distributions. One need not stop with a mixture of only two elementary distributions - indeed there are even infinite mixture models. However, for the purposes of this text we will limit ourselves to mixtures involving only two distributions.

There is little general software for mixture models. There is a contributed package to R (nor1mix) for handling mixtures of simple (univariate) normal distributions such as the above example, there are some stand-alone packages for mixtures of normal distributions, and there are some packages for other specialized applications.

Example 1 Find the probability density for the data above.

Suppose boys have probability p of being sampled and have a mean μ_b and a variance σ_b^2 and, similarly, girls have a probability $1-p$ of being sampled with a mean of μ_g and a variance σ_g^2. Then the density function for a random point is

$$f(y|p,\mu_b,\sigma_b^2,\mu_g,\sigma_g^2) = \frac{1}{\sqrt{2\pi}}\left(\frac{p}{\sigma_b}e^{(y-\mu_b)^2/(2\sigma_b^2)} + \frac{1-p}{\sigma_g}e^{(y-\mu_g)^2/(2\sigma_g^2)}\right)$$

Note that mixture models have an inherent parameterization problem - if you replace p with $1-p$ and change the order of the boys' and girls' means and variances, you get the same likelihood. That is, if you search over all p's, means and variances, you will find that your likelihood function is bimodal with (at least two) identical peaks. Obvious ways around this is to require that the first mean in the parameter list is smaller than the second mean. Of course, unless we have an idea about the numbers of boys and girls sampled or where the means of the boys and girls should be relative to each other, we won't know from the data alone (with no sex information recorded) whether the smaller mean is the boys' mean or the girls'. But we can use the mixture model, after its parameters are estimated, to give the posterior probability a given point is in each of the two components. ■

Not all mixtures can be teased apart. If some events occur as a Poisson process with rate λ_1 and other events occur with rate λ_2 then the mixture of the two is simply a Poisson process with rate $\lambda_1+\lambda_2$. So, unless you have other information identifying the separate events, the two types can not be teased apart.

Example 2 A rather less trivial example of the use of mixtures comes from the analysis of the positions of crossovers along a tetrad. A little biology background will be useful to understanding the motivation here. When you produce an egg or a sperm, one of your diploid cells (with 23 chromosomes you inherited from your mother and 23 you inherited from your father) replicates all its chromosomes once and then divides twice, creating 4 cells with 23 chromosomes each. This process is called meiosis. Crossing over or recombination between pairs of maternally inherited and paternally inherited chromosomes occurs after the chromosomes duplicate and pair up and before

the divisions occur. When the chromosomes duplicate and pair up, the group of four chromosomes - the maternally inherited one and its duplicate and the paternally inherited one and its duplicate - is called a tetrad. The result of crossing over between maternal and paternal chromsomes is that the version of say, chromosome 1, that gets passed into an individual sperm cell or egg cell is not all maternally inherited DNA or paternally inherited DNA, but some combination of the two. In some organisms, such as yeast, all four products of meiosis can be recovered together. In humans, this is not possible. The analysis below is for tetrads. To analyze human data, the additional complication of "thinning" is required.

Distance along a chromosome can be measured in a variety of ways. The way that is useful here is Morgans, which is the expected number of crossovers on an individual chromosome. The length of an interval on the tetrad is simply 2 times the length in Morgans (only 2 out of the 4 chromsomes are affected by any crossover on the tetrad). That is, the rate of crossing over on the tetrad is twice the rate for an individual chromosome.

One model for the position of crossovers along a tetrad is that the position is exponentially distributed. This model is called the no-interference model. Crossovers simply occur at random. A few organisms seem to have this model. The distance between crossovers on the tetrad here has density $f(x) = 2e^{-2x}$.

Another model that fits more organisms well is the counting model. In this model, the spacing between crossovers follows a Gamma distribution with rate 2ν and shape parameter ν. If ν is a positive integer, then this is the count-poisson process where every ν^{th} point in a process with rate 2ν is a success and the intervening points are failures.

Let

$$f(x|\lambda,\nu) = \frac{\lambda^\nu}{\Gamma(\nu)} x^{\nu-1} e^{-\lambda x}$$

be the density for the Gamma distribution with rate λ and scale ν. Then the density for the for the counting model is $f(x|2\nu,\nu)$ and the density for the no-interference model is $f(x|2,1)$.

Before we consider the mixture model, let's consider how we would calculate the probability of observing crossovers at positions $s_0, s_1, \ldots, s_{(n-1)}$ along a tetrad of total length s Morgans. If we consider the crossing over process as a stationary renewal process, we see that we want to change the data to be of the form $x_0, x_1, \ldots, x_n$ where x_0 is the distance to the first crossover, x_1 is the distance between the first and second crossovers, etc..., and x_n is the distance from the last crossover to the end of the tetrad. The density for x_1, ..., $x_{(n-1)}$ is just the Gamma density given above. The endpoints require special consideration:

The density for the first crossover is the probability there is no crossover up to x_0 times the incremental probability that there is a crossover in $(x_0, x_0 + dx)$ which is

$$[1 - F(x_0|2\nu,\nu)](2dx)$$

where F is the cumulative distribution for $f(x|2\nu,\nu)$. The probability there is no crossover from x_n to the end of the chromosome is the probability that the distance to the crossover is greater than x_n and so is

$$1 - F(x_0|2\nu,\nu).$$

One last case requires consideration. What happens if the tetrad has no crossover whatsoever? In this case, x_0 is the length of the tetrad. The probability that this happens is the probability that the first crossover is off the chromosome:

$$1 - G(x_0|2\nu, \nu) \equiv \int_{x_0}^{\infty} [1 - F(x|2\nu, \nu)](2dx)$$

Thus

$$\Pr(x_0, x_1, \ldots, x_n) = \begin{cases} 1 - G(x_0|2\nu, \nu) & \text{if } n = 0 \\ 2[1 - F(x_0|2\nu, \nu)][1 - F(x_1|2\nu, \nu)] & \text{if } n = 1 \\ 2[1 - F(x_0|2\nu, \nu)] \prod_{i=1}^{n-1} f(x_i|2\nu, \nu)[1 - F(x_n|2\nu, \nu)] & \text{otherwise} \end{cases}$$

Now what happens if there is a mixture of independently placed (no-interfering) crossovers and crossovers that are spaced according to a counting model with $\nu > 1$. The above model (roughly) holds for each component. Suppose non-interfering crossovers occur with probability p. The shape remains the same but the rate is lowered to $2p$. Similarly, the counting crossovers occur with probability $1-p$ and their rate is lowered to $2(1-p)\nu$.

For distances x_0, x_1, $\ldots$, x_n as above, we consider the 2^n possible partitions into distances y_0, y_1, $\ldots$, y_j for the non-interfering crossovers and z_0, z_1, $\ldots z_k$ for the counting crossovers where the sum of each set is the length of the chromosome.

Considering each set separately, we see that Thus

$$\Pr(y_0, \ldots, y_j|p) = \begin{cases} \int_{y_0}^{\infty} [1 - F(y|2p, 1)](2pdy)) & \text{if } n = 0 \\ 2p[1 - F(y_0|2p, 1)][1 - F(y_1|2p, 1)] & \text{if } n = 1 \\ 2p[1 - F(y_0|2p, 1)] \prod_{i=1}^{j-1} f(y_i|2p, 1)[1 - F(y_j|2p, 1)] & \text{otherwise} \end{cases}$$

and

$$\Pr(z_0, \ldots, z_k|1-p, \nu) = \begin{cases} \int_{z_0}^{\infty} [1 - F(z|2\nu(1-p), \nu)](2(1-p)dz) & \text{if } n = 0 \\ 2(1-p)[1 - F(y_0|2(1-p)\nu, \nu)][1 - F(y_1|2(1-p)\nu, \nu)] & \text{if } n = 1 \\ 2(1-p)[1 - F(y_0|2(1-p)\nu.\nu)] \prod_{i=1}^{k-1} f(y_i|2(1-p)\nu, \nu)[1 - F(y_k|2(1-p)\nu, \nu)] & \\ & \text{otherwise} \end{cases}$$

The probability of the data is

$$\Pr(x_0, x_1, \ldots, x_n|p, \nu) = \sum_{\text{partitions}} \Pr(y_0, y_1, \ldots, y_j|p, 1)\Pr(z_0, z_1, \ldots, z_k|1-p, \nu).$$

The likelihood function is then the product of the probability of the observed tetrad patterns. This example requires a complicated calculation of the mixture density. ■

References and Readings

[1] N T J Bailey. *Introduction to the Mathematical Theory of Genetic Linkage*. Oxford University Press, London, 1961.

[2] K W Broman and Weber J L. Characterization of human crossover interference. *Am J Hum Genet*, 66:1911–1926, 2000.

[3] E A Housworth and Stahl F W. Crossover interference in humans. *Am J Hum Genet*, 73:188–197, 2003.

[4] G J McLachlan and Peel D. *Finite Mixture Models*. John Wiley & Sons, Inc., New York, 2000.

[5] H Zhao and Speed T P. On genetic map functions. *Genetics*, 142:1369–1377, 1996.

Conceptual Exercises for Section 1.3

1. To get the crossover pattern for a chromosome (rather than the tetrad) you randomly thin the tetrad crossovers, selecting on average half of the crossovers to go to any particular chromsome. What is the distribution of the inter-crossover distances along a chromosome under the no-interference model?

♢ 1.4 Combinatorial Models: Phylogeny Estimation

♢ 1.5 Derivative Free Search Methods in One Dimension

We are assuming that the functions we want to optimize, such as the likelihood functions discussed in the previous sections, are unpleasant and differentiating them with respect to their parameters is worth avoiding. Thus, we are going to only discuss derivative-free optimization algorithms. This section covers two derivative-free univariate minimization algorithms and an algorithm to determine a bracket for the minimum when natural boundaries for the parameters are not available. One of the most efficient multivariate minimization algorithm, Powell's method, discussed in the next section, relies on univariate line searches.

The first optimization algorithm we discuss is the simple golden section algorithm. The golden section algorithm relies on the beautiful mathematical properties of the golden ratio

$$r = \frac{\sqrt{5}-1}{2}.$$

When you break a line from 0 to 1 at r, the ratio of the longer segment to the whole is the same as the ratio of the shorter segment to the longer segment. That is,

$$\frac{r}{1} = \frac{1-r}{r} \qquad \text{or} \qquad r^2 = 1-r$$

A rectangle in golden section proportions can be divided into the union of a square and a rectangle that itself is in golden section proportions.

The nice thing about using the golden section in a minimization routine is that you can limit the number of new function evaluations required at each step to one. Routines that minimize the number of function evaluations are critical when dealing with likelihood functions, and become increasingly valuable as the sample size increases.

All routines assume that there is only one (local) minimum in the interval to find. (We abuse the definition of local minima here to include local minima at the endpoints of the interval if the function is larger at all values interior to the interval in a neighborhood of the endpoint.) If there are multiple local minima, no routine short of a complete search (possibly accomplished by some branch and bound algorithm that does not require evaluation at every single point) will be guaranteed to converge on the global minimum, although some are probabilistically likely to do so (such as simulated annealing algorithms).

Given that there is a single local minimum for a function f in the interval $[a, b]$, the golden section algorithm considers two points, $x < y$ where $x = b - r * (b - a) = ra + (1 - r)b$ and $y = a + r * (b - a) = rb + (1 - r)a$. If $f(x) < f(y)$ then you know that the minimum of the function is in the interval $[a, y]$. Note that

$$a + r(y - a) = a + r(rb + (1 - r)a - a) = (1 - r^2)a + r^2 b = ra + (1 - r)b = x$$

So that when we set $b = y$, we also have as our new y our old x. There is a symmetrical argument in the case when $f(x) > f(y)$.

The golden section algorithm requires you to provide an interval on which the minimum will occur. In some statistical examples, this interval occurs naturally. For instance, a correlation has natural

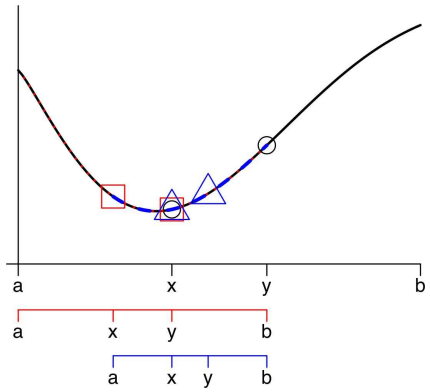


Figure 1.4: An illustration of the golden section algorithm

bounds -1 and $+1$. Other parameterizations of the same model may not lead to such natural fixed bounds. For instance, the bounds for covariances depends on the values of the variances of the two variables involved, which may also need estimating. Whether one estimates variances or covariances is simply a choice of parameterization. A variance is naturally bounded below by 0, but needs bounding above. Means have no obvious theoretical bounds either below or above. Although bounds for means and variances may be guessed at by examining data in each case, we need am all-purpose algorithm for bracketing minima.

We say that a triple $a < c < b$ *bracket* a minimum for the function f if $f(c)$ is smaller than either $f(a)$ or $f(b)$.

To bracket a minimum, we start with 2 points, $a < c$. Assume that $f(c) < f(a)$. (If not, we can reverse the argument below.) We need a third point, b, beyond c. Obtaining this point requires a somewhat arbitrary scaling factor. We will use a factor of 2.

If $a < c < b$ do not bracket the minimum, then we can try to fit a parabola through these points. We can compute where the minimum of the parabola occurs (u). The formula for u is:

$$u = c - \frac{1}{2}\frac{(c-a)^2[f(c)-f(b)] - (b-c)^2[f(c)-f(a)]}{(c-a)[f(c)-f(b)] + (b-c)[f(c)-f(a)]}$$

If u provides a bracket missed by c then $c < u < b$. If all goes well, $f(u) < f(b) < f(c)$ so that the bracket is $c < u < b$. If $f(u) < f(c) < f(a)$ then $a < c < u$ is a bracket. But we have to worry in this case because f is increasing within (u, c) and decreasing within (c, b). But, nonetheless, we have a bracket for at least a local minimum.

If $u > b$, then we might want to use u provide a endpoint for a new set to fit a parabola to. But sometimes, this might be a bad idea. For instance, if the function f is very flat around a, b, and c, u may be extremely far away and the parabolic fit may be unreasonable. (Further, in such a case, the denominator in the definition of u may be too small to be handled well by the code.) So we set a limit as to how far we will move in any one step relative to the length of our current interval.

A basic algorithm can be constructed from these observations:

1. Define the downhill direction.

2. Find a third point in the downhill direction. Check if we have found a bracket.

3. If not, fit a parabola through these points. If the denominator of the minimum for the parabola is close to zero, do not consider this point but a large bounding value instead.

4. If the point that gives the minimum to the parabola is within our original interval, see if it provides use with a bracket. If the bracket is on the wrong piece of the original interval, print a warning. Otherwise, if it doesn't provide a bracket, it wasn't a good point, so use a small extension in the right direction instead.

5. If the point that gives the minimum to the parabola is not within our original interval, see if it is a reasonable distance away. If it isn't too far away, use it as the new endpoint.

6. If the point is too far away, use a large bounding value instead.

There are many possible modifications to the algorithm proposed above. The scaling factors are rather arbitrary. Clearly, there will be problems applying the above algorithm as it is currently written to functions that have a limited domain.

The final univariate, derivative-free, algorithm we will discuss is Brent's algorithm. It can converge on the minimum at a quadratic rate when the function is approximately a parabola. This rate of convergence is ideal for maximum likelihood settings when one has enough data and begins reasonably close to the maximum likelihood estimate. Of course, who knows what happens far away from the maximum likelihood estimate. Brent's algorithm has a fallback plan: if the parabolic fit is very bad, use a golden section step instead.

A parabolic step is declared good if it falls within the current bounding interval and if the movement required is less than half the movement of the step before last. This second requirement is heuristic - you allow one bad step to occur before you throw away the parabolic approach because the quadratic convergence of the parabolic approach means that it could make it up to you the next time.

The implementation is somewhat tricky. You should not evaluate points that are too close together because you will have roundoff error problems. How close is too close? Second order Taylor series expansions indicate that too close is on the order of the square root of machine precision:

$$f(x) \approx f(b) + \frac{1}{2} f''(b)(x-b)^2$$

implies that when

$$|x-b| < \sqrt{\epsilon}|b| \sqrt{\frac{2|f(b)|}{b^2 f''(b)}}$$

the second order correction is negligible, (here ϵ is machine precision). The square root term is written that way because, for approximately quadratic functions, it is O(1) for large $|b|$.

Brent's algorithm highlights many features of good programming: it minimizes function evaluations, it knows when to stop because continuing will not be informative, it has an efficient main strategy with good convergence properties, it has a less efficient but sound fallback plan if it looks like the main strategy is not working in a particular case, and it has a robust mechanism for determining which of the two strategies to use.

Brent's algorithm is given in R in Appendix C. However, it is also already coded in R by the command "optimize." In fact, if you look at the help menu for the commands "optimize", "optim," and "nlm," you see that most of the algorithms we discuss have been pre-programmed for you.

References and Readings

[1] W H Press, Teukolsky S A, Vetterling W T, and Flannery B P. *Numerical Recipes in Fortran 77: The art of scientific computing.* Cambridge University Press, Port Chester, New York, second edition, 1992.

Conceptual Exercises for Section 1.5

1. The simple mixed model in genetics and phylogenetics is of the form:

   $$\mathbf{y} \sim N(\mu\mathbf{1}, \tau^2\mathbf{G} + \sigma_e^2\mathbf{I})$$

   Reparameterize the model using $\sigma^2 = \tau^2 + \sigma_e^2$ and $h^2 = \dfrac{\tau^2}{\tau^2 + \sigma_e^2}$. For each value of h^2 can you find explicit formulae for the maximum likelihood estimates of σ^2 and μ? This method reduces the likelihood function to a function of the single parameter h^2 where the natural limits for h^2 are 0 and 1. For the data in Project 1, use a univariate optimization algorithm to find the maximum likelihood estimates for log body mass. Do the same for the metabolic rate.

2. Design a more appropriate bracketing algorithm for a variance parameter than the generic bracketing algorithm given in this section. That is, modify the algorithm so that the bracketing interval does not include negative numbers, ever. Is there another way of handling the problem of bracketing intervals including negative numbers but negative variance estimates possibly not being allowed within your likelihood function? That is, is there another way of handling the bracketing of a minimum for a function not defined for $x < 0$.

3. Both the bracket and Brent's algorithm use position where the minimum of a parabola through 3 points $(a, f(a))$, $(b, f(b))$, and $(c, f(c))$ occurs. Determine the formula for this point yourself and see if it agrees with the one given in the algorithms.

♢ 1.6 Powell's Method for Multidimensional Searches

Powell's method (and the multivariate optimization methods based on using the derivatives of the function) are based on the notion of conjugate directions. Consider trying to find the minimum of a paraboloid $z = f(x, y) = x^2 + 5 * y^2$, say. If we start at any initial points (x_0, y_0) and minimize over y holding x fixed and then minimize over x holding y fixed, you see that we get to the minimum at $(x, y) = (0, 0)$ in these two steps.

However, if the paraboloid is $z = f(x, y) = \frac{(x+y)^2}{2} + 5\frac{(x-y)^2}{2}$, then using the approach of minimizing parallel to the axes provides only a slow approach to the minimum. (Try it.) The problem is that these directions are interfering with each other. For any quadratic form, there is a set of conjugate directions so that the minimization in one direction is not changed when we move in another direction. OUR function is not a quadratic form, but it can be approximated by one locally and will look quadratic at the minimum.

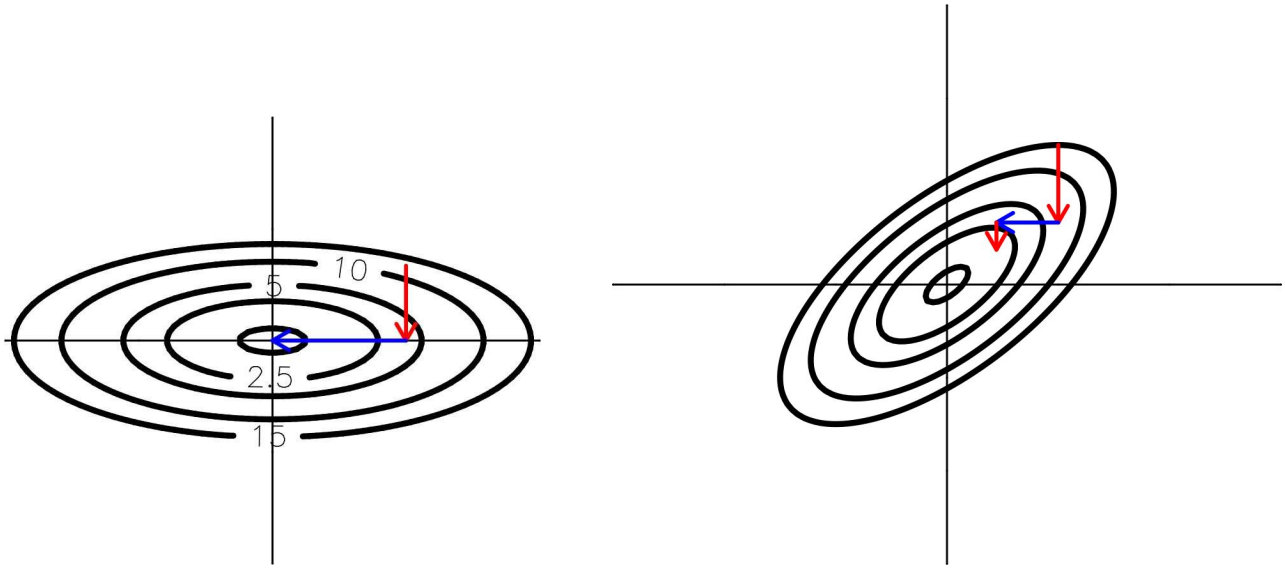


Figure 1.5: Illustrations of minimizing paraboloids moving only parallel to the axes.

Approximate f by its second order Taylor series:

$$f(\mathbf{x} + \mathbf{P}) \approx c + \mathbf{b}' \cdot \mathbf{x} + \frac{1}{2}\mathbf{x}' \cdot \mathbf{A} \cdot \mathbf{x}$$

where $c = f(\mathbf{P})$, $\mathbf{b} = \nabla f|_{\mathbf{P}}$, and $\mathbf{A}$ is the Hessian matrix with $A_{i,j} = \frac{\partial^2}{\partial x_i \partial x_j}|_{\mathbf{P}}$.

Based on the approximation for f, $\nabla f \approx \mathbf{A} \cdot \mathbf{x} + \mathbf{b}$. The change in the gradient as we move in some direction is:

$$\delta(\nabla f) \approx \mathbf{A} \cdot (\delta \mathbf{x})$$

If we find a minimum in some direction $\mathbf{u}$, the gradient at the minimum is perpendicular to $\mathbf{u}$. We want to preserve this property when we choose a new direction in which to move. That is, we

want to find $\mathbf{v}$ so that

$$0 = \mathbf{u}' \cdot \delta(\nabla f) \approx \mathbf{u}' \cdot \mathbf{A} \cdot \mathbf{v}$$

If this holds, directions $\mathbf{u}$ and $\mathbf{v}$ are said to be *conjugate* to each other and a collection of vectors that are pairwise conjugate is called a *conjugate set*. If we can come up with a linearly independent conjugate set, then, by definition, we will reach the minimum of a quadratic form with p line searches where p is the dimension of $\mathbf{x}$. Why?

Let f be defined as:

$$f(\mathbf{x}) = \mathbf{x}' \cdot \mathbf{A} \cdot \mathbf{x} - 2\mathbf{b}' \cdot \mathbf{x} + c$$

where $\mathbf{A}$ is positive definite and symmetric. If we have a conjugate set $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_m\}$, then the minimum of $f(\mathbf{x})$ for $\mathbf{x}$ in the span of the conjugate set occurs at

$$\sum_{i=1}^{m} \beta_i \mathbf{u}_i \qquad \text{where} \qquad \beta_i = \frac{\mathbf{u}_i' \cdot \mathbf{b}}{\mathbf{u}_i' \mathbf{A} \mathbf{u}_i}$$

To show this, let $\mathbf{x} = \sum_{i=1}^{m} \alpha_i \mathbf{u}_i$ and note that:

$$\begin{aligned}
f(\mathbf{x}) &= \left(\sum_{i=1}^{m} \alpha_i \mathbf{u}_i\right)' \cdot \mathbf{A} \cdot \left(\sum_{i=1}^{m} \alpha_i \mathbf{u}_i\right) - 2\mathbf{b}' \cdot \left(\sum_{i=1}^{m} \alpha_i \mathbf{u}_i\right) + c \\
&= \sum_{i=1}^{m} \alpha_i^2 \mathbf{u}_i' \cdot \mathbf{A} \cdot \mathbf{u}_i - 2\sum_{i=1}^{m} \alpha_i \mathbf{b}' \cdot \mathbf{u}_i + c \\
&= \sum_{i=1}^{m} (\alpha_i - \beta_i)^2 \mathbf{u}_i' \cdot \mathbf{A} \cdot \mathbf{u}_i + 2\sum_{i=1}^{m} \alpha_i \beta_i \mathbf{u}_i' \cdot \mathbf{A} \cdot \mathbf{u}_i - 2\sum_{i=1}^{m} \alpha_i \mathbf{b}' \cdot \mathbf{u}_i - \sum_{i=1}^{m} \beta_i^2 \mathbf{u}_i' \cdot \mathbf{A} \cdot \mathbf{u}_i + c \\
&= \sum_{i=1}^{m} (\alpha_i - \beta_i)^2 \mathbf{u}_i' \cdot \mathbf{A} \cdot \mathbf{u}_i + \textit{Const.}
\end{aligned}$$

which is clearly minimized when α_i is equal to β_i as claimed.

Where does this get us? It says that the line minimum of

$$f\left(\sum_{i \neq j} \alpha_i \mathbf{u}_j + t\mathbf{u}_i\right)$$

occurs at $t = \beta_i$ and that we can do the minimization over the conjugate directions in any order to reach the minimum of the quadratic form.

How do we determine a set of conjugate directions. The following was Powell's idea in 1964: If the minimum of $f(\mathbf{x})$ in the direction $\mathbf{u}$ from the point $\mathbf{x}_1^*$ is at $\mathbf{x}_1$ and the minimum in the direction $\mathbf{u}$ from the point $\mathbf{x}_2^*$ is at $\mathbf{x}_2$, then $\mathbf{x}_2 - \mathbf{x}_1$ is conjugate to $\mathbf{u}$. Why? Note that, by definition

$$\frac{d}{dt} f(\mathbf{x}_1 + t\mathbf{u}) = \frac{d}{dt} f(\mathbf{x}_2 + t\mathbf{u}) = 0$$

and that

$$\frac{d}{dt}f(\mathbf{x}+t\mathbf{u}) = \frac{d}{dt}\left((\mathbf{x}+t\mathbf{u})'\mathbf{A}(\mathbf{x}+t\mathbf{u}) - 2\mathbf{b}'(\mathbf{x}+t\mathbf{u}) + c\right) = 2\mathbf{u}'(\mathbf{A}(\mathbf{x}+t\mathbf{u}) - \mathbf{b})$$

so that, when $t = 0$, $\mathbf{u}'\mathbf{A}(\mathbf{x}_2 - \mathbf{x}_1) = 0$.

For functions that are not quadratic forms, we are not guaranteed to reach the minimum in p line searches but Powell's algorithm based on conjugate directions does converge quadratically to the minimum.

Powell's algorithm is deceptively simple. Let $\{\mathbf{e}_i, i = 1, \ldots, p\}$ be an orthonormal basis for your domain. Initially set $\mathbf{u}_i = \mathbf{e}_i$. Repeat the following steps p times:

- Save your starting position $\mathbf{P}_0$
- For $i = 1, 2, .., p$ move from $\mathbf{P}_{i-1}$ to the minimum along direction $\mathbf{u}_i$ and call the position of this minimum $\mathbf{P}_i$.
- Bump up the directions: for $i = 1, 2, \ldots, p-1$, set $\mathbf{u}_i = \mathbf{u}_{i+1}$.
- Replace the last direction, set it to the best overall move: $\mathbf{u}_p = \mathbf{P}_p - \mathbf{P}_0$.
- Move from $\mathbf{P}_p$ to the minimum along direction $\mathbf{u}_p$ and call the position of this minimum $\mathbf{P}_0$.

The last move on the previous iteration and the next-to-last move on the current iteration are in the same direction so that the new direction added is conjugate to the last direction added. Note that each iteration involves $p + 1$ line minimizations and so the entire procedure above involves $p(p + 1)$ steps. If f is quadratic, we are guaranteed to be at the minimum. If f is not quadratic, we need to reset the directions to our orthonormal basis and restart the routine above because, by the end of the routine, our direction set has folded up on itself. Here is a picture of Powell's algorithm in 2 dimensions with a paraboloid and in general;

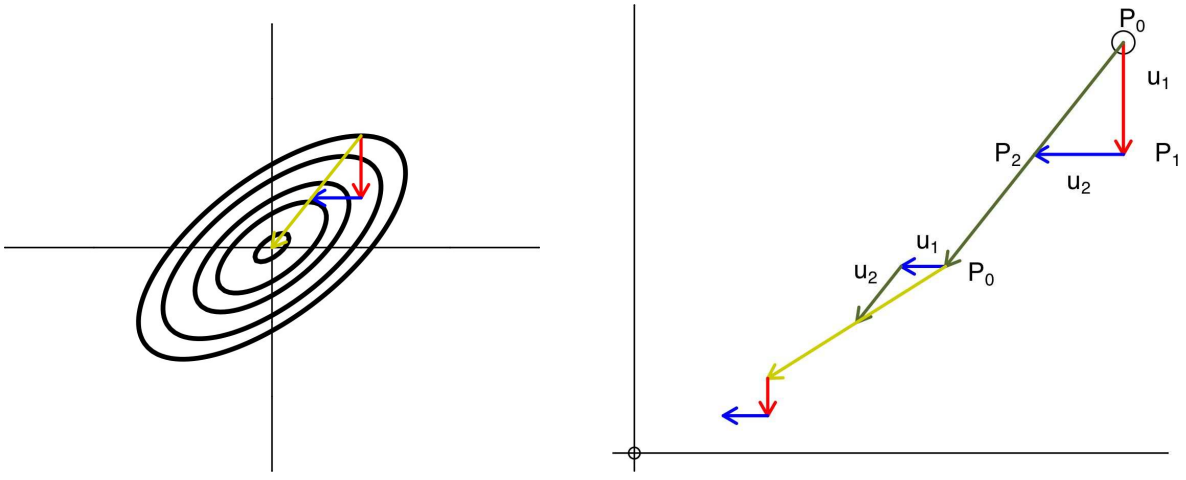


Figure 1.6: Illustrations of Powell's algorithm in 2 dimensions with a paraboloid and in general.

Powell's method as described above is simple to program, requires no differentiation, and achieves quadratic convergence (which means that if you are near the minimum or if you function is nearly quadratic, you reach the minimum quickly, but does not mean anything if you are far away from the minimum and your function is nasty).

References and Readings

[1] R P Brent. *Algorithms for Minimization without Derivatives.* Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

[2] M J D Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computation Journal*, 7:155–162, 1964.

[3] W H Press, Teukolsky S A, Vetterling W T, and Flannery B P. *Numerical Recipes in Fortran 77: The art of scientific computing.* Cambridge University Press, Port Chester, New York, second edition, 1992.

Conceptual Exercises for Section 1.6

1. Powell's algorithm still must call a line minimization function. If you have to bracket the minimum each time as well as find the minimum within the bracket, this could be computationally burdensome. What things could you try to include in an implementation of Powell's method to reduce the computational time required to bracket the minimum?

♢ 1.7 The Simplex Method for Multidimensional Searches

The simplex algorithm of Nelder and Mead involves a simplex (hence, its name) moving around, reflecting, reflecting and expanding, contracting, and shrinking. Assuming we have p variables, we identify the simplex with $p+1$ points in p-space. The algorithm is as follows:

- Order the point so that

  $$f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \cdots \leq f(\mathbf{x}_p) \leq f(\mathbf{x}_{(p+1)})$$

  so that $\mathbf{x}_1$ is the best point found so far and $\mathbf{x}_{(p+1)}$ is the worst.

- Let $\mathbf{c}$ be the center of the first p best points; i. e.,

  $$\mathbf{c} = \frac{\sum_{i=1}^{p} \mathbf{x}_i}{p}$$

- Reflect the worst point through the center of the others. That is, let $\mathbf{x}_r = \mathbf{c} + \alpha(\mathbf{c} - \mathbf{x}_{(p+1)})$.

- If the reflected point is better than the second worst one but worst than the best one, accept it. That is, if $f(\mathbf{x}_1) \leq f(\mathbf{x}_r) \leq f(\mathbf{x}_p)$ then replace $\mathbf{x}_{(p+1)}$ with $\mathbf{x}_r$ and start the algorithm from the beginning again.

- If the reflected point is better than the best one, then try expanding in that direction. That is, if $f(\mathbf{x}_r) \leq f(\mathbf{x}_1)$ then compute $\mathbf{x}_s = \mathbf{c} + \beta(\mathbf{c} - \mathbf{x}_{(p+1)})$

- If the expansion is better still, use it. That is, $f(\mathbf{x}_s) \leq f(\mathbf{x}_r)$ then replace $\mathbf{x}_{(p+1)}$ with $\mathbf{x}_s$. Otherwise, replace $\mathbf{x}_{(p+1)}$ with $\mathbf{x}_r$. Start the algorithm from the beginning again.

- If reflecting does not work, try contracting. That is, if you get here, compute $\mathbf{x}_c = \mathbf{c} + \gamma(\mathbf{c} - \mathbf{x}_{(p+1)})$. If $f(\mathbf{x}_c) \leq f(\mathbf{x}_{(p+1)})$, replace $\mathbf{x}_{(p+1)}$ with $\mathbf{x}_c$ and start the algorithm form the beginning. (This step allows the algorithm to "ooze down a valley floor" in Press et al.'s language)

- If you get here, you contract towards the best point. (This step allows the "amoeba" to "pass through the eye of a needle" in Press et al.'s language.) Replace $\mathbf{x}_i$ with $\mathbf{x}_1 + \delta(\mathbf{x}_i - \mathbf{x}_1)$.

Usually, $\alpha = 1$, $\beta = 2$, $\gamma = 0.5$, and $\delta = 0.5$.

References and Readings

Conceptual Exercises for Section 1.7

1. Does the Simplex Algorithm work in one dimension as a line search?
2. How might you modify the algorithm to deal with parameters that have natural bounds (correlations between -1 and 1, variances greater than zero, etc...)?

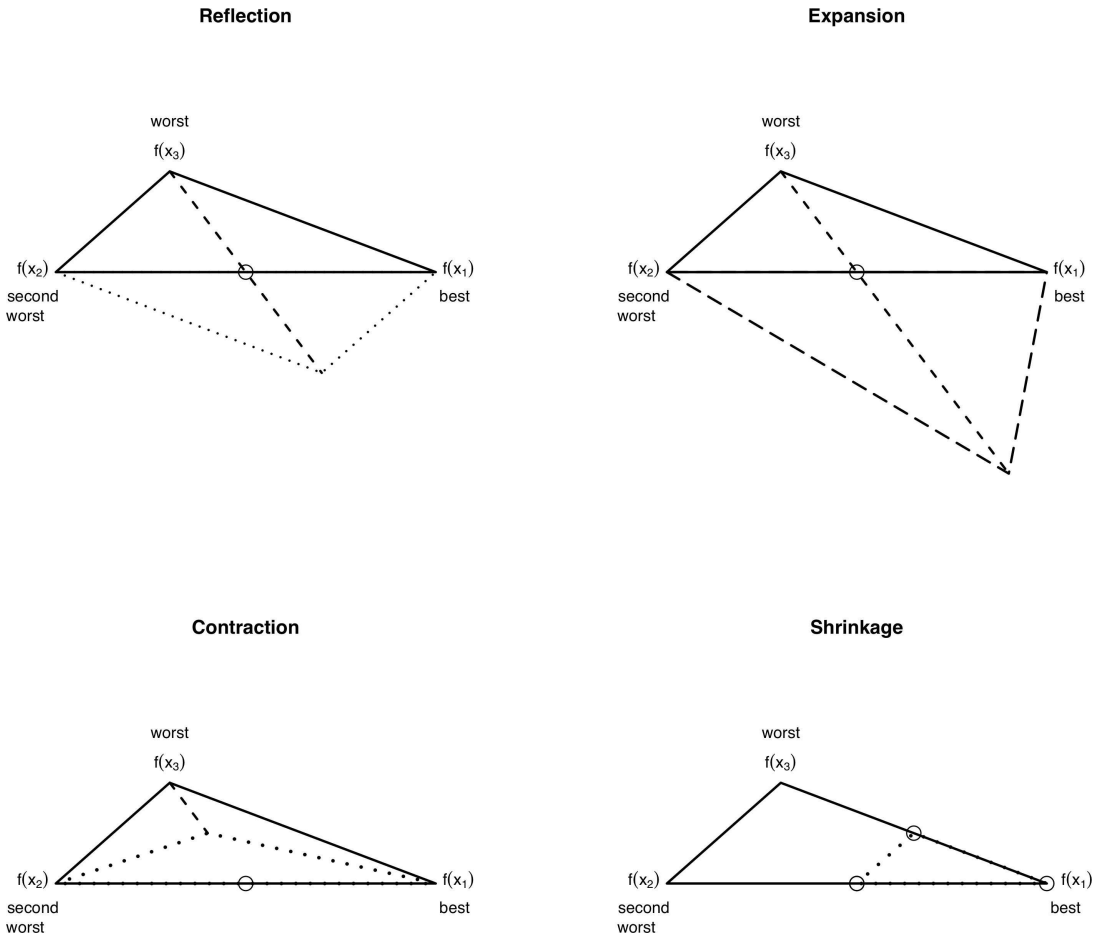


Figure 1.7: The possible moves in the Simplex "Amoeba" algorithm of Nelder and Mead

♢ 1.8 The Expectation-Maximation Algorithm

The EM algorithm is a technique for finding maximum likelihood estimators in the presence of missing data. The technique was used informally before Dempster, Laird, and Rubin developed a rigorous theoretical framework for the method in 1977. Missing data can be true missing data - patients drop out of studies, plants in the lab die, and so forth. However, several of the models previously discusses can be viewed as missing data problems. In mixed models, the random effects can be viewed as missing data. In the mixture model, which distribution a specific data point x_i was taken can be viewed as missing data. In the structural equation models, the values of the latent variables can be viewed as missing data. In each case, if we knew the missing data, the models would become much easier to analyze. The EM algorithm draws on that simplicity.

Let $\mathbf{Y}_{\text{obs}}$ be the data we actually measure. Let $\mathbf{Y}_{\text{mis}}$ be the missing data. Let $\mathbf{Y}$ be the complete data (both the observed and the missing data). Note that the distribution of the complete data can be factored as:

$$f(\mathbf{Y}|\boldsymbol{\theta}) = f(\mathbf{Y}_{\text{obs}}|\boldsymbol{\theta})f(\mathbf{Y}_{\text{mis}}|\mathbf{Y}_{\text{obs}}, \boldsymbol{\theta})$$

where f refers to the corresponding density function (that is, $f(\mathbf{Y}|\boldsymbol{\theta})$ is the density of the complete data, $f(\mathbf{Y}_{\text{obs}}|\boldsymbol{\theta})$ is the density for the observed data, and $f(\mathbf{Y}_{\text{mis}}|\mathbf{Y}_{\text{obs}}, \boldsymbol{\theta})$ is the conditional density of the missing data given the observed data).

Let l denote the corresponding log likelihood functions. The log likelihood function of the complete data can thus be decomposed as:

$$l(\boldsymbol{\theta}|\mathbf{Y}) = l(\boldsymbol{\theta}|\mathbf{Y}_{\text{obs}}) + \ln f(\mathbf{Y}_{\text{mis}}|\mathbf{Y}_{\text{obs}}, \boldsymbol{\theta})$$

Thus,

$$l(\boldsymbol{\theta}|\mathbf{Y}_{\text{obs}}) = l(\boldsymbol{\theta}|\mathbf{Y}) - \ln f(\mathbf{Y}_{\text{mis}}|\mathbf{Y}_{\text{obs}}, \boldsymbol{\theta})$$

Take the conditional expectation of both sides given the observed data and the current estimate of $\boldsymbol{\theta}$ which we will denote $\boldsymbol{\theta}^{(k)}$ to get:

$$l(\boldsymbol{\theta}|\mathbf{Y}_{\text{obs}}) = Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}) - H(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$$

where

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}) = \int l(\boldsymbol{\theta}|\mathbf{Y}_{\text{obs}}, \mathbf{Y}_{\text{mis}}) f(\mathbf{Y}_{\text{mis}}|\mathbf{Y}_{\text{obs}}, \boldsymbol{\theta}^{(k)}) dY_{\text{mis}}$$

and

$$H(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}) = \int \left(\ln f(\mathbf{Y}_{\text{mis}}|\mathbf{Y}_{\text{obs}}, \boldsymbol{\theta})\right) f(\mathbf{Y}_{\text{mis}}|\mathbf{Y}_{\text{obs}}, \boldsymbol{\theta}^{(k)}) dY_{\text{mis}}$$

Note that $H(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$ is maximized at $\boldsymbol{\theta} = \boldsymbol{\theta}^{(k)}$ by Jensen's inequality ($E\ln(X) \leq \ln(EX)$ since log is concave.)

The EM algorithm chooses $\boldsymbol{\theta}^{(k+1)}$ to maximize (the M step) the conditional expectation (the E step) $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$. Note that

$$l(\boldsymbol{\theta}^{(k+1)}|\mathbf{Y}_{\text{obs}}) - l(\boldsymbol{\theta}^{(k)}|\mathbf{Y}_{\text{obs}}) = [Q(\boldsymbol{\theta}^{(k+1)}|\boldsymbol{\theta}^{(k)}) - Q(\boldsymbol{\theta}^{(k)}|\boldsymbol{\theta}^{(k)})] + [H(\boldsymbol{\theta}^{(k)}|\boldsymbol{\theta}^{(k)}) - H(\boldsymbol{\theta}^{(k+1)}|\boldsymbol{\theta}^{(k)})]$$

so that the likelihood is non-decreasing at each iteration. If the likelihood remains constant for the iteration, then the difference in the Q's is zero as well. And if θ^* is a point where the gradient of the likelihood function is zero, then it is also a point where the gradient of Q is zero.

The EM algorithm can be notoriously slow to converge and it can converge to a saddle point or to local rather than global optimums. However, it can also be simple to program and an important tool when working with large pedigrees because clever manipulations allow you to implement it without inverting large relationship matrices (see [6] and [7]).

The EM algorithm is also not a true algorithm in that it has to be adapted to each problem separately. Let us consider several examples.

Example 1 Consider the example of a mixture of two normal distributions and assume that the mean of population 1 is smaller than the mean for population 2 so that the two populations are identifiable. For each observed data point y_i there is a missing data value z_i which is 1 if data point i is from population 1 and zero otherwise. Let $p(k), \mu_1(k), \sigma_1^2(k), \mu_2(k)$, and $\sigma_2^2(k))$ be the estimates for the parameters at the end of the k^{th} iteration. For the E-step, we compute

$$Q(p, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2|p(k), \mu_1(k), \sigma_1^2(k), \mu_2(k), \sigma_2^2(k)) =$$
$$E\left(l(p, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2|\mathbf{y}, \mathbf{z})|\mathbf{y}, p(k), \mu_1(k), \sigma_1^2(k), \mu_2(k), \sigma_2^2(k)\right)$$

The complete likelihood function is:

$$\begin{aligned} l(p, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2|\mathbf{y}, \mathbf{z}) &= -\frac{n}{2}\log(2*\pi) - \frac{\sum z_i}{2}\log(\sigma_1^2) - \frac{\sum z_i(y_i-\mu_1)^2}{2*\sigma_1^2} \\ &\quad -\frac{n-\sum z_i}{2}\log(\sigma_2^2) - \frac{\sum(1-z_i)*(y_i-\mu_2)^2}{2*\sigma_2^2} \end{aligned}$$

Note that the complete-data log likelihood function is linear in the unobserved variables, z_i. The complete-data maximum likelihood estimates can be written as:

$$\hat{\mu}_1 = \frac{\sum z_i y_i}{\sum z_i} \qquad \hat{\sigma}_1^2 = \frac{\sum\left(z_i y_i - \frac{\sum z_i y_i}{\sum z_i}\right)^2}{\sum z_i}$$

$$\hat{\mu}_2 = \frac{\sum(1-z_i)y_i}{\sum(1-z_i)} \qquad \hat{\sigma}_1^2 = \frac{\sum\left((1-z_i)y_i - \frac{\sum(1-z_i)y_i}{\sum(1-z_i)}\right)^2}{\sum(1-z_i)}$$

$$\hat{p} = \frac{\sum z_i}{n}$$

The new estimates of these parameters relies on the conditional expectation of the missing data z_i given the observed data and current parameter estimates. The conditional expectation of the missing data is:

$$E\left(z_i|\mathbf{y}, p(k), \mu_1(k), \sigma_1^2(k), \mu_2(k), \sigma_2^2(k)\right) = \frac{p(k)f_1(y_i)}{p(k)f_1(y_i) + (1-p(k))f_2(y_i)} \equiv a_i$$

where f_1 and f_2 are the normal densities with means $\mu_1(k)$, $\mu_2(k)$ and variances $\sigma_1^2(k)$, $\sigma_2^2(k)$, respectively.

Replacing z_i with its conditional expectation gives new estimates of the parameters. That is,

$$\mu_1(k+1) = \frac{\sum a_i y_i}{\sum a_i} \qquad \sigma_1^2(k+1) = \frac{\sum \left(a_i y_i - \frac{\sum a_i y_i}{\sum a_i}\right)^2}{\sum a_i}$$

$$\mu_2(k+1) = \frac{\sum (1-a_i) y_i}{\sum (1-a_i)} \qquad \sigma_1^2(k+1) = \frac{\sum \left((1-a_i) y_i - \frac{\sum (1-a_i) y_i}{\sum (1-a_i)}\right)^2}{\sum (1-a_i)}$$

$$\hat{p}(k+1) = \frac{\sum a_i}{n}$$

The process is repeated until it converges on a solution. If one is worried about whether the solution is a global maximum, one can re-start the process using different initial values for the parameters, as is usually done to gain confidence in the optimum. ■

Example 2 Consider the simple mixed model $y_i = \mu + a_i + \epsilon_i$ where $\mathbf{a}$ is normal with mean zero and variance $\tau^2\mathbf{G}$ and $\boldsymbol{\epsilon}$ is normal with mean zero and variance $\sigma^2\mathbf{I}$ independent of the a_i's. Here that a_i's can be viewed as missing data. The complete data log likelihood function is

$$l(\mu, \tau^2, \sigma^2|\mathbf{y}, \mathbf{a}) = -n\log(2\pi) - \frac{1}{2}\log(|\mathbf{G}|) - \frac{1}{2}n\log(\tau^2) - \frac{\mathbf{a}'\mathbf{G}^{-1}\mathbf{a}}{2\tau^2} - \frac{1}{2}n\log(\sigma^2) - \frac{(\mathbf{y}-\mathbf{a}-\mu)'(\mathbf{y}-\mathbf{a}-\mu)}{2\sigma^2}$$

Sufficient statistics can be given as:

$$\sum(y_i - a_i), \qquad \sum(y_i - a_i)^2, \qquad \mathbf{a}'\mathbf{G}^{-1}\mathbf{a}$$

and maximum likelihood estimates based on the complete-data likelihood function are

$$\hat{\mu} = \frac{\sum(y_i - a_i)}{n}, \qquad \hat{\sigma}^2 = \frac{\sum(y_i - a_i - \hat{\mu})^2}{n}, \qquad \hat{\tau}^2 = \frac{\mathbf{a}'\mathbf{G}^{-1}\mathbf{a}}{n}$$

In order to compute the conditional expectations of these statistics given the observed data and current parameter estimates, we make use of the following facts from multivariate normal theory:

If

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{a} \end{pmatrix} \sim N\left(\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}\right)$$

Then the conditional distribution of $\mathbf{a}$ given $\mathbf{y}$ is normal with mean $\boldsymbol{\mu}_2+\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\mathbf{y}-\boldsymbol{\mu}_1)$ and variance $\boldsymbol{\Sigma}_{22}-\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}$.

Further, $E(\mathbf{y}'\mathbf{A}\mathbf{y}) = \text{tr}\,(\mathbf{A}\text{var}(\mathbf{y})) + \text{mean}(\mathbf{y})'\mathbf{A}\text{mean}(\mathbf{y})$.

Let $\mu(k)$, $\tau^2(k)$, and $\sigma^2(k)$ be the current parameter estimates. Let $\mathbf{z}$ be the conditional expectation of $\mathbf{a}$ given $\mathbf{y}$ and the current parameter estimates. Note that, in our application, $\boldsymbol{\Sigma}_{11} = \tau^2(k)\mathbf{G}+\sigma^2(k)\mathbf{I}$, $\boldsymbol{\Sigma}_{12} = \tau^2(k)\mathbf{G}$, $\boldsymbol{\Sigma}_{21} = \tau^2(k)\mathbf{G}$, and $\boldsymbol{\Sigma}_{22} = \tau^2(k)\mathbf{G}$. Also note that $\boldsymbol{\mu}_1 = \mu(k)\mathbf{1}$ and $\boldsymbol{\mu}_2 = 0$. Let $\mathbf{V}(k) = \tau^2(k)\mathbf{G}+\sigma^2(k)\mathbf{I}$. Thus, the conditional expectation of the missing data given the observed data and current parameter estimates is:

$$\mathbf{z} = \tau^2(k)\mathbf{G}\left[\mathbf{V}(k)\right]^{-1}(\mathbf{y}-\mu(k)\mathbf{1})$$

and the conditional variance of the missing data is:

$$\tau^2(k)\mathbf{G}-\tau^2(k)\mathbf{G}\left[\mathbf{V}(k)\right]^{-1}\tau^2(k)\mathbf{G} = \tau^2(k)\mathbf{G}\left[\mathbf{V}(k)\right]^{-1}\left(\mathbf{V}(k)-\tau^2(k)\mathbf{G}\right) = \tau^2(k)\sigma^2(k)\mathbf{G}\left[\mathbf{V}(k)\right]^{-1}$$

The updated parameter estimates are

$$\begin{aligned}
\mu(k+1) &= E(\hat{\mu}|\mathbf{y},\mu(k),\tau^2(k),\sigma^2(k)) &&= \frac{\sum(y_i-z_i)}{n}\\
\tau^2(k+1) &= E(\hat{\tau}^2|\mathbf{y},\mu(k),\tau^2(k),\sigma^2(k)) &&= \frac{\mathbf{z}'\mathbf{G}^{-1}\mathbf{z}+\text{tr}\left(\sigma^2(k)\tau^2(k)\left[\mathbf{V}(k)\right]^{-1}\right)}{n}\\
\sigma^2(k+1) &= E(\hat{\sigma}^2|\mathbf{y},\mu(k),\tau^2(k),\sigma^2(k)) &&= \frac{\sum(y_i-\mu(k)-z_i)^2+\text{tr}\left(\sigma^2(k)\tau^2(k)\mathbf{G}\left[\mathbf{V}(k)\right]^{-1}\right)}{n}
\end{aligned}$$

References and Readings

[1] A P Dempster, Laird N M, and Rubin D B. Maximum likelihood from incomplete data via the em algorithm (with discussion). *Journal of the Royal Statistical Society, Ser. B*, pages 1–38, 1977.

[2] R J A Little and Rubin D B. *Statistical Analysis with Missing Data*. John Wiley and Sons, Hoboken, New Jersey, 2002.

[3] G J McLachlan and Peel D. *Finite Mixture Models*. John Wiley & Sons, Inc., New York, 2000.

[4] G J McLachlan and Krishnan T. *The EM Algorithm and Extensions*. John Wiley & Sons, Inc., New York, 1997.

[5] S R Searle, Casella G, and McCulloch C E. *Variance Components*. John Wiley and Sons, Hoboken, New Jersey, 1992.

[6] E A Thompson and Shaw R G. Pedigree analysis for quantitative traits: variance components without matrix inversion. *Biometrics*, 46:399–413, 1990.

[7] E A Thompson and Shaw R G. Estimating polygenic models for multivariate data on large pedigrees. *Genetics*, 131:971–978, 1992.

Conceptual Exercises for Section 1.8

1.

♢ 1.9 Combinatorial Search Techniques: Simulated Annealing

♢ 1.10 Review

When determining an optimum is a high-dimensional parameter space, you should never trust the output of one run of an optimization algorithm. The first "optimum" you find could be a flat region of parameter space, a saddle point, a local rather than global optimum, or some other erroneous answer. Restarting the simplex method at the optimum can provide evidence that you are at least at a local minimum. It is also advisable to run the optimization code with multiple starting points to provide some (probabilistic) assurance that you have found the global optimum. Visualization of the nature of parameter space is difficult at best in high dimensions. Looking at various one and two dimensional projections helps understand some of what is going on, but not everything.

For example, take the data for project 1 and consider modeling the metabolic rate via a mixed model. A slight distortion of the negative log likelihood function will cause it to be defined everywhere and to have four-fold symmetry:

Let $\mathbf{V}(\tau^2, \sigma^2) = (|\tau^2| + 0.001)\mathbf{G} + (|\sigma^2| + 0.001)\mathbf{I}$. Then the negative log likelihood function is

$$l(\mu, \tau^2, \sigma^2|\mathbf{y}) = n\log(2\pi) + \frac{1}{2}\log(|\mathbf{V}(\tau^2, \sigma^2)|) + \frac{1}{2}(\mathbf{y} - \mu)'\mathbf{V}^{-1}(\mathbf{y} - \mu)$$

Optimizing this function using R's built in Nelder-Mead algorithm with starting parameter values $\mu = 0$, $\tau^2 = 1$, and $\sigma^2 = 2$ gives you an optimum value of 316.07638246 which occurs at the point $\mu = 79.50211$, $\tau^2 = 1101.37303$, $\sigma^2 = 702.50504$. If one looks at the function at the parameter values $\mu = 79.50$, $\tau^2 = 1101$, $\sigma^2 = 550$ instead, one sees that the value is 315.8765 which is smaller. What's going on?

In the direction $(0.08917550, -0.01887277, -0.13559291)$, we see are are at a minimum. In the direction , we see that we are level. So this point is a kind of a saddle point.

Conceptual Exercises for Section 1.10

1.

♢ 1.11 Projects

Instructions

All projects should be written up as short research papers. You should present the data, the question, your methods, and your conclusions so that a bright researcher who is not necessarily a statistician can understand them.

Project 1: The data for come from Degen *et al.* 1998 and were used in the study of Freckleton *et al.* 2002. The body mass and average dietary metabolic rate was measured in numerous species of rodents for which a phylogeny is given. The rodents are separated into two groups: desert and non-desert species. Consider a mixed model for the two traits. Does a model that allows for separate means for desert versus non-desert species fit significantly better than the model that uses common means for the two habitats? Does a model that allows for separate means and separate variances for the two habitats fit the data significantly better than the model that allows for separate means but not separate variances?

References and Readings

[1] A A Degen, Kam M, Khokhlova I S, and Barraclough T G. Average daily metabolic rate in rodents: Habitat and dietary comparisosns. *Functional Ecology*, 12:63–73, 1998.

[2] R P Freckleton, Harvey P H, and Pagel M. Phylogenetic analysis and comparative data. *American Naturalist*, 160:712–726, 2002.

Chapter 2

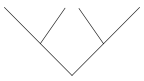
Techniques and Uses for Simulation

♢ 2.1 Asymptotic Theory and its Limitations

Asymptotic normal theory for maximum likelihood estimators essentially says that with independent, identically distributed (i.i.d.) samples, as the sample size increases, the point estimates are asymptotically unbiased and the variance-covariance matrix is given by the inverse of the Hessian of the negative log likelihood function.

For asymptotic theory to be useful, the samples need not be exactly independent, but too much dependence can be a problem. Additionally, the true parameter value must be in the interior of parameter space and the log likelihood function needs to be smooth (for instance, it needs to have continuous second derivatives!). Further, there is the requirement that the support of the likelihood function does not depend on the parameters. A typical counterexample to this condition is the uniform $[0, \theta]$ distribution.

Example 1 Suppose we have a model $y \sim N(\mu, \sigma^2 \mathbf{G}_1)$ where $\mathbf{G}_1$ is the relationship matrix given by the phylogeny:  and $\mathbf{G}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Suppose we can only increase the sample size by dividing the tips. That is, for a sample of size 4, the phylogeny is:  and $\mathbf{G}_2 = \begin{pmatrix} 1 & 1/2 & 0 & 0 \\ 1/2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1/2 \\ 0 & 0 & 1/2 & 1 \end{pmatrix}$

For a sample of size 8, the phylogeny is:  and

$$\mathbf{G}_3 = \begin{pmatrix} 1 & 3/4 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 3/4 & 1 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 1 & 3/4 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 3/4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 3/4 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 3/4 & 1 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 1 & 3/4 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 3/4 & 1 \end{pmatrix}$$

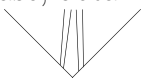
The maximum likelihood estimate for the mean is

$$\hat{\mu} = \frac{\mathbf{1}'\mathbf{G}^{-1}\mathbf{y}}{\mathbf{1}'\mathbf{G}^{-1}\mathbf{1}}$$

and the variance in this estimate is

$$\text{var}(\hat{\mu}) = \frac{\text{var}(\mathbf{1}'\mathbf{G}^{-1}\mathbf{y})}{(\mathbf{1}'\mathbf{G}^{-1}\mathbf{1})^2} = \frac{\sigma^2}{\mathbf{1}'\mathbf{G}^{-1}\mathbf{1}}$$

Thus, $\mathbf{1}'\mathbf{G}^{-1}\mathbf{1}$ is the "effective sample size" fir the estimator $\hat{\mu}$. In this case, no matter how often we divide the tips, the effective sample size never exceeds 3.

Moral: More data does not necessarily make asymptotic theory valid. The data have to be i.i.d. or "effectively" i.i.d. In this case, obtaining more independent samples would involve sampling closer to the root of the phylogeny:  ■

Example 2 The sample size requirements for asymptotic normality to hold can be quite large if there are a large number of parameters in the model. Example we will consider is if the heritability in the data for your first project is .8

■

Example 3 When the maximum likelihood estimate is too close to the boundary or on the boundary, the theory fails because you suspect the true, unknown, parameter to be on the boundary. The example is for the variables in your first project.

■

References and Readings

Conceptual Exercises for Section 2.1

1. Use simulations to determine the level of significance actually achieved by the t-test with data that is highly non-normal. For instance, consider samples of size 10 from an exponential distribution. What happens as the sample size is increased to 20? 30?

♢ 2.2 Non-Parametric Bootstrapping

Suppose you have a sample of size n from a completely unknown distribution function F. Call the sample $\mathbf{X} = (X_1, X_2, \ldots, X_n)$. Give each element of the sample probability $1/n$ and call this empirical distribution $\hat{F}$. A single bootstrap sample is a sample of size n taken from $\hat{F}$, or, equivalently, a sample of size n from $(X_1, X_2, \ldots, X_n)$ with replacement. The bootstrap sample is typically labeled with stars: $\mathbf{X}^* = (X_1^*, X_2^*, \ldots, X_n^*)$ and its empirical distribution function is labeled $\hat{F}^*$. The distribution of $\mathbf{X}^*$ given $\hat{F}$ serves as an approximation to the distribution of $\mathbf{X}$ given the unknown distribution F. (Note: the data may be univariate or multivariate. That is, X_i may be an element of R^d.)

Let $\theta = \theta(F)$ be a parameter of interest (a mean, a variance, a correlation, a ratio of means, etc...) and let $\hat{\theta} = \hat{\theta}(\hat{F})$ be an estimate of θ taken from the sample (a sample mean, a sample variance, a sample correlation, etc...). (Note that we are implicitly assuming that the order in which the sample is drawn is irrelevant when calculating the statistic.) Consider $Q(\hat{F}, F) = \hat{\theta}(\hat{F}) - \theta(F)$. We are interested in approximating the distribution of $Q(\hat{F}, F)$ by $Q(\hat{F}^*, \hat{F})$ which can be determined via repeated bootstrap simulation.

We will give a semi-rigorous justification of the bootstrap in a special case where the unknown distribution function F is a finite discrete distribution taking on values $x_1, x_2, \ldots, x_L$ with probability $\mathbf{p} = (p_1, p_2, \ldots, p_L)$. A random sample of size n from F has distribution $\hat{\mathbf{p}} = (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_L)$ where $\hat{p}_l$ is the number of times x_l appears in the sample divided by the sample size, n. Thus $Q(\hat{F}, F) = Q(\hat{\mathbf{p}}, \mathbf{p})$. Similarly, for a bootstrap sample, let $\hat{\mathbf{p}}^*$ represent the proportion of times each element appears in the bootstrapped sample.

We will show that, under certain regularity conditions, the asymptotic distribution of $Q(\hat{\mathbf{p}}, \mathbf{p})$ given $\mathbf{p}$ is the same as the asymptotic distribution of $Q(\hat{\mathbf{p}}^*, \hat{\mathbf{p}})$ given $\hat{\mathbf{p}}$.

Assume $Q(\mathbf{p}, \mathbf{p}) = 0$ for all $\mathbf{p}$ (equivalently, $\hat{\theta} = \theta$). Let

$$[\mathbf{u}(\mathbf{r}, \mathbf{q})]_{(l)} = \frac{\partial Q(\mathbf{r}, \mathbf{q})}{\partial r_l}$$

Assume $\mathbf{u}(\mathbf{r}, \mathbf{q})$ exists continuously in an open neighborhood of $(\mathbf{p}, \mathbf{p})$ and that $\mathbf{u}(\mathbf{p}, \mathbf{p}) \neq \mathbf{0}$. Then, by Taylor series expansions,

$$\begin{aligned} Q(\hat{\mathbf{p}}, \mathbf{p}) &= Q(\mathbf{p}, \mathbf{p}) + (\hat{\mathbf{p}} - \mathbf{p})(\mathbf{u}(\mathbf{p}, \mathbf{p}) + \boldsymbol{\epsilon}_n) = (\hat{\mathbf{p}} - \mathbf{p})(\mathbf{u}(\mathbf{p}, \mathbf{p}) + \boldsymbol{\epsilon}_n) \\ Q(\hat{\mathbf{p}}^*, \hat{\mathbf{p}}) &= Q(\hat{\mathbf{p}}, \hat{\mathbf{p}}) + (\hat{\mathbf{p}}^* - \hat{\mathbf{p}})(\mathbf{u}(\hat{\mathbf{p}}, \hat{\mathbf{p}}) + \hat{\boldsymbol{\delta}}_n) = (\hat{\mathbf{p}}^* - \hat{\mathbf{p}})(\mathbf{u}(\mathbf{p}, \mathbf{p}) + \hat{\boldsymbol{\epsilon}}_n) \end{aligned}$$

where $\boldsymbol{\epsilon}_n$, $\hat{\boldsymbol{\delta}}_n$, and $\hat{\boldsymbol{\epsilon}}_n$ all go to zero with probability 1. We have also used the fact that $\hat{\mathbf{p}}$ converges to $\mathbf{p}$ with probability 1.

Note that the distribution of $\hat{\mathbf{p}}$ given $\mathbf{p}$ is multinomial with sample size n and probability vector $\mathbf{p}$. Similarly, the distribution of $\hat{\mathbf{p}}^*$ given $\hat{\mathbf{p}}$ is multinomial with sample size n and probability vector $\hat{\mathbf{p}}$. Since $\hat{\mathbf{p}} \to \mathbf{p}$ with probability 1,

$$\sqrt{n}(\hat{\mathbf{p}} - \mathbf{p})|\mathbf{p} \to N(\mathbf{0}, \Sigma_{\mathbf{p}})$$

and

$$\sqrt{n}(\hat{\mathbf{p}}^* - \hat{\mathbf{p}})|\hat{\mathbf{p}} \to N(\mathbf{0}, \Sigma_{\mathbf{p}})$$

where $\Sigma_{\mathbf{p}}$ has (l,m) entry $p_l(\delta_{l,m}-p_m)$. Thus, both $\sqrt{n}Q(\hat{\mathbf{p}},\mathbf{p})$ and $\sqrt{n}Q(\hat{\mathbf{p}}^*,\hat{\mathbf{p}})$ converge to a normal mean 0 variance $\mathbf{u}'(\mathbf{p},\mathbf{p})\Sigma_{\mathbf{p}}\mathbf{u}(\mathbf{p},\mathbf{p})$.

One can always view the bootstrapped distribution as a multinomial distribution as in the above example, but the distribution of the original data, of course, cannot be viewed as such in general. The usual way to show that bootstrapping is a good method, even an excellent method, is to use Edgeworth expansions and their Cornish-Fisher inverses. We will discuss these without too much detail below, but first we need some definitions:

A *confidence point* $\hat{\theta}(\alpha)$ for a parameter θ is a method for determining the α-percentile for the distribution of $\hat{\theta}$. The method may or may not work. For example, if you have a sample from a normal distribution, you compute confidence point for the mean as $\bar{X}+t_\alpha(n-1)\hat{\sigma}/\sqrt{n}$ where $t_\alpha(n-1)$ is the α-percentile for the t-distribution with $(n-1)$ degrees of freedom. This confidence point is very good if the data is normal, quite good in most circumstances, but can be very inaccurate if the data is non-normal enough.

There are natural ways of computing confidence points using the bootstrapping principle. The *percentile* bootstrap method is as follows: For each bootstrapped resample i, compute the statistic for that sample $\hat{\theta}_i^*$. Order the bootstrapped values (in the order of smallest to largest). The ordered statistics are: $\hat{\theta}^*_{(1)},\hat{\theta}^*_{(2)},\ldots,\hat{\theta}^*_{(B)}$. The α-percentile estimate is $\hat{\theta}^*_{([\alpha B])+1}$ for $\alpha<1$.

The *percentile-t* bootstrap method is based on the statistical notion of pivoting. When you subtract the mean and divide by the variance you are making your statistic (almost) scale-free. The methods is: for each bootstrapped resample i, compute the statistic for that sample $\hat{t}_i=\frac{\hat{\theta}_i^*-\hat{\theta}}{\hat{\sigma}_i/\sqrt{n}}$. Order the bootstrapped values (in the order of smallest to largest). The ordered statistics are: $\hat{t}^*_{(1)},\hat{t}^*_{(2)},\ldots,\hat{t}^*_{(B)}$. The α-percentile estimate is $\hat{t}^*_{([\alpha B])+1}$ for $\alpha<1$. The percentile-t confidence point is then $\bar{X}+\hat{t}^*_{([\alpha B])+1}\hat{\sigma}/\sqrt{n}$

We have described three methods of determining confidence points. The problem is that we do not know if the confidence points we get from these methods are any good. There are two ways of defining "goodness."

The *accuracy* of a confidence point $\hat{\theta}(\alpha)$ for a parameter θ refers to how far from the nominal level of coverage the confidence level falls. That is, the accuracy of $\hat{\theta}(\alpha$ for α-coverage is $|P(\theta<\hat{\theta}(\alpha))-\alpha|$. First order accuracy means that the accuracy is $O(1/\sqrt{n})$ and second order accuracy is $O(1/n)$ where n is the sample size.

The *correctness* of a confidence point $\hat{\theta}(\alpha)$ for a parameter θ refers to how how far from some ideal value the confidence point lies. That is, if $\hat{\theta}_{\text{exact}}(\alpha)$ is the ideal confidence point, the correctness of $\hat{\theta}(\alpha$ is $|\hat{\theta}_{\text{exact}}(\alpha)-\hat{\theta}(\alpha|$. A confidence point is first order correct if the correctness is $O_p(1/\sqrt{n})\hat{\sigma}$ and second order correct if it is $O_p(1/n)\hat{\sigma}$ where $\hat{\sigma}$ is a measure of the standard error of $\hat{\theta}$ and is typically itself on the order of $O(1/\sqrt{n})$. The order of correctness implies the same order in accuracy. We will see why below.

An Edgeworth expansion is a series expansion of a cumulative distribution function using the normal distribution. That is, if S_n is a standardized statistic (either $\frac{\hat{\theta}-\theta}{\sigma}$ or $\frac{\hat{\theta}-\theta}{\hat{\sigma}}$) based on an i.i.d. sample of size n (under various regularity conditions given below)

$$P(S_n < x) = \Phi(x) + \frac{1}{\sqrt{n}} p_1(x)\phi(x) + \frac{1}{n} p_2(x)\phi(x) + O(\frac{1}{n^{3/2}})$$

where Φ is the cumulative normal distribution, ϕ is the normal density, $p_1(x)$ is a first order polynomial correction for skewness, and $p_2(x)$ is a polynomial correction for kurtosis and the second-order effects of skewness. The Edgeworth expansion can continue into an infinite sum, but it often fails to converge and terms past the first few given above are rarely used. The expansion above is useful in explaining the properties of the bootstrap.

Edgeworth expansions come from considering the characteristic function for the random variable S_n:

$$\chi_n(t) = Ee^{itS_n} = \int_{-\infty}^{\infty} e^{itx} dP(S_n \leq x) = \exp\left(\kappa_{1,n}(it) + \frac{1}{2}\kappa_{2,n}(it)^2 + \dots\right) = 1 + E(S_n)(it) + E(S_n^2)(it)^2 + \dots$$

The coefficients, $\kappa_{j,n}$ are called the j^{th} cumulants. Equating terms in the series expansion we see that formulas for the first few cumulants are:

$$\kappa_{1,n} = E(S_n) \quad \kappa_{2,n} = \mathrm{Var}(S_n) \quad \kappa_{3,n} = E(S_n - ES_n)^3 \quad \kappa_{4,n} = E(S_n - ES_n)^4 - 3(\mathrm{var}(S_n))^2 \quad \text{etc...}$$

As a regularity condition, we assume that

$$\kappa_{j,n} = n^{-(j-2)/2}\left(k_{j,1} + \frac{k_{j,2}}{n} + \frac{k_{j,3}}{n^2} + \dots\right)$$

where $k_{1,1} = 0$ because of the centering and $k_{2,1} = 1$ because of the scaling.

Thus,

$$\begin{aligned}
\chi_n(t) &= \exp\left(-\frac{t^2}{2} + \frac{k_{1,2}(it) + \frac{1}{3!}k_{3,1}(it)^3}{\sqrt{n}} + \frac{\frac{1}{2!}k_{2,2}(it)^2 + \frac{1}{4!}k_{4,1}(it)^4}{n} + O\left(\frac{1}{n^{3/2}}\right)\right) \\
&= e^{-\frac{t^2}{2}}\left(1 + \frac{k_{1,2}(it) + \frac{1}{6}k_{3,1}(it)^3}{\sqrt{n}} + \frac{\frac{1}{2}k_{2,2}(it)^2 + \frac{1}{24}k_{4,1}(it)^4 + \frac{1}{2}[k_{1,2}(it) + \frac{1}{6}k_{3,1}(it)^3]^2}{n} + O\left(\frac{1}{n^{3/2}}\right)\right) \\
&= e^{-\frac{t^2}{2}}\left(1 + \frac{r_1(it)}{\sqrt{n}} + \frac{r_2(it)}{n} + O\left(\frac{1}{n^{3/2}}\right)\right)
\end{aligned}$$

To get the distribution function for S_n we need to invert the expansion to get

$$P(S_n \leq x) = \Phi(x) + \frac{R_1(x)}{\sqrt{n}} + \frac{R_2(x)}{n} + O\left(\frac{1}{n^{3/2}}\right)$$

where $R_j(x)$ is a function whose Fourier-Stiltjes transform is $e^{-t^2/2}r_j(it)$ for $j = 1, 2$. To determine $R_j(x)$, note that

$$e^{-t^2/2} = \int_{-\infty}^{\infty} e^{itx} d\Phi(x)$$

and that, by integration by parts,

$$e^{-t^2/2} = (-it)^{-j} \int_{-\infty}^{\infty} e^{itx} d\Phi^{(j)}(x)$$

. Thus,

$$(it)^j e^{-t^2/2} = \int_{-\infty}^{\infty} e^{itx} (-1)^j d\Phi^{(j)}(x)$$

. If we let D denote the derivative operator, this implies that

$$r_j(it)e^{-t^2/2} = \int_{-\infty}^{\infty} e^{itx} d\left(r_j(-D)\Phi(x)\right)$$

. Thus, $R_j(x) = r_j(-D)\Phi(x)$.

The derivatives of the cumulative distribution function for the standard normal distribution are given in terms of the Hermite polynomials.

$$(-d)^j \Phi(x) = -H_{j-1}(x)\phi(x)$$

where $\phi(x)$ is the density function for the standard normal and $H_0(x) = 1$, $H_1(x) = x$, $H_2(x) = x^2 - 1$, $H_3(x) = x(x^2 - 3)$, $H_4(x) = x^4 - 6x^2 + 3$, and $H_5(x) = x(x^4 - 10x^2 + 15)$.

Combining facts, we see that

$$p_1(x) = -(k_{1,2} + \frac{1}{6} k_{3,1}(x^2 - 1)$$

and

$$p_2(x) = -x \left(\frac{1}{2}(k_{2,2} + k_{1,2}^2) + \frac{1}{24}(k_{4,1} + 4k_{1,2}k_{3,1})(x^2 - 3) + \frac{1}{72} k_{3,1}^2 (x^4 - 10x^2 + 15) \right)$$

Now, what does this have to do with bootstrapping accuracy? The essential difference between the cumulative distribution function for the true population (S_n) and the cumulative distribution function for the bootstrapped distribution ($\hat{S}_n$) is that the true moments are replaced by their estimates from the data. That is, $k_{j,i}$ are replaced by their sample values, $\hat{k}_{j,i}$. The estimates are $O_p\left(\frac{1}{\sqrt{n}}\right)$ from their true values for large classes of practical problems by central limit theorem arguments (or, equivalently, as part of the regularity conditions). Thus, $\hat{S}_n$ is within $O\left(\frac{1}{n}\right)$ of S_n. For percentile-t confidence intervals, this implies second order accuracy. However, for naive, straight percentile, since we are calculating some version of $\hat{\sigma}$ for use in the confidence interval, we only have first order accuracy (we assumed the true variance was known for the result above, but it isn't and has to be estimated in some way to obtain a confidence interval.)

Since percentile-t intervals have second order accuracy, they appear to be better. However, there are many cases of strange behavior of such intervals. Further, they lack the property of being transformation-invariant. That is, if we modify our statistic, say, to be the square of the mean

rather than the mean itself, the percentile-t interval for the transformed statistic is not the transform applied to the endpoints of the interval for the original statistic.

There are ways of improving the naive percentile bootstrap method that makes it second-order accurate. These methods include iterating the bootstrap and the BC_a algorithm. These are the topics of the next section.

Let's end with one simple example to highlight the issues in this section.

Example 1 Suppose $X_i \sim N(\mu, \sigma^2)$. Let $\theta = \mu^2$, $\hat{\theta} = (\bar{X})^2$, $\Sigma = 2\mu\sigma$, and $S_n = \dfrac{(\bar{X})^2 - \mu^2}{\Sigma/\sqrt{n}}$. The Edgeworth expansion of S_n involves computing and decomposing the cumulants. For this, we will need the moments of the normal distribution: $EX = \mu$, $EX^2 = \mu^2 + \sigma^2$, $EX^3 = \mu(\mu^2 + 3\sigma^2)$, $EX^4 = \mu^4 + 6\mu^2\sigma^2 + 3\sigma^4$, $EX^5 = \mu(\mu^4 + 10\mu^2\sigma^2 + 15\sigma^4)$, and $EX^6 = \mu^6 + 15\mu^4\sigma^2 + 45\mu^2\sigma^4 + 15\sigma^6$. Note that

$$
\begin{aligned}
ES_n &= \frac{E(\bar{X})^2 - \mu^2}{\Sigma/\sqrt{n}} = \frac{\mu^2 + \sigma^2/n - \mu^2}{\mu\sigma/\sqrt{n}} = \frac{\sigma}{\sqrt{n}\mu} \\
\text{Var}(S_n) &= \text{Var}\left(\frac{\sqrt{n}(\bar{X})^2}{\Sigma}\right) = \frac{n}{\Sigma^2}(E\bar{X}^4 - (E\bar{X}^2)^2) \\
&= \frac{n}{\Sigma^2}(\mu^4 + 6\mu^2\sigma^2/n + 3\sigma^4/n^2 - (\mu^2 + \sigma^2/n)^2) \\
&= 1 + \frac{\sigma^2}{2n\mu^2} \\
E(S_n - ES_n)^3 &= E\left(\frac{\bar{X}^2 - \mu^2}{\Sigma/\sqrt{n}} - \frac{\sigma^2}{\Sigma\sqrt{n}}\right) \\
&= \frac{n^{3/2}}{\Sigma^3} E\left(\bar{X}^2 - (\mu^2 + \sigma^2/n)\right)^3 \\
&= \frac{n^{3/2}}{\Sigma^3} E\left(\bar{X}^6 - 3\bar{X}^4(\mu^2 + \sigma^2/n) + 3\bar{X}^2(\mu^2 + \sigma^2/n)^2 - (\mu^2 + \sigma^2/n)^3\right) \\
&= \frac{n^{3/2}}{\Sigma^3} E\left((\mu^6 + 15\mu^4\sigma^2/n + 45\mu^2\sigma^4/n^2 + 15\sigma^6/n^3)\right. \\
&\qquad -3(\mu^4 + 6\mu^2\sigma^2/n + 3\sigma^4/n^2)(\mu^2 + \sigma^2/n) \\
&\qquad \left. +3(\mu^2 + \sigma^2/n)^3 - (\mu^2 + \sigma^2/n)^3\right) \\
&= \frac{n^{3/2}}{\Sigma^3}\left(\frac{8\sigma^6}{n^3} + \frac{24\mu^2\sigma^4}{n^2}\right) \\
&= \frac{3\sigma}{\mu n^{1/2}} + \frac{\sigma^3}{\mu^3 n^{3/2}}
\end{aligned}
$$

We skip the fourth moments of S_n. In general, smooth, normalized, functions of means satisfy the regularity conditions needed for the Edgeworth expansions to work as claimed.

References and Readings

[1] B Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7:1–26, 1979.

[2] B Efron and Tibshirani R J. *An introduction to the bootstrap*. Chapman Hall, New York, 1993.

[3] P Hall. *The Bootstrap and Edgeworth Expansion*. Springer-Verlag, New York, 1992.

Conceptual Exercises for Section 2.2

1. Use the Edgeworth expansion formula to show that is you know a confidence point up to a first or second order correctness, then you also know it up to first or second order accuracy, respectively.

♢ 2.3 Better Bootstrapping Methods

We would like to preserve the transformation-invariant properties of the naive percentile bootstrap but also obtain second order accuracy. This goal can be accomplished in two ways: iterating the bootstrap and with a bias-correction with acceleration algorithm (BC_a.).

Iterating the Bootstrap

One way to achieve better accuracy is to iterate the bootstrap. How does that work in practice?

Example 1 Suppose we want to provide a confidence interval for the mean of a highly skewed distribution. There are several different obvious methods of modifying the naive bootstrap confidence interval. The following imposes an additive correction. Let $\hat{\theta}(\beta)$ be the β-percentile of the distribution of $\hat{\theta}$. In general, this is unknown and we estimate via bootstrapping as $\hat{\theta}^*(\beta)$ (the β-percentile of the estimates obtained from the bootstrapped samples). Of course, this estimate may not achieve nominal coverage - it may be off by $O(\frac{1}{\sqrt{n}})$ with, possibly, some large constant. That is,

$$P\left(\theta \in (-\infty, \hat{\theta}^*(1-\alpha)]\right) \neq \alpha$$

Consider the modified interval: $(-\infty, \hat{\theta}^*(1-\alpha)+t]$. We want to estimate t so that the interval achieves (or comes closer to achieving) nominal coverage. So we estimate t via the bootstrap. That is, we find $\hat{t}$ so that

$$P\left(\hat{\theta} \in (-\infty, \hat{\theta}^{**}(1-\alpha)+\hat{t}]\right) = \alpha$$

A picture of this process is:

$$\hat{\theta} \text{ from } X = \left\{ \begin{array}{l} \hat{\theta}_1^* \text{ from } X_1^* \left\{ \begin{array}{c} \hat{\theta}_1^{**}(1) \text{ from } X_1^{**}(1) \\ \hat{\theta}_1^{**}(2) \text{ from } X_1^{**}(2) \\ \vdots \\ \hat{\theta}_1^{**}(B) \text{ from } X_1^{**}(B) \end{array} \right\} \hat{\theta}_1^{**}(1-\alpha) \\ \\ \hat{\theta}_2^* \text{ from } X_2^* \left\{ \begin{array}{c} \hat{\theta}_2^{**}(1) \text{ from } X_2^{**}(1) \\ \hat{\theta}_2^{**}(2) \text{ from } X_2^{**}(2) \\ \vdots \\ \hat{\theta}_2^{**}(B) \text{ from } X_2^{**}(B) \end{array} \right\} \hat{\theta}_2^{**}(1-\alpha) \\ \\ \vdots \\ \\ \hat{\theta}_B^* \text{ from } X_B^* \left\{ \begin{array}{c} \hat{\theta}_B^{**}(1) \text{ from } X_B^{**}(1) \\ \hat{\theta}_B^{**}(2) \text{ from } X_B^{**}(2) \\ \vdots \\ \hat{\theta}_B^{**}(B) \text{ from } X_B^{**}(B) \end{array} \right\} \hat{\theta}_B^{**}(1-\alpha) \end{array} \right.$$

Now, find the $\hat{t}$ adjustment that makes $\hat{\theta}$ below $\hat{\theta}^{**}(1-\alpha)+\hat{t}$ $100(1-\alpha)\%$ of the time. ■

Iterating the bootstrap increases the accuracy by a factor of $O(\frac{1}{\sqrt{n}})$. If the confidence interval is two sided, the accuracy is actually increased by a factor of $O(\frac{1}{n})$. The formalism for a proof of the first statement is as follows:

Let $Q_t(\hat{F}, F) = I(\theta < \hat{\theta} + t) - (1 - \alpha)$. To find a $100(1 - \alpha)\%$ one-sided confidence interval for θ, we find t so that $E(Q_t(\hat{F}, F)|F) = 0$. Since we do not know F, the bootstrapping principle says that we find $t = T(\hat{F})$ so that $E(Q_t(\hat{F}^*, \hat{F})|\hat{F}) = 0$. From the last section, we know we can do this with accuracy $O_p(\frac{1}{\sqrt{n}})$. That is,

$$E(Q_{T(\hat{F})}(\hat{F}, F)|F) = \frac{c(F)}{\sqrt{n}} + O(\frac{1}{n})$$

The iterating principle creates a new function $T(\cdot, u) = T(\cdot) + u$. (This function could (naturally) be multiplicative instead of linear $(1 + u)T(\cdot)$ or it could take the $[(1 - \alpha) + u]$-percentile rather than the $(1 - \alpha)$-percentile plus u.) We now want to solve the equation

$$E(Q_{T(\hat{F},u)}(\hat{F}, F)|F) = 0$$

which we do via the (iterated) bootstrap as illustrated above.

This method improves the order of accuracy by a factor of $O(\frac{1}{\sqrt{n}})$. Why? Provided that $\frac{d}{du}E(Q_{T(\hat{F},u)}(\hat{F}, F)|F)|_{u=0}$ is asymptotic to a non-zero constant $(d(F))$as $n \to \infty$, the Taylor series expansion of $E(Q_{T(\hat{F},u)}(\hat{F}, F)|F)$ about $u = 0$ gives

$$E(Q_{T(\hat{F},u)}(\hat{F}, F)|F) = \frac{c(F)}{\sqrt{n}} + d(F)u + O(\frac{1}{n} + u^2)$$

The solution to $E(Q_{T(\hat{F},u)}(\hat{F}, F)|F) = 0$ is

$$u(F) = -\frac{c(F)}{\sqrt{n}d(F)} + O(\frac{1}{n})$$

Similarly, the solution with F replaced by the empirical distribution $\hat{F}$ is

$$u(\hat{F}) = -\frac{c(\hat{F})}{\sqrt{n}d(\hat{F})} + O(\frac{1}{n})$$

But under mild regularity conditions (c and d are continuous functionals),

$$c(\hat{F}) = c(F) + O_p(\frac{1}{\sqrt{n}}) \quad \text{and} \quad d(\hat{F}) = d(F) + O_p(\frac{1}{\sqrt{n}})$$

Thus, $u(\hat{F}) = u(F) + O_p(\frac{1}{n})$ and so

$$E(Q_{T(\hat{F},u(\hat{F}))}(\hat{F}, F)|F) = E(Q_{T(\hat{F},u(F))}(\hat{F}, F)|F) + O(\frac{1}{n}) = O(\frac{1}{n})$$

demonstrating that the accuracy of the interval has improved from $O(\frac{1}{\sqrt{n}})$ to $O(\frac{1}{n})$.

A Correction Algorithm

We will describe the BC_a algorithm developed by Bradley Efron for improving the accuracy of bootstrap confidence intervals while preserving the transformation-invariance properties of the naive bootstrap intervals. The proof of the second order accuracy and the formula for estimating the acceleration constant a rely on the Edgeworth expansion technique and the multinomial distribution approximation (supported on the observed sample values) to the underlying distribution.

The BC_a algorithm chooses as endpoints for the confidence interval $\hat{\theta}(\alpha_1)$ and $\hat{\theta}(\alpha_2)$ (percentiles of the bootstrapped estimates of θ) where α_1 and α_2 are given by the following formulas:

$$\alpha_1 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z(\alpha)}{1 - \hat{a}(\hat{z}_0 + z(\alpha))}\right) \quad \text{and} \quad \alpha_2 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z(1-\alpha)}{1 - \hat{a}(\hat{z}_0 + z(1-\alpha))}\right)$$

where Φ is the cumulative distribution function for the standard normal distribution, $z(\alpha)$ is the α-quantile of the standard normal distribution, $\hat{z}_0$ is an estimate of the bias (described below), and $\hat{a}$ is a factor called the acceleration and is the rate of change in the standard error of $\hat{\theta}$ with respect to the normalized value of θ.

The estimate of the bias based on B bootstrapped distributions is

$$\hat{z}_0 = \Phi^{-1}\left(\frac{\#\{\hat{\theta}^* < \hat{\theta}\}}{B}\right)$$

The estimate of the acceleration can be shown to be given by the following jackknife-based formula

$$\hat{a} = \frac{\sum_{i=1}^{n}(\hat{\theta}_J - \hat{\theta}_{J_i})^3}{6\left(\sum_{i=1}^{n}(\hat{\theta}_J - \hat{\theta}_{J_i})^2\right)^{3/2}}$$

where $\hat{\theta}_{J_i}$ is the estimate of θ obtained by leaving the i^{th} point out of the sample and $\hat{\theta}_J$ is the average of these jackknifed estimates.

References and Readings

Conceptual Exercises for Section 2.3

1. Use the iterative bootstrap to give an approximate 95% confidence interval for the mean of the distribution from which the following data were collected: $\{5.8, 0.98, 48.7, 1.78, 6.77, 25.3, 1.8, 2.1, 10.7, 1.42, 4.77, 7.0, 0.015, 6.24, 6.9\}$.

♢ 2.4 Bootstrapping Examples and Counter-Examples

We will demonstrate the bootstrapping method as applied to several problems and also discuss some examples where the results of bootstrapping are poor.

Example 1 The mean of a normal distribution. To assess the accuracy of the bootstrap, we have to simulate many samples from the normal distribution, obtain a confidence intervals for the mean from bootstrapping each sample, and consider how many of these intervals cover the true mean. In this example, all reasonable methods work well. Indeed, the coverage obtained from 1000 samples each of size 100 taken from a $N(10, 2)$ distribution shows that the mean is covered 94.8% of the time (essentially, dead on target.) ■

Example 2 The mean of a gamma distribution with shape $\alpha = .3$ and rate $\lambda = .1$. Again, the accuracy of the bootstrap is assessed by bootstrapping the confidence interval for 1000 samples from this distribution. The intervals obtained from the naive percentile method cover the mean (3) xx% of the time and the intervals obtained from the BCa method cover the mean yy% of the time. ■

Example 3 There are obvious problems where bootstrapping is unlikely to be of much use. Suppose we have a distribution on some interval $[0, \theta]$ and we want to estimate θ. The only logical estimate we have is the maximum of the data we observe. But bootstrapping cannot provide information about how much larger θ should be. We need to collect additional data for that problem and/or develop an estimate based on a parametric model of the distribution. ■

Example 4 An example where bootstrapping gives biased estimates (in some sense) is the following. Suppose we want to know whether the mean of a distribution is less that a fixed value, μ_0 or not. We could form the statistic: $I(\hat{\mu} < \mu_0)$. For the original sample, this statistic will be 0 or 1. To determine the probability the mean is really less that μ_0 (the expected value of this statistic), we can bootstrap the original data, compute the bootstrapped statistic on each bootstrapped data set, and average the results. If we do this for samples of size 100 from $N(10, 2)$, the probability $\hat{\mu}$ is less than 10.2 is 0.83 and the bootstrapped estimate is 0.76. Indeed, for probabilities over 50%, the bootstrap will tend to underestimate them. Note that the statistic is certainly not converging to a nice normal distribution. ■

Example 5 Bootstrapping is can approximate the permutation test and can be adapted to handle more complicated testing situations. For example, consider two samples $\{y_1, y_2, \ldots, y_{n_y}\}$ and $\{z_1, z_2, \ldots, z_{n_y}\}$. If we are interested in whether the means of the two populations are equal, and if we are willing to test whether the two population distribution functions are equal, then we can simply permute the combined data between the two categories to get a permutation test. Bootstrapping can approximate this permutation test by selecting B permutations at random and using those instead. If we use a standard t-test, then the original statistic was

$$t = \frac{\bar{y} - \bar{z}}{\hat{\sigma}\sqrt{1/n_y + 1/n_z}}$$

where $\hat{\sigma}$ is the pooled standard deviation estimate. For each bootstrapped permutation of the original data we compute

$$t^* = \frac{\bar{y}^* - \bar{z}^*}{\hat{\sigma}^*\sqrt{1/n_y + 1/n_z}}$$

The approximate p-value is $\frac{\#\{t^* > t\}}{B}$.

However, if the two populations do not have equal variances then their members are not exchangeable and the permutation test is invalid. However, bootstrapping provides an alternative method. Let $\bar{x}$ be an estimate of the mean of the combined data set. Consider $\tilde{y}_i = y_i - \bar{y} + \bar{x}$ and $\tilde{z}_i = z_i - \bar{z} + \bar{z}$. Then $\tilde{y}$ and $\tilde{z}$ have equal means by construction and their variances are the same as the original two samples. We resample n_y values with replacement from y and n_z values with replacement from z. If we consider the usual unequal-variances t-statistic, the statistic for the original data is

$$t = \frac{\bar{y} - \bar{z}}{\sqrt{\hat{\sigma}_y^2/n_y + \hat{\sigma}_z^2/n_z}}$$

and the statistic for each bootstrapped data set is:

$$t^* = \frac{\bar{\tilde{y}}^* - \bar{\tilde{z}}^*}{\sqrt{\hat{\sigma}_{\tilde{y}^*}^2/n_y + \hat{\sigma}_{\tilde{z}^*}^2/n_z}}$$

The approximate p-value is $\frac{\#\{t^* > t\}}{B}$. ■

References and Readings

Conceptual Exercises for Section 2.4

1.

♢ 2.5 Parametric Bootstrapping

We will consider various bootstrapping techniques and explore the advantages of parametric bootstrapping over normal approximations for estimating the shape and rate parameters of a Gamma distribution.

Recall that the density of the Gamma distribution is:

$$f(x) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x}$$

for $x > 0$ where $\alpha > 0$ is the shape parameter and $\lambda > 0$ is the rate parameter.

Given data $X_1, X_2, \ldots, X_n$ from such a distribution, we can approximate the parameter α and λ in various ways. Using the fact that the mean of a Gamma distribution is α/λ and the variance is α/λ^2 we can use the method of moments estimates $\hat{\lambda} = \dfrac{\bar{X}}{\hat{\sigma}^2}$ and $\hat{\alpha} = \left(\dfrac{\bar{X}}{\hat{\sigma}}\right)^2$.

We can consider the maximum likelihood estimates as well. The maximum likelihood estimate for α can not be determined explicitly - it must be found numerically. The likelihood function of the parameters given the data is:

$$L(\lambda, \alpha | X_1, X_2, \ldots, X_n) = \frac{\lambda^{n\alpha}}{(\Gamma(\alpha))^n} \left(\prod X_i\right)^{\alpha-1} e^{-\lambda \sum X_i}$$

and the log likelihood function is:

$$\log L = n\alpha \log \lambda - n \log \Gamma(\alpha) + (\alpha - 1) \sum \log X_i - \lambda \sum X_i$$

Taking both partial derivatives with respect to the parameters, setting them equal to zero, and solving yields:

$$\begin{aligned} \frac{\partial log L}{\partial \lambda} &= \frac{n\alpha}{\lambda} - \sum X_i = 0, \qquad \hat{\lambda} = \frac{n\hat{\alpha}}{\sum X_i} \\ \frac{\partial log L}{\partial \alpha} &= n \log \lambda - n \frac{\Gamma'(\alpha)}{\Gamma(\alpha)} + \sum \log X_i = n \log \frac{n\alpha}{\sum X_i} - n \frac{\Gamma'(\alpha)}{\Gamma(\alpha)} + \sum \log X_i = 0 \end{aligned}$$

The estimate $\hat{\alpha}$ has to be found by numerically finding the root of the last equation.

Let's consider the performance of various possible 95% confidence intervals for the shape and scale parameters when the true distribution has shape $\alpha = 0.2$ and scale $\lambda = 1$.

Example 1 The confidence interval for the shape parameter α obtained by asymptotic normality has the form $(\hat{\alpha} - 1.96\hat{\sigma}_\alpha, \hat{\alpha} + 1.96\hat{\sigma}_\alpha)$ where $\hat{\alpha}$ is the maximum likelihood estimate for the shape parameter and $\hat{\sigma}_\alpha$ is its estimated standard deviation which is approximated using the inverse of the Hessian of the negative log likelihood function:

$$\mathrm{Var}(\alpha, \lambda) \approx \left(- \begin{bmatrix} \frac{\partial^2 \log L}{\partial \alpha^2} & \frac{\partial^2 \log L}{\partial \alpha \partial \lambda} \\ \frac{\partial^2 \log L}{\partial \alpha \partial \lambda} & \frac{\partial^2 \log L}{\partial \lambda^2} \end{bmatrix} \right)^{-1}$$

The estimate for the variance in the shape is the upper left hand entry in this matrix. The estimate for the covariance between the shape and scale parameters are the off-diagonal entries and the estimate for the variance of the scale parameter is the lower right hand entry.

For samples of size 10 from a Gamma distribution with true shape 0.2 and true scale 1, the confidence interval obtained this way covers the true shape parameter 97% of the time, so this confidence interval is a little conservative.

Similarly, the confidence interval for the scale parameter has the form $(\hat{\lambda} - 1.96\hat{\sigma}_\lambda, \hat{\lambda} + 1.96\hat{\sigma}_\lambda)$. In this example, the confidence interval obtained assuming normality covers the true scale parameter $\lambda = 1$ 100% of the time, which is extremely conservative. ■

Example 2 The confidence interval obtained by the naive parametric percentile method is determined as follows. For a given sample X, find the maximum likelihood estimates for α ($\hat{\alpha}$) and λ ($\hat{\lambda}$). Let n be the sample size of X. Draw a large number B of samples of size n from a Gamma distribution with shape $\hat{\alpha}$ and scale $\hat{\lambda}$. For each of these bootstrapped samples, find the maximum likelihood estimates for α ($\hat{\alpha}^*$) and λ ($\hat{\lambda}^*$. Order these estimates and remove the 2.5 and 97.5 percentiles as the confidence interval for α and λ, respectively.

For samples of size 10 from a Gamma distribution with true shape 0.2 and true scale 1, the confidence interval for *alpha* obtained this way covers the true shape parameter 86% of the time and the confidence interval for λ obtained this way covers the true scale parameter 79% of the time. These are grossly ant-conservative coverage levels. ■

Example 3 One reason that the naive percentile interval performs so poorly is because the distribution of $\hat{\alpha}$ is skewed to the right. One idea to correct for this is to use the quantity $p = \hat{\alpha} - \alpha$ with the hopes that this quantity does not depend on α. This method is called the "basic" or "standard" bootstrap method. If we knew the 2.5 and 97.5 percentiles of p, our confidence interval for α would be $(\hat{\alpha} - p_{0.975}, \hat{\alpha} - p_{0.025}) = (2\hat{\alpha} - \hat{\alpha}^*(.25), 2\hat{\alpha} - \hat{\alpha}^*(.975))$. Note that if the distribution of $\hat{\alpha}$ is skewed to the right, the interval here is skewed to the left (in the opposite direction) whereas the naive percentile interval is skewed right.

For samples of size 10 from a Gamma distribution with true shape 0.2 and true scale 1, the confidence interval obtained using pivoting covers the true shape parameter 92% of the time and the true scale parameter 86.5% of the time. (Note: these are the "basic" bootstrap confidence intervals in R.) ■

Example 4 The confidence interval obtained by the parametric percentile-t method is determined as follows. For a given sample X, find the maximum likelihood estimates for α ($\hat{\alpha}$) and λ ($\hat{\lambda}$). Also determine $\hat{\sigma}_\alpha$ and $\hat{\sigma}_\lambda$ using the Hessian of the log likelihood function as described above. The confidence intervals will have the form $(\hat{\alpha}+t^*_{0.025}\hat{\sigma}_\alpha, \hat{\alpha}+t^*_{0.975}\hat{\sigma}_\alpha)$ and $(\hat{\lambda}+t^*_{0.025}\hat{\sigma}_\lambda, \hat{\lambda}+t^*_{0.975}\hat{\sigma}_\lambda)$ where the distribution of t^* is determined by the studentized results of the bootstrap for α and λ, respectively.

Let n be the sample size of X. Draw a large number B of samples of size n from a Gamma distribution with shape $\hat{\alpha}$ and scale $\hat{\lambda}$. For each of these bootstrapped samples, find the maximum likelihood estimates for α ($\hat{\alpha}^*$) and λ ($\hat{\lambda}^*$) and estimate the variances using the Hessian of the log likelihood function with the bootstrapped data. Compute $t^* = \dfrac{\hat{\alpha}^* - \hat{\alpha}}{\hat{\sigma}^*_\alpha}$ which is approximately pivotal - that is, it does not depend on $\hat{\alpha}$. Order these estimates and remove the 2.5 and 97.5 percentiles for

the confidence interval for α. Compute $t^* = \dfrac{\hat{\lambda}^* - \hat{\lambda}}{\hat{\sigma}^*_\lambda}$. Order these estimates and remove the 2.5 and 97.5 percentiles for the confidence interval for λ.

For samples of size 10 from a Gamma distribution with true shape 0.2 and true scale 1, the confidence interval obtained this way covers the true shape parameter 94% of the time and the confidence interval for the scale parameter covers the true scale parameter 97% of the time. These are pretty accurate coverage levels given our small sample size. ■

Example 5 The confidence interval obtained by the BC_a method is determined as follows: For a given sample X, find the maximum likelihood estimates for α ($\hat{\alpha}$) and λ ($\hat{\lambda}$). Let n be the sample size and let $\hat{\alpha}^*$ be the (ordered) B bootstrapped re-estimates of α obtained from samples of size n drawn from a Gamma distribution with shape $\hat{\alpha}$ and scale $\hat{\lambda}$. The BC_a method gives a confidence interval of the form $(\hat{\alpha}^*(\beta_1), \hat{\alpha}^*(\beta_2))$ where

$$\beta_1 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z(0.025)}{1 - \hat{a}(\hat{z}_0 + z(0.025))}\right)$$

$$\beta_2 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z(0.975)}{1 - \hat{a}(\hat{z}_0 + z(0.975))}\right)$$

where

$$\hat{z}_0 = \Phi^{-1}\left(\frac{\#(\hat{\alpha}^* < \hat{\alpha})}{B}\right)$$

and

$$\hat{a} = \frac{\sum(\hat{\alpha}_J - \hat{\alpha}_{J_i})^3}{6(\sum(\hat{\alpha}_J - \hat{\alpha}_{J_i})^2)^{3/2}}$$

where $\hat{\alpha}_{J_i}$ is the jackknife estimate obtained by omitting data point i and $\hat{\alpha}_J$ is the average of the jackknife estimates. The confidence interval for λ is determined similarly.

For samples of size 10 from a Gamma distribution with true shape 0.2 and true scale 1, the confidence interval obtained this way covers the true shape parameter 94% of the time and the interval for the scale parameter covers the true scale parameter 92% of the time. ■

Which intervals are to be preferred? One cannot base such a broad question on the outcome of the results for one small sample size taken from one particular distribution. However, one also cannot expect the basic bootstrap and the naive percentile bootstrap to perform well with small sample sizes; they simply do not have the accuracy needed to perform well as we have shown in the previous section. The basic interval appears to perform slightly better than the naive interval above but that is not always the case. For instance, if we consider small samples from an exponential distribution and try to estimate the rate, the percentile method outperforms the basic (pivoted) method. The reason is that the basic method is not range preserving. The estimate of the rate of an exponential distribution should always be positive, but the confidence interval for the rate by either the normal approximation or the basic bootstrap method can have a lower endpoint which is negative.

The naive percentile method is "range preserving" and will work better in general because of its range-preserving property.

Bootstrap confidence intervals based on the studentized and BC_a methods are much more computationally burdensome. The studentized method requires some estimate of the standard error in the statistic and the BC_a method requires information about the influence each data point has on the statistic. However, both will generally be more accurate than the simpler methods.

Another basic approach to improve bootstrap methodology is to find a transformation of the statistic which makes its distribution more normal. Everything works better under normality. One can then use the inverse of the transformation to obtain a confidence interval for the original statistic from the confidence interval of the transformed statistic. For example, for the rate of an exponential distribution, the rate is the reciprocal of the mean. Means and location parameters in general will have more accurate bootstrap coverages. If we use bootstrapping to place a confidence interval around the mean of the exponential distribution, we can just take the reciprocals of the endpoints as the endpoints for a confidence interval for the rate. This technique will improve the coverage of any method.

R code used to generate the above examples

```
B = 1000

f <- function(t){
n = length(X)
out = n*log(t/mean(X)) - n*digamma(t) + sum(log(X))
out
}

alpha_stat <- function(X, freq){
f <- function(t){
n = sum(freq)
sum_log = 0
for (i in 1:length(X)){
sum_log = sum_log + ifelse(freq[i] == 0, 0, log(freq[i]*X[i]))
}
out = n*log(n*t/sum(freq*X)) - n*digamma(t) + sum_log
out
}
alpha = uniroot(f,
                interval=c(0.000001,  10))$root
lambda = alpha*n/sum(freq*X)
alpha
}

lambda_stat <- function(X, freq){
f <- function(t){
n = sum(freq)
sum_log = 0
```

```
for (i in 1:length(X)){
sum_log = sum_log + ifelse(freq[i] == 0, 0, log(freq[i]*X[i]))
}
out = n*log(n*t/sum(freq*X)) - n*digamma(t) + sum_log
out
}
alpha = uniroot(f,
                interval=c(0.000001,  10))$root
lambda = alpha*n/sum(freq*X)
lambda
}

gamma.gen <- function(data, params){
n = length(data)
alpha_est = params[1]
lambda_est = params[2]
boot_data = rgamma(n, alpha_est, lambda_est)
boot_data
}

statistic <- function(boot_data){
f <- function(t){
n = length(boot_data)
out = n*log(t/mean(boot_data)) - n*digamma(t) + sum(log(boot_data))
out
}
negloglike<- function(params){
alpha = params[1]
lambda = params[2]
n = length(X)
out = -(n*alpha*log(lambda) - n*log(gamma(alpha))
        + (alpha - 1)*sum(log(boot_data)) - lambda*sum(boot_data))
out
}

alpha_moment = (mean(boot_data)/sd(boot_data))^2
alpha_boot = uniroot(f,
                interval=c(alpha_moment/100,
                alpha_moment*100))$root
lambda_boot = alpha_boot/mean(boot_data)
var = solve(fdHess(c(alpha_boot, lambda_boot), negloglike)$Hessian)
var_alpha = var[1,1]
var_lambda = var[2,2]
c(alpha_boot, var_alpha, lambda_boot, var_lambda)
}
```

```
alpha_normal = 0
alpha_percent = 0
alpha_basic = 0
alpha_t = 0
alpha_bca = 0
lambda_normal = 0
lambda_percent = 0
lambda_basic = 0
lambda_t = 0
lambda_bca = 0

for(j in 1:B){
X = rgamma(10, .2, 1)
alpha_moment = (mean(X)/sd(X))^2
alpha_hat = uniroot(f,
                interval=c(alpha_moment/100,
                alpha_moment*100))$root
lambda_hat =  length(X)*alpha_hat/sum(X)
boot_out = boot(X, statistic, 1000, sim="parametric",
           ran.gen=gamma.gen, mle=c(alpha_hat, lambda_hat))
L_alpha = empinf(data=X, stype="f", type="jack",
                 statistic=alpha_stat)
L_lambda = empinf(data=X, stype="f", type="jack",
                  statistic=lambda_stat)
ci_alpha = boot.ci(boot_out, index=c(1,2),
           type=c("norm", "basic", "perc", "stud", "bca"),
           L=L_alpha)
ci_lambda = boot.ci(boot_out, index=c(3,4),
            type=c("norm", "basic", "perc", "stud", "bca"),
            L = L_lambda)
if (ci_alpha$norm[2] <= .2 && .2 <= ci_alpha$norm[3])
     alpha_normal = alpha_normal + 1
if (ci_alpha$perc[4] <= .2 && .2 <= ci_alpha$perc[5])
    alpha_percent = alpha_percent + 1
if (ci_alpha$basic[4] <= .2 && .2 <= ci_alpha$basic[5])
    alpha_basic = alpha_basic + 1
if (ci_alpha$stud[4] <= .2 && .2 <= ci_alpha$stud[5])
    alpha_t = alpha_t + 1
if (ci_alpha$bca[4] <= .2 && .2 <= ci_alpha$bca[5])
    alpha_bca = alpha_bca + 1

if (ci_lambda$norm[2] <= 1 && 1 <= ci_lambda$norm[3])
    lambda_normal = lambda_normal + 1
if (ci_lambda$perc[4] <= 1 && 1 <= ci_lambda$perc[5])
```

```
    lambda_percent = lambda_percent + 1
if (ci_lambda$basic[4] <= 1 && 1 <= ci_lambda$basic[5])
    lambda_basic = lambda_basic + 1
if (ci_lambda$stud[4] <= 1 && 1 <= ci_lambda$stud[5])
    lambda_t = lambda_t + 1
if (ci_lambda$bca[4] <= 1 && 1 <= ci_lambda$bca[5])
    lambda_bca = lambda_bca + 1

}

alpha_normal/B
alpha_percent/B
alpha_basic/B
alpha_t/B
alpha_bca/B
lambda_normal/B
lambda_percent/B
lambda_basic/B
lambda_t/B
lambda_bca/B

##   RESULTS   ##
#> alpha_normal/B
#[1] 0.979
#> alpha_percent/B
#[1] 0.856
#> alpha_basic/B
#[1] 0.904
#> alpha_t/B
#[1] 0.94
#> alpha_bca/B
#[1] 0.939
#> lambda_normal/B
#[1] 1
#> lambda_percent/B
#[1] 0.787
#> lambda_basic/B
#[1] 0.865
#> lambda_t/B
#[1] 0.974
#> lambda_bca/B
#[1] 0.916


### CODE FOR JUST THE RATE OF AN EXPONENTIAL DISTRIBUTION ####
```

```
B = 1000

lambda_stat <- function(X, freq){
n = sum(freq)
lambda = n/sum(freq*X)
lambda
}

exp.gen <- function(data, lambda){
n = length(data)
boot_data = rexp(n, lambda)
boot_data
}

statistic <- function(boot_data){
negloglike<- function(lambda){
n = length(X)
out = -(n*log(lambda) - lambda*sum(boot_data))
out
}
lambda_boot = 1/mean(boot_data)
var = solve(fdHess(c(lambda_boot), negloglike)$Hessian)
c(lambda_boot, var)
}


lambda_normal = 0
lambda_percent = 0
lambda_basic = 0
lambda_t = 0
lambda_bca = 0

for(j in 1:B){
X = rexp(10, 1)
alpha_moment = (mean(X)/sd(X))^2
lambda_hat =  1/mean(X)
boot_out = boot(X, statistic, 1000, sim="parametric",
                ran.gen=exp.gen, mle=lambda_hat)
L_lambda = empinf(data=X, stype="f", type="jack",
                  statistic=lambda_stat)
ci_lambda = boot.ci(boot_out, index=c(1,2),
            type=c("norm", "basic", "perc", "stud", "bca"),
```

```
            L = L_lambda)
if (ci_lambda$norm[2] <= 1 && 1 <= ci_lambda$norm[3])
    lambda_normal = lambda_normal + 1
if (ci_lambda$perc[4] <= 1 && 1 <= ci_lambda$perc[5])
    lambda_percent = lambda_percent + 1
if (ci_lambda$basic[4] <= 1 && 1 <= ci_lambda$basic[5])
     lambda_basic = lambda_basic + 1
if (ci_lambda$stud[4] <= 1 && 1 <= ci_lambda$stud[5])
    lambda_t = lambda_t + 1
if (ci_lambda$bca[4] <= 1 && 1 <= ci_lambda$bca[5])
    lambda_bca = lambda_bca + 1
}

lambda_normal/B
lambda_percent/B
lambda_basic/B
lambda_t/B
lambda_bca/B
```

References and Readings

Conceptual Exercises for Section 2.5

1.

♢ 2.6 Projects

Instructions

All projects should be written up as short research papers. You should present the data, the question, your methods, and your conclusions so that a bright researcher who is not necessarily a statistician can understand them.

Project 1: Design and carry out a simulation study to assess the robustness of one-way analysis of variance to violations in the assumptions of normality and homogeneous variances. Does a bootstrapping approach cure any robustness problems found? Write a short article about your results using appropriate tables and graphics. Describe your methods so that a lay person can understand your techniques and results.

References and Readings

Chapter 3

Bayesian Methods

♢ 3.1 Bayesian Philosophy and Theory

For a frequentist, a parameter (say, a mean) is fixed. Frequentists find confidence intervals for their parameters where the interval is random (it depends on the data which is a random sample from the population) and, by its method of construction, the interval has a certain probability of covering the true mean.

For a Bayesian, a parameter is random. Prior to collecting data, we can place some probability distribution of the parameter. The distribution might be flat (also called uniform or non-informative) or it might be a mathematically convenient conjugate prior (more about this later) or it might be some distribution that reflects prior knowledge of what the likely outcomes for the parameter are. But parameters have probability distributions according to Bayesians.

Bayesians argue that their point of view is natural and the one people adopt in their everyday thinking about probability. For instance, a frequentist never would use the phrase: "the probability the mean is in this interval is ... " because the word "probability" modifies the mean in that phrase and the mean is not random, it is fixed. The thing that is random is the interval. So a frequentist is forced to use language like "the probability the interval covers the mean is ..." A Bayesian would argue that this latter phrase is unnatural and we all want to say and even think of the situation as it is described by the former phrase.

Moreover, people do utilize prior experience when making decisions, even when evaluating the outcome of new studies. For instance, many observational studies on large groups of people suggest strong evidence for eating certain foods in order to avoid diseases such as cancer or for not eating certain foods in order to avoid disease. For instance, at one point in a large study of nurses, it was believed that eating margarine was better for your health than eating butter. Now the reverse is believed based on a later study of the same group of nurses. After being exposed to many reversals in food recommendations suggested by these large trials, I, for one, have learned to discount them. (Note that both Bayesians and frequentists might discount the results from all such studies due to their observational nature and their potential problems of confounding variables.)

The formal mathematical theory for the Bayesian method is as follows: before you collect data, you form a prior probability distribution on the parameter of interest: $p(\theta)$. The data, $\mathbf{y}$ follows a probability model,$p(\mathbf{y}|\theta)$, whose form is known in advance to determining the prior distribution on θ. (That is, you know, in advance of collecting the data, whether you will be modeling it using the normal distribution, the binomial distribution, the gamma distribution, etc...) From these, we use Bayes's theorem to get a posterior probability distribution on θ:

$$p(\theta|\mathbf{y}) = \frac{p(\mathbf{y},\theta)}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\theta)p(\theta)}{p(\mathbf{y})}$$

where $p(\mathbf{y}) = \int p(\mathbf{y}|\theta)p(\theta)d\theta$ (or the corresponding sum if the prior distribution on θ is discrete.) Note that $p(\mathbf{y})$ is just a proportionality constant needed to make the posterior distribution a probability distribution.

Example 1 Suppose you have data from independent Bernoulli trials:

$$y_i = \begin{cases} 1 & \text{with probability } p \\ 0 & \text{with probability } 1-p \end{cases}$$

Assume a flat, uniform, non-informative prior probability distribution on the parameter p (uniform on $(0,1)$). Let $x = \sum y_i$. Then the posterior distribution on the parameter p is:

$$\Pr(p|y_1, y_2, \ldots, y_n) = \frac{\Pr(y_1, y_2, \ldots, y_n|p)\Pr(p)}{\Pr(y_1, y_2, \ldots, y_n} = \frac{p^x(1-p)^{n-x}}{\int_0^1 p^x(1-p)^{n-x}dp}$$

Even if the denominator appears to be difficult to evaluate, you can figure out what it must be because the numerator is in the form of a Beta distribution on p. A Beta distribution on θ has the following probability distribution: $\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$. In our posterior distribution, $\alpha = x+1$ and $\beta = n-x+1$. Thus, we know that

$$\int_0^1 p^x(1-p)^{n-x}dp = \frac{\Gamma(x+1)\Gamma(n-x+1)}{\Gamma(n+2)}.$$

■

Note that the uniform $[0,1]$ distribution itself is a member of the Beta family with $\alpha = 1$ and $\beta = 1$ so that our prior is a Beta distribution and our posterior is a Beta distribution. Moreover, the preservation of the Beta distribution occurs for arbitrary Beta priors: if the prior is a Beta distribution with parameters α_0 and β_0, then the posterior distribution is also Beta with parameters $x+\alpha_0$ and $n-x+\beta_0$. This property is called *conjugacy* and the Beta distribution is called the conjugate prior distribution for the Bionomial model. Probability models for data that are members of exponential families will have natural conjugate priors and, in general, these are the only distributions with natural conjugate priors.

There are several logical Bayesian analog's to the frequentist's maximum likelihood point estimator for a parameter. One could use the mode of the posterior distribution, for instance. If the prior distribution on the parameter is flat, the mode of the posterior distribution is exactly the frequentist's maximum likelihood estimator. Other natural point estimates include the mean of the posterior distribution, the median of the posterior distribution, and, indeed, any measure of the center of the posterior distribution such as the trimmed means.

The Bayesian analog of the frequentist's confidence interval (say the 95% confidence interval) is the *credibility* or *posterior probability* interval. This interval is determined by the cut-offs for

the distribution that put 2.5% of the probability in the tails. For a frequentist using the normal approximation to the binomial distribution, the confidence interval will always be centered around the most likely value $\hat{p} = \frac{x}{n}$. For a Bayesian, the credibility interval can be asymmetric to reflect the skewness in the posterior Beta distribution.

Example 2 For example, assuming a flat prior, a sample of size $n = 100$, and a sum of positive responses $x = 30$, the posterior distribution on p is Beta$(31, 71)$. The 95% Bayesian credibility interval may be found by determining the cut-offs for this Beta distribution that put 2.5% of the total probability in the tails. In R, these cut-offs can be found with commands like

```
qbeta(0.025, 31, 71).
```

The 95% credibility interval is $(0.2189787, 0.3961471)$. For comparison, the frequentist 95% confidence interval is

$$\hat{p} \pm 1.96\sqrt{\hat{p}(1-\hat{p})/n} = 0.3 \pm 0.08981849 = (0.2101815, 0.3898185).$$

The Bayesian interval and the frequentist interval are very similar because the sample size is relatively large.

If $x = 3$ from a sample of size $n = 10$, the Bayesian confidence interval would be $(0.109, 0.610)$ reflecting the skewness in the posterior Beta$(4, 8)$ distribution whereas the frequentist interval would be $(0.016, 0.584)$ which is symmetric around .3 by construction. ■

The most often heard criticism of the Bayesian philosophy is that the prior distribution is arbitrary and capricious, changes to the prior distribution change the posterior predictions, and that makes for an inherently flawed statistical method. The Bayesians counter with arguments about the robustness of the conclusions with changes in the prior distribution. Of course, if you change the prior enough, you can change the posterior predictions substantially. But if you get roughly the same outcomes with reasonable changes in the prior distribution, the Bayesian results are relatively robust. The methods used to determine robustness are somewhat arbitrary and depend on the problem at hand. For instance, if we use other Beta priors (not just the uniform distribution) in the example above, with more of a peak at $p = 1/2$, say, then our conclusions remain relatively robust as long as $\alpha + \beta << n$ where n is our sample size. That is, we need priors that do not swamp the information we get from our sample.

References and Readings

[1] A Gelman, Carlin J B, Stern H S, and Rubin D B. *Bayesian Data Analysis*. Chapman Hall, CRC, Boca Raton, Florida, 1995.

Conceptual Exercises for Section 3.1

1. Suppose that the prior distribution on the proportion of successes in a series of i.i.d. Bernoulli trials is the tent function, $\Pr(p) = 2 - 4 * |p - 0.5|$. Find the form of the posterior distribution without evaluating the proportionality constant $p(\mathbf{y})$. How might you find $p(\mathbf{y})$ approximately?

♢ 3.2 Metropolis and Metropolis-Hastings Algorithms

Recall that the Bayesian posterior distribution on parameter(s) θ given data $\mathbf{y}$ is

$$\Pr(\theta|\mathbf{y}) = \frac{\Pr(\mathbf{y}|\theta)\Pr(\theta)}{\Pr(\mathbf{y})}$$

where $\Pr(\mathbf{y}|\theta)$ is the likelihood model for the data, $\Pr(\theta)$ is the prior distribution on the parameter(s) θ, and $\Pr(\mathbf{y}) = \int \Pr(\mathbf{y}|\theta) d\Pr(\theta)$ is the proportionality constant needed to make the posterior distribution a probability distribution.

One problem with our ability to use the Bayesian method is that the posterior distribution may not be a standard distribution and we may not be able to calculate the proportionality constant easily or at all. For instance, if our parameter space is a complicated combinatorial space, such as the space of all permutations on m objects or the space of all bifurcating trees with m tips, calculating the proportionality constant could require summing over a prohibitively vast number of states.

Fortunately, the Metropolis algorithm (and its generalization, the Metropolis-Hastings algorithm) allow us to sample from the posterior distribution without having to know the proportionality constant. The Metropolis algorithm requires a method for drawing an initial parameter from parameter space. It also requires a jumping rule at each time-step, t, that tells us how to select a new point in parameter space given our current point. This jumping rule must be symmetric; that is, the probability of jumping from a to b must be the same as the probability of jumping from b to a.

The Metropolis Algorithm

- Draw an initial point θ^0 using an initial probability distribution on parameter space $\Pr_0(\theta)$.

Repeat the following steps

- Sample a new point θ^* using the jumping distribution $J_t(\theta^*|\theta^{t-1})$. The jumping distribution must satisfy $J_t(\theta_b|\theta_a) = J_t(\theta_a|\theta_b)$ for all θ_a, θ_b.
- Calculate the relative importance ratio

  $$r = \frac{\Pr(\theta^*|\mathbf{y})}{\Pr(\theta^{t-1}|\mathbf{y})}$$

  Note that this ratio involves the posterior distribution on the parameter but does not depend the proportionality constant needed to guarantee that the distribution is a probability distribution.
- If $r \geq 1$ then set $\theta^t = \theta^*$. That is, if the new point is more likely, you always make the jump. Otherwise let $\theta^t = \begin{cases} \theta^* & \text{with probability } r \\ \theta^{t-1} & \text{otherwise} \end{cases}$. That is, if the new point is less likely, we sometimes jump to it and sometime do not jump to it where our chances of jumping depends on how much less likely it is relative to our current point.

Claim: This process determines a Markov chain with unique stationary distribution equal to $\Pr(\theta|\mathbf{y})$ as long as the jumping distribution is not too restrictive.

The Metropolis algorithm gives a Markov chain. The chain is irreducible, aperiodic, and recurrent if every state can be reached from every other state with positive probability. The chain will meet this requirement as long as the jumping distribution allows for enough movement among states. Any irreducible, aperiodic, recurrent Markov chain has a unique stationary distribution. We show that it must be $\Pr(\theta|\mathbf{y})$ by showing that $\Pr(\theta|\mathbf{y})$ is stationary. That is, assume θ^{t-1} is drawn from $\Pr(\theta|\mathbf{y})$. Consider θ_a and θ_b where $\Pr(\theta_b|\mathbf{y}) \geq \Pr(\theta_a|\mathbf{y})$. Then

$$\Pr(\theta^{t-1} = \theta_a, \theta^t = \theta_b|\mathbf{y}) = \Pr(\theta_a|\mathbf{y})J_t(\theta_b|\theta_a)$$

since we always make the jump since θ_b is more likely than θ_a. Similarly,

$$\Pr(\theta^{t-1} = \theta_b, \theta^t = \theta_a|\mathbf{y}) = \Pr(\theta_b|\mathbf{y})J_t(\theta_a|\theta_b)\frac{\Pr(\theta_a|\mathbf{y})}{\Pr(\theta_b|\mathbf{y})} = \Pr(\theta_a|\mathbf{y})J_t(\theta_b|\theta_a).$$

This argument demonstrates that the joint distribution of θ^t and θ^{t-1} given the data is symmetric ($\Pr(\theta^{t-1} = \theta_b, \theta^t = \theta_a|\mathbf{y}) = \Pr(\theta^{t-1} = \theta_a, \theta^t = \theta_b|\mathbf{y})$ for all θ_a, θ_b. This symmetry implies that the marginal distributions for θ^t and θ^{t-1} are the same. Thus $\Pr(\theta|\mathbf{y})$ is the stationary distribution of the chain given by the Metropolis algorithm. □

There are some important issues involved in the use of this algorithm. One issue is that $\theta^t, \theta^{t+1}, \theta^{t+2}, \ldots$ are not independent. To get an approximately independent sample, one can sample only every m^{th} element in this chain. One can also ignore this issue and recognize that the empirical distribution given by the dependent sample will represent a smaller effective sample size than the nominal sample size. For example, the empirical distribution determine by a sample of 1000 dependent values for θ may be equivalent to the empirical distribution determined by a sample of 100 independent values of θ, and, thus, coarser than one would expect for so large a sample.

Another issue is that we need methods to determine when the Markov chain provided by the Metropolis algorithm converges. There are many ad hoc methods. The following one, proposed by Gelman and Rubin (1992), is based on considering the within-chain and between-chain variance for at least 2 chains - that is, two runs of the Metropolis algorithm that begin at different starting points. Consider J chains each run for n iterations of the Metropolis algorithm. Let $\theta_{i,j}$ be the i^{th} value from chain j ($1 \leq i \leq n$), $\bar{\theta}_{\cdot,j}$ be the average value for chain j, $\bar{\theta}_{\cdot,\cdot}$ be the average value for all the chains combined, and define

$$B = \frac{n}{J-1}\sum_{j=1}^{J}(\bar{\theta}_{\cdot,j} - \bar{\theta}_{\cdot,\cdot})^2 \quad \text{and} \quad W = \frac{1}{J(n-1)}\sum_{j=1}^{J}\sum_{i=1}^{n}(\theta_{i,j} - \bar{\theta}_{\cdot,j})^2$$

Then a general rule of thumb is that we can start accepting values produced by the chains when

$$\sqrt{\hat{R}} = \sqrt{\frac{\frac{n-1}{n}W + \frac{1}{n}B}{W}}$$

is less than 1.2 or so. (Note that R depends on n as it is calculated based on the first n values of the chain. We need to keep track of it for all n to be sure that we do not lose convergence as we can do if we use a poor choice for the jumping distribution.)

One has to be somewhat careful when implementing any generic rule such as the one above. If one has a multivariate parameter space, one can apply the above rule coordinate-wise and wait until the value of $\sqrt{\hat{R}}$ is less than 1.2 for every coordinate. However, there may be more sensible things to do that take the multidimensional nature of the parameter space into account. The above rule also may not be of much use for combinatorial parameters. Further, the above rule is based on an ANOVA-like method and may not work well if the asymptotic distribution for the parameter is highly non-normal.

Other issues involve the choice of the starting distribution and the choice of the jumping distribution. Choosing an initial point for the chain from a distribution as close to the stationary distribution as possible is ideal. If the initial point is chosen from a distribution that is far from the stationary distribution, we will have a longer "burn-in" period and will have to wait longer before we can collect data from the Metropolis algorithm. An ideal jumping distribution allows for a reasonable amount of mixing in the Markov chain. That is, we want each jump to go a reasonable distance but also have a reasonably high probability of being accepted. As a general rule of thumb, you want about 50% of jumps accepted in with univariate parameters and 25% of jumps accepted in high-dimensional parameter spaces. There are also practical considerations: you want a jumping distribution that is easy to sample from.

We do not need the Metropolis algorithm to sample from the posterior distribution of the Bernoulli probability parameter with a Beta prior and a Bernoulli model on the data. We know that the posterior distribution is a Beta distribution, we know the proportionality constant, and standard routines is most software programs will sample from the posterior distribution and perform any posterior analysis we wish through numerical integration. However, we will use this example to illustrate issues with the Metropolis algorithm.

Example 1 Suppose our prior distribution on p is uniform $(0, 1)$, we have the outcome of n independent Bernoulli trials, $y_1, y_2, \ldots, y_n \sim \text{Bernoulli}(p)$, $x = \sum_{i=1}^{n} y_i$, and we know that the posterior distribution is proportional to $p^x(1-p)^{n-x}$. If we want to use the Metropolis algorithm to sample from the posterior distribution, we need to specify a symmetric jumping distribution. One possible choice is to use the uniform distribution on $(0, 1)$ so that $J_t(p^*|p^{t-1})$ does not depend on t and does not depend on p^{t-1}. R code to implement this procedure when the sample size is 30 and a value of $p = .2$ is used to generate the data is given as follows:

```
# generate data with a specific value for p
x=rbinom(1, 30, .2)
n = 30
# obtain the proportional posterior distribution on p
# via metropolis-hastings algorithm
pr <- function(n, x, p_in){
p_in^(x)*(1 - p_in)^(n-x)
}
#create the jumping function
J1 <- function(t, p_in){
p_out = runif(1, 0, 1)
p_out
}
```

```
# analyze using 2 chains with separated starting points
proportion_accepted = 0
J1chain1 = array(NA, dim=c(1000))
J1chain2 = array(NA, dim=c(1000))
J1_R_hat = array(NA, dim=c(1000))
J1_R_hat[1] = Inf
J1chain1[1] = runif(1, 0, .25)
J1chain2[1] = runif(1, .75, 1)
for(i in 2:1000){
J1chain1[i] = J1(i, J1chain1[i-1])
J1chain2[i] = J1(i, J1chain2[i-1])
r1 = pr(n, x, J1chain1[i])/pr(n, x, J1chain1[i-1])
r2 = pr(n, x, J1chain2[i])/pr(n, x, J1chain2[i-1])
uniform_toss = runif(1, 0, 1)
if(uniform_toss > r1) J1chain1[i] = J1chain1[i-1]
else proportion_accepted = proportion_accepted + 1
uniform_toss = runif(1, 0, 1)
if(uniform_toss > r2) J1chain2[i] = J1chain2[i-1]
else proportion_accepted = proportion_accepted + 1
B = i*(mean(J1chain1, na.rm=TRUE) - mean(J1chain2, na.rm=TRUE))^2/2
W = (1/2)*(var(J1chain1, na.rm=TRUE) + var(J1chain2, na.rm=TRUE))
J1_R_hat[i] = sqrt((((i-1)/i)*W + (1/i)*B)/W)
} # end for i in 2:1000
proportion_accepted = proportion_accepted/(1998)
proportion_accepted
```

In this example, the chains converge rapidly but the proportion of jumps accepted in a bit low, around 20%. ■

We might want to choose a jumping distribution that does not range so widely but, instead, samples around p^{t-1}, such as a uniform distribution on $(p^{t-1} - 1/\sqrt{30}, p^{t-1} + 1/\sqrt{30})$. However, due to the edge effects when p^{t-1} is close to zero or one, this jumping distribution will not satisfy the symmetry requirements of the Metropolis algorithm. That is, there will be some values of p and p^* for which $J_t(p^*|p) \neq J_t(p|p^*)$. Hastings provided the following useful extension of the Metropolis algorithm when it is inconvenient to require symmetry in the jumping distribution. The simple modification replaces the previous calculation of the importance ratio with

$$r = \frac{\Pr(\theta^*|\mathbf{y})/J_t(\theta^*|\theta^{t-1})}{\Pr(\theta^{t-1}|\mathbf{y})/J_t(\theta^{t-1}|\theta^*)}$$

Example 2 Consider using the jumping distribution $J_t(p^*|p^{t-1})$ being the uniform distribution on $(p^{t-1} - 1/\sqrt{30}, p^{t-1} + 1/\sqrt{30})$ in the previous example. The following code implements the Hastings modification of the Metropolis algorithm:

```
# generate data with a specific value for p
```

```
x=rbinom(1, 30, .2)
n = 30
# obtain the posterior distribution on p
# via metropolis-hastings algorithm
pr <- function(n, x, p_in){
p_in^(x)*(1 - p_in)^(n-x)
}
#create the jumping function
J2 <- function(t, p_in){
p_out = runif(1, max(0,  p_in - 1/sqrt(30)), min(1,  p_in + 1/sqrt(30)))
p_out
}
#use the Metropolis-Hastings Algorithm
proportion_accepted = 0
J2chain1 = array(NA, dim=c(1000))
J2chain2 = array(NA, dim=c(1000))
J2_R_hat = array(NA, dim=c(1000))
J2_R_hat[1] = Inf
J2chain1[1] = runif(1, 0, .5)
J2chain2[1] = runif(1, .5, 1)
for(i in 2:1000){
J2chain1[i] = J2(i, J2chain1[i-1])
J2chain2[i] = J2(i, J2chain2[i-1])
r1 = (pr(n, x, J2chain1[i])/ (min(1,  J2chain1[i-1] + 1/sqrt(30)) - max(0,  J2chain1[i-1] -
r2 = (pr(n, x, J2chain2[i])/ (min(1,  J2chain2[i-1] + 1/sqrt(30)) - max(0,  J2chain2[i-1] -
uniform_toss = runif(1, 0, 1)
if(uniform_toss > r1) J2chain1[i] = J2chain1[i-1]
else proportion_accepted = proportion_accepted + 1
uniform_toss = runif(1, 0, 1)
if(uniform_toss > r2) J2chain2[i] = J2chain2[i-1]
else proportion_accepted = proportion_accepted + 1
B = i*(mean(J2chain1, na.rm=TRUE) - mean(J2chain2, na.rm=TRUE))^2/2
W = (1/2)*(var(J2chain1, na.rm=TRUE) + var(J2chain2, na.rm=TRUE))
J2_R_hat[i] = sqrt((((i-1)/i)*W + (1/i)*B)/W)
} # end for i in 2:1000
proportion_accepted = proportion_accepted/(1998)
proportion_accepted
```

In this example, convergence is a little slower (instead of occurring instantly, it takes about 20 interations. However, with this small price, we accept about 50% of the jumps and thus, are not stuck at the same point too often in the output chain. ■

Example 3 The jumping distribution $J_t(p^*|p^{t-1})$ can also depend on time. However, one must be careful. Suppose we wanted to use the uniform distribution on $(p^{t-1} - 1/t, p^{t-1} + 1/t)$ with the proper modifications for the edge effects at 0 and 1. The result is that the chains slow down so much, they

often do not converge to the stationary distribution within 1000 iterations. Even when they do, the effective sample size will be much, much smaller than the nominal size. Further, too many of the jumps are accepted - over 98% of them - indicating that the jumping distribution is not sampling enough of the parameter space to be effective. ■

References and Readings

[1] S Chib and Greenberg E. Understanding the metropolishastings algorithm. *The American Statistician*, 49:327335, 1995.

[2] A Gelman, Carlin J B, Stern H S, and Rubin D B. *Bayesian Data Analysis*. Chapman Hall, CRC, Boca Raton, Florida, 1995.

[3] A Gelman and Rubin D B. Inference from iterative simulations using multiple sequences. *Statistical Science*, 7:457–472, 1992.

[4] W K Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[5] N Metropolis, Rosenbluth AW, Rosenbluth M N, Teller A H, and Teller E. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

Conceptual Exercises for Section 3.2

1. Show that the stationary distribution of the Metropolis-Hastings algorithm is the appropriate posterior distribution, $\Pr(\theta|\mathbf{y})$.

2. Write the code to implement Example 3 in this section.

♢ 3.3 Nuisance Parameters and Conserved Synteny Example

In the case of estimating a Bernoulli or Binomial proportion, I would argue that the Bayesian philosophy needlessly complicates a simple situation. In the following example, I would argue that the Bayesian approach is appropriate and demanded. The question below was posed to me by John Postlethwait, a developmental biologist at the University of Oregon. Some work on the problem had been done previously but generally did not take into account the dependencies inherent in the data. Here is the problem:

Species, such as mice and humans (and even yeast and humans), share many of the same genes. For instance, there is a gene on Mouse chromosome 5, Adora2A, which is the adenosine A2a receptor. This gene shares much sequence similarity with a gene, ADORA2A, on human chromosome 22. For instance, the first 44 amino acids in the human and mouse versions of the gene can be aligned as follows:

```
Human:  MPIMGSSVYITVELAIAVLAILGNVLVCWAVWLNSNLQNVTNYF
           ||||||| ||||||||||||||||||||| ||||||||| |
Mouse:     MGSSVYIMVELAIAVLAILGNVLVCWAVWINSNLQNVTNFFVVS
```

Further, the gene is responsible for the similar things - response to inflamation - in both creatures. Thus we are pretty sure the genes are orthologs - that is a gene in two species that evolves from a single gene in their most recent common ancestor.

A conserved synteny is a pair of chromosomes - one from each species - on which one or more orthologs reside. In this case, mouse chromosome 5 and human chromosome 22 have a conserved synteny. If genomes tended to be shuffled around a lot in evolution, we might expect every possible chromosome pair to be a conserved synteny. However, when we map genes to two species, we see this is not the case. For instance, genes on human chromosome 22 have orthologs on mouse chromosomes 6, 8, 10, 11, 15, and 16 but not on mouse chromsomes 1-5, 7, 9, 12-14, or 17-19. (We exclude the sex chromosomes X and Y for these purposes since they have specialized roles and the genes on them tend to remain on them and not move to other chromosomes.)

One possible representation of this data is by listing the counts of orthologs occurring between two species in an $r \times c$ table representing the r chromosomes from the first species and the c chromosomes from the second species. Data from two species with 5 and 6 chromosomes respectively, might looks something like the following:

Species 1	Species 2 chromosomes					
chromosomes	1	2	3	4	5	6
1	0	20	31	0	0	0
2	0	0	0	53	29	0
3	21	0	0	0	0	15
4	5	25	0	0	31	0
5	0	23	0	0	4	0

The question we wish to address is the following: if we map 1000 of, say, 30,000 possibly orthologs between mice and humans and see a specific number of conserved syntenies, what is the total number of conserved syntenies we are likely to see when the entire genome of both organisms are mapped?

How might a frequentist think about this problem? Clearly the model is a multinomial distribution with a sample of size $n = 1000$ and $m = r \times c$ categories. The single best guess for the number of conserved syntenies is the number we see. But we can think about placing an upper bound on this number by calculating a cut-off on the total probability of the currently empty cells. That is, we need a bound on q where q represents the total probability of all the chromosome pairs that do not, at this point, share any orthologs. So all 1000 genes have been mapped to other chromosome pairs and we want the value of q so that

$$(1-q)^{1000} \geq 0.05 \qquad \text{so} \qquad q \leq 1 - (0.05)^{1/1000} = 0.003.$$

With this value, 90 genes or so may eventually get mapped to the now-current cells. If the genes fall randomly, then we expect something around 90 additional cells to eventually be occupied but this is an unbelievable result to biologists because they do not expect many more cells to be occupied.

How might a Bayesian think about this problem? The statistic of interest is l the number of cells in the $m = r \times c$ table that will ever be occupied. Suppose we see k conserved syntenies (k non-zero cells). We know $l \geq k$. Suppose our data consists of observing n_i orthologs in cell c_i, $1 \leq i \leq k$. The probability model on the data is multinomial:

$$\Pr(n_1 \text{ genes in cell } c_1, \ldots, n_k \text{ genes in cell } c_k) = \binom{n}{n_1 \quad n_2 \quad \ldots \quad n_k} p_1^{n_1} p_2^{n_2} \cdots p_k^{n_k}$$

The parameters p_1, p_2, *dots*, p_k are nuisance parameters necessary for the model but not of interest really to our questions. There are actually $l - 1$ such nuisance probability parameters p_1, p_2, *dots*, p_l where the last one is determined by the first $l - 1$ by the requirement that $p_1 + p_2 + \ldots p_l = 1$. A reasonable model for these parameters is uniform on the simplex $p_1 + p_2 + \cdots + p_{l-1} \leq 1$ and the density function for this prior distribution is $\dfrac{1}{(l-1)!}$.

Further nuisance parameters include which l cells the data will ever fall into. The prior distribution here (before you see the data and thus know which cells some of the data will ever fall into) is uniform on the possible choices of l cells taken from $m = r \times c$ cells. This prior distribution has density $\dfrac{1}{\binom{m}{l}}$.

The posterior distribution on the parameter of interest, l, the number of cells that will ever contain genes, integrates (or sums) out the nuisance parameters:

$\Pr(l|n_1 \text{ genes in cell } c_1, \ldots, n_k \text{ genes in cell } c_k)$

$$\begin{aligned}
&\propto \sum_{\substack{\text{choices for } l-k \\ \text{additional cells}}} \Pr(l|n_1 \text{ genes in cell } c_1, \ldots, n_k \text{ genes in cell } c_k, c_{k+1}, \ldots, c_l) \frac{1}{\binom{m}{l}} \\
&= \Pr(l|n_1 \text{ genes in cell } c_1, \ldots, n_k \text{ genes in cell } c_k, c_{k+1}, \ldots, c_l) \frac{\binom{m-k}{l-k}}{\binom{m}{l}} \\
&= \int\int\cdots\int \Pr(l|n_1 \text{ genes in cell } c_1, \ldots, n_k \text{ genes in cell } c_k, c_{k+1}, \ldots, c_l, p_1, p_2, \ldots, p_l) \times \\
&\qquad\qquad \frac{\binom{m-k}{l-k}}{\binom{m}{l}} \frac{1}{(l-1)!} dp_1 dp_2 \ldots dp_{l-1} \\
&= \int\int\cdots\int \binom{n}{n_1 n_2 \ldots n_k} \frac{\binom{m-k}{l-k}}{\binom{m}{l}} \frac{1}{(l-1)!} dp_1 dp_2 \ldots dp_{l-1} \\
&= \frac{\binom{m-k}{l-k}}{\binom{n+l-1}{l-1}\binom{m}{l}}
\end{aligned}$$

In this case, the posterior distribution is so nice, we can easily find the scaling constant by finding

$$C = \sum_{l=k}^{m} \frac{\binom{m-k}{l-k}}{\binom{n+l-1}{l-1}\binom{m}{l}}$$

The density function for l is then

$$f(l) = \frac{1}{C} \frac{\binom{m-k}{l-k}}{\binom{n+l-1}{l-1}\binom{m}{l}}$$

and the tail of the distribution is easy to calculate for specific values of m, n, and k.

There are serious problems with the application of this theoretical technique. These problems come about in the data collection procedure. Biologists find orthologs by "blasting" the genes already known against the DNA sequences in a new species. Thus the orthologs found do not represent a random sample of all the possible orthologs. They represent genes that are fairly well conserved. It may be that these genes have less freedom to move around on the genomes, perhaps due to requiring shared regulation or function with nearby genes. Perhaps, more distance, less conserved orthologs are more free to move around the genome and more conserved syntenies will eventually be found than this method predicts.

There is a professional question about whether developing idealistic models for imperfect data (in this case, when the data do not meet the model's assumptions of independence and random sampling)

is a worthwhile professional activity. Obviously, I personally was happy to do so in this case. Others would not be.

References and Readings

[1] A Gelman, Carlin J B, Stern H S, and Rubin D B. *Bayesian Data Analysis*. Chapman Hall, CRC, Boca Raton, Florida, 1995.

[2] E A Housworth and Postlethwait J. Measures of synteny conservation between species pairs. *Genetics*, 162:441–448, 2002.

Conceptual Exercises for Section 3.3

1.

♢ 3.4 Combinatorial Example

Suppose you want to rank m children in terms of their intelligence (ability). The parameter of interest, the ranking, is a permutation on the integers $1-m$, π. Even before considering the model for the data we can note that a flat, uniform, uninformative prior is permissible and, perhaps, desirable.

The data consists of the scores on n tests. Each set of scores ranks the students (with ties allowed). Thus, the data could be provided in a table such as:

	Exam				
child	1	2	3	$\cdots$	m
1	90	80	100	$\cdots$	70
2	95	100	85	$\cdots$	90
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
n	100	85	95	$\cdots$	50

It could be that the exams are all on the same scale and come from a normal distribution with different means for each child but a common standard deviation, say. We are going to assume that the exams are not on the same scale and that the important information conveyed by an exam is the ranking that exam places on the student. That is, the data from the i^{th} exam is really a ranking σ_i of the m students with ties allowed.

We need a form for the probability model $\Pr(\sigma_i|\pi)$ where π is the "true" ranking of the students. We are going to impose a somewhat arbitrary one that decays exponentially with the "distance" between two rankings. The distance between rankings is defined to be the number of times they disagree; that is, the distance between σ_i and π, $d(\sigma_i, \pi)$, is the number of pairs (j, k) so that $\sigma_i(j) < \sigma_i(k)$ but $\pi(j) > \pi(k)$ or $\sigma_i(j) > \sigma_i(k)$ but $\pi(j) < \pi(k)$. Note that these inequalities are strict so that ties do not count as disagreeing with either possible ordering of the individuals. Also note that this distance is between 0 and $m(m-1)/2$, since $m(m-1)/2$ is the number of possible pairs and rankings in reverse order of each other will have all pairs in conflict.

We assign the following probability distribution on the observed rankings given a true one:

$$\Pr(\sigma_i|\pi) \propto e^{-\frac{2}{m-1}d(\sigma_i,\pi)}$$

Thus, the posterior probability distribution on the true rankings, π, is given, up to a proportionality constant, by

$$\Pr(\pi|\sigma_1, \sigma_2, \ldots, \sigma_n) \propto e^{-\frac{2}{m-1}\sum_{i=1}^{n} d(\sigma_i,\pi)}$$

The proportionality constant is determined by summing the right hand side above over all permutations on the m children. If m is large, this summation may require a great deal of computational time. Further, it is not clear that knowing the proportionality constant and hence the actual probability distribution will be more helpful to use that using the Metropolis-Hastings algorithm without computing the proportionality constant.

To use the Metropolis-Hastings algorithm, we need to choose a jumping distribution on the permutations on m children. There are some obvious choices. We could choose to jump to a permutation at random. What are the advantages and disadvantages of choosing a random permutation? Getting one is easy but it is unlikely that the one we choose will be accepted; therefore the acceptance rate for the jump will be extremely low. We could define "neighboring" permutations and jump to a neighbor. The acceptance rate will likely be reasonably high because neighbors do not differ very much. What is a "neighboring" permutation. We could define permutations to be neighbors if their distance is 1. The two permutations differ by interchanging ranks 1 and 2, 2 and 3, ..., or $m-1$ and m. Such a jumping distribution is symmetric so the simple Metropolis algorithm works. This jumping distribution takes a while to wander around permutation-space. By going a distance of 1 each time, it will take $m(m-1)/2$ steps to guarantee that you can jump from any permutation to any other permutation with some small but positive probability.

How do we assess convergence for a combinatorial parameter? We need some measure of between-chain and within-chain variance. We can utilize our distance measure between permutations to get such a measure or we could view the rankings as a m-dimensional vector of numbers and use the maximum of the m individual convergence statistics.

Individual questions can then be answered using the empirical posterior distribution of rankings. For instance, to find the probability child l ranks first in the class, you can simply determine the proportion of generated posterior rankings for which child l ranks first.

References and Readings

[1] A Gelman, Carlin J B, Stern H S, and Rubin D B. *Bayesian Data Analysis*. Chapman Hall, CRC, Boca Raton, Florida, 1995.

Conceptual Exercises for Section 3.4

1.

♢ 3.5 Projects

Instructions

All projects should be written up as short research papers. You should present the data, the question, your methods, and your conclusions so that a bright researcher who is not necessarily a statistician can understand them.

Project 1: Data on the exam scores of 10 students are provided. Write the code to find the Bayesian posterior distribution on the ranks of the students. What is the Bayesian posterior probability that John ranks first in the group? What is the Bayesian posterior probability that the boys' average rank is higher than the girls' average rank? When you write your report, make sure you discuss the model so that it is clear and a lay person can understand it. Also discuss how a frequentist might approach this question. Is there inherently anything better about the Bayesian approach in this case?

References and Readings

Chapter 4

Contingency Tables

♢ 4.1 Sampling Schemes and Contingency Tables

♢ 4.2 The χ^2 Statistic

♢ 4.3 To Condition or Not to Condition?

We will consider the philosophy of analyzing information that can be summarized in a 2×2 table. The question most often being asked is whether two independent proportions are equal or not and here are many possible tests for addressing the question when all the cell counts are large: the Chi-square test, the Chi-square test with Yates's correction, the difference in the normal approximations to the binomial distribution based-test, the log odds ratio test, and Fisher's exact test spring immediately to mind. In fact, when all the cell counts are large, all of these tests give essentially the same result and any difference in statistical philosophy can be swept under the rug since the final result of the test will not differ in any vital way when another test is used.

When some of all of the expected cell counts are small, however, one test may give a relatively small p-value and another test may give a relatively large p-value - with enough of a difference between them that it is difficult to argue that the tests are essentially equivalent. Philosophy becomes important. The big philosophical debate with tables with small expected counts is whether or not to condition on the marginal totals.

Reportedly, Fisher proposed his Exact Test at a tea party when a woman attending the party claimed she could tell whether the milk was poured into the cup before the tea or after. Fisher designed and carried out an experiment right then and there. He told the woman there would be 8 cups of tea - four with the milk poured first and four with the milk poured second. The woman had to guess which of the cups had the milk poured first and which had the milk poured second. Thus, both the row totals and the column totals in this experiment were fixed in advanced (a highly unusual situation.)

The resulting table was:

	milk first	milk second
guess milk first	3	1
guess milk second	1	3

Fisher noticed that the significance can be computed using the hypergeometric distribution. (Imagine an urn containing 8 balls, 4 black representing that the milk was poured first and 4 white representing that the milk was poured second. The woman draws out 4 balls at random to constitute her guess that the milk was poured first. What is the probability she draws out 3 or more black balls?)

The p-value of the test in this case is

$$\frac{\binom{4}{3}\binom{4}{1}}{\binom{8}{4}} + \frac{\binom{4}{4}\binom{4}{0}}{\binom{8}{4}} = \frac{17}{70}$$

There are a few things to note about these problems. The woman did not have much continuity in the p-values she could achieve. If she has guessed all four cups correctly, she would have had a highly significant p-value of 1/70. As it was, she achieved the next best p-value possible, which was an insignificant 17/70.

Another thing is that the calculation above is perfectly correct. The marginals were fixed in advance and the hypergeometric distribution is the correct distribution. However, in most scientific

problems, at least one marginal is not fixed in advance but is determined by the experiment. Why should one calculate the significance conditional on the observed margins, as Fisher's exact test would have us do?

This problem has been a source of controversy for over 50 years. Indeed, George Barnard proposed a method for an exact test without fixed marginals, called CSM, in the 1940's. We will not discuss that test here, because, among other things, Barnard entered into a debate with Fisher over whether it was appropriate to condition on the marginals or not and Fisher won the debate. Indeed, Fisher commented that Barnard was the only statistician who had ever admitted he was wrong. Later, Barnard wrote that the name of the test also referred to his hated Company Sergeant Major (he developed the test shortly after serving in WWII) and that "the test, like the man, is best forgotten. "

The argument that Fisher presented that convinced Barnard was based on the notion of ancillary statistics. We leave the definition of ancillarity for a moment and focus on Fisher's example. Consider a flowering plant. Suppose we are interested in the probability p that the plant will flower purple instead of white and we want to test whether $p = 1/2$ or is less than a half. Suppose we have four specimens each of which have probability $3/4$ of flowering. On cultivation all four specimens flower white. The one-sided p-value conditional on the fact that all four plants flowered is $1/16$. Should we multiply this by the probability that the four plants would flower, which in this case is $81/256$? If we do, then not only does the result change from being insignificant to significant, a good horticulturist who can get the plants to flower with probability 1 would be penalized for his cultivation skills.

Barnard's initial counter-argument included that this example has little to do with 2×2 tables. Whether a plant flowers or not above is assumed to be independent of the color of the flowers. The margin totals in a 2×2 table are not independent of the log odds ratio for the two proportions. However, upon reflection, Barnard agreed with Fisher that the margins should be considered fixed. The argument why is based on (approximate) ancillarity. As in Fisher's example we say that $\mathbf{Y}$ is an ancillary statistic for θ if

$$\Pr(\mathbf{X}, \mathbf{Y}|\theta, \varphi) = \Pr(\mathbf{X}|\theta, \mathbf{Y})\Pr(\mathbf{Y}|\varphi)$$

That is, $\mathbf{Y}$ contains no information about θ. In Fisher's example, the number of plants that the horticulturist gets to flower contains no information about the result of the flowering (white or purple.)

Assume we have an experiment where we fix the row totals n_1 and n_2. Thus, the resulting data can be summarized in a table such as:

a	b	n_1
c	d	n_2
m_1	m_2	n

Assume a is Binomial with n_1 trials and probability π_1 of success and c is independent and Binomial with n_2 trials and probability π_2 of success. The joint probability density is given by

$$\Pr(a, m_1|\pi_1, \pi_2) = \binom{n_1}{a}\pi_1^a(1-\pi_1)^{n_1-a}\binom{n_2}{m_1-a}\pi_2^a(1-\pi_2)^{n_2-(m_1-a)}$$

If we change notation and let θ be the odds ratio, $\dfrac{\pi_1(1-\pi_2)}{\pi_2(1-\pi_1)}$, and φ be the odds product, $\dfrac{\pi_1\pi_2}{(1-\pi_1)(1-\pi_2)}$ then

$$\Pr(a, m_1|\theta, \varphi) = \binom{n_1}{a}\binom{n_2}{m_1-a}\theta^a\left(\sqrt{\theta\varphi}+1\right)^{-n_1}\left(\sqrt{\frac{\varphi}{\theta}}\right)^{m_1}\left(1+\sqrt{\frac{\varphi}{\theta}}\right)^{-n_2} = \Pr(a|m_1,\theta)\Pr(m_1|\theta,\varphi)$$

Thus the marginal, m_1, does contain some information about θ. The ancillary argument does not quite hold. The argument becomes that the marginal m_1 contains very little information about θ. This argument is indeed vague but you can try to figure out what information you get about the odds ratio $(ad)/(bc)$ from knowing the marginals in the table above.

References and Readings

[1] R J A Little. Testing the equality of two independent binomial proportions. *The American Statistician*, 43:283–288, 1989.

[2] F Yates. Test of significance for 2×2 contingency tables. *Journal of the Royal Statistical Society, Series A*, 247:426–463, 1984.

Conceptual Exercises for Section 4.3

1.

♢ 4.4 The Use of χ^2 as a Measure of Independence

♢ 4.5 Random Effects MetaAnalyses

Simpson's paradox implies that you have to be careful when combining the results of 2×2 tables because you can reverse the results. One classic example is the Berkeley admissions data. When you just consider the percentage of women versus men who were admitted to graduate school (to the largest 6 departments) at Berkeley in 1973, you get the following 2×2 table:

	accepted	rejected	total
women	557	1278	1835
men	1198	1493	2691
total	1755	2771	4526

Such a table makes it seem like Berkeley was discriminating against women. The odds ratio of women's acceptance to men's is $= \dfrac{(557)(1493)}{(1198)(1278)} = 0.54$ meaning that a woman's odds was about half of a man's odds of being accepted. Log odds are approximately normally distributed with variance $\dfrac{T^3}{R_1 R_2 C_1 C_2}$ where R_i is the row total for row i, C_j is the column total for column j, and T is the total number of subjects in the table. For this table, the variance is 0.0039. A significance test for equal odds thus gives $Z = \dfrac{\ln .54}{\sqrt{0.0039}} = -9.8$. Thus, the odds a woman is admitted to one of the largest 6 departments at Berkeley is significantly lower than the odds a man is admitted.

Of course, students are not admitted to Berkeley for graduate school by a common board. They are admitted by department. When the data for the 6 largest departments is broken down by department, you see the results contained in the next table. (Can you guess which departments they were? They do not look like math department numbers to me. At the time, Berkeley has 101 graduate programs. These departments probably contained engineering and law.)

The Mantel-Haenszel test allows one to control for the confounding factor (department) while combining the results of the separate 2×2 table. To conduct the test, you find the "excess" for each table. You can choose any one of the four cells. Here, for the sake of verbal explanation, we will consider the excess of men admitted to Berkeley. The excess is simply the observed minus the expected for that cell (and it would be the same, with possibly a sign change, for any other cell chosen.) The variance of the excess is $\dfrac{R_1 R_2 C_1 C_2}{TT(T-1)}$ where R_i is the row total for row i, C_j is the column total for column j, and T is the total number of subjects in the table. (The variance formula comes from the variance of the hypergeometric distribution.)

Since the total excess of male admissions is negative, men may be being discriminated against. They are certainly not being discriminated for. So we see immediately that the p-value of the test is greater than 50%. In general, the total excess divided by the square root of the total variance is approximately standard normal under the null hypothesis of equal odds in every table.

Department A					
sex	accepted	rejected	totals	excess	variance
f	89	19	108		
m	512	313	825		
	601	332	933	-19.43	21.9
Department B					
sex	accepted	rejected	totals	excess	variance
f	17	8	25		
m	353	207	560		
	370	215	585	-1.19	5.57
Department C					
sex	accepted	rejected	totals	excess	variance
f	202	391	593		
m	120	205	325		
	322	596	918	6	47.86
Department D					
sex	accepted	rejected	totals	excess	variance
f	131	244	375		
m	138	279	417		
	269	523	792	-3.63	44.34
Department E					
sex	accepted	rejected	totals	excess	variance
f	94	299	393		
m	53	138	191		
	147	437	584	4.92	24.25
Department F					
sex	accepted	rejected	totals	excess	variance
f	24	317	341		
m	22	351	373		
	46	668	714	-2.03	10.75
totals				-15.36	154.69

The departments above are fixed effects and the Mantel-Haenszel deals with them well. There are other ways of analyzing data such as these, including logistic regression, which also handles the confounding effect of departments. The question we want to turn to is how to control for random effects, which naturally arise in meta-analyses when you want to combine the results from multiple published studies. For example consider the effect of high and low intakes of Omega-3 fatty acid on mortality. The next table summarizes the results of 3 cohort studies. How do we model the random study effect?

When the outcome of each study is fixed, Y_i is $N(\mu, \sigma_i^2)$. When there is a random study effect we model $Y_i \sim N(\theta_i, \sigma_i^2)$ where $\theta_i \sim N(\mu, \tau^2)$. Thus, $Y_i \sim N(\mu, \sigma_i^2 + \tau^2)$ assuming independence in the residual errors. We call τ^2 the heterogeneity variance.

Dolecek 1991					
mortality	high	low	totals	log odds ratio	variance
died	72	99	171		
lived	1179	1208	2387		
	1251	1307	2558	-0.29	0.025

Erkkila 2003					
mortality	high	low	totals	log odds ratio	variance
died	5	16	21		
lived	127	117	244		
	132	133	265	-1.25	0.207

Hu 2003					
mortality	high	low	totals	log odds ratio	variance
died	49	77	126		
lived	442	410	852		
	491	487	978	-0.52	0.036

Since the variances are not equal, due to differing within-study error variances, the best guess (an unbiased minimum-variance linear estimator) for the overall mean effect, μ is a weighted average of the within-sample estimates:

$$\hat{\mu} = \frac{\sum w_i Y_i}{\sum w_i} \text{ where } w_i = \frac{1}{\sigma_i^2 + \tau^2}$$

Even if the within-study variances can be estimated well, the between-study heterogeneity variance, τ^2 is not known. We can test whether it is zero or not because, under the null hypothesis that there is no heterogeneity variance, the statistic $Q(\mathbf{w}) = \sum w_i(Y_i - \hat{\mu})^2$ where the weights are taken as $w_i = 1/\hat{\sigma}_i^2$ for both this statistic and the calculation of $\hat{\mu}$. Under the hull hypothesis that $\tau^2 = 0$, $Q(w)$ is distributed as a Chi-square distribution with $k-1$ degrees of freedom where k is the number of studies.

Note that for *any* choice of weights,

$$\begin{aligned}
EQ(\mathbf{w}) &= \sum w_i E\left(Y_i - \frac{\sum w_j Y_j}{\sum w_j}\right)^2 \\
&= \sum w_i E\left(\frac{\sum w_j (Y_i - Y_j)}{\sum w_j}\right)^2 \\
&= \sum \frac{w_i}{\left(\sum w_j\right)^2} \sum\sum w_j w_l E\left((Y_i - \mu)^2 - 2(Y_i - \mu)(Y_j - \mu) + (Y_j - \mu)(Y_l - \mu)\right) \\
&= \sum w_i(\sigma_i^2 + \tau^2) - 2\sum \frac{w_i^2}{\sum w_j}(\sigma_i^2 + \tau^2) + \sum \frac{w_l^2}{\sum w_j}(\sigma_l^2 + \tau^2) \\
&= \sum \left(w_i - \frac{w_i^2}{\sum w_j}\right)(\sigma_i^2 + \tau^2)
\end{aligned}$$

If we choose $w_i = 1/\sigma_i^2$, then $EQ(\mathbf{w}) = (k-1) + \tau^2 \sum \left(w_i - \frac{w_i^2}{\sum w_j} \right)$ so that

$$\hat{\tau}^2 = \frac{Q(\mathbf{w}) - (k-1)}{\sum w_i - \frac{w_i^2}{\sum w_j}}$$

is an unbiased estimator of τ^2. Since we do not like variance estimates to be negative, we usually let $\hat{\tau}^2$ be the maximum of zero and the above estimate, which makes the estimate biased but more natural. This estimator is called the DerSimonian-Laird estimate for the heterogeneity variance after its authors.

After you estimate τ^2, you re-estimate the mean:

$$\hat{\mu} = \frac{\sum w_i Y_i}{\sum w_j} \text{ where } w_i = \frac{1}{\hat{\sigma}_i^2 + \hat{\tau}^2}$$

The estimate of the variance in $\hat{\mu}$ is $\hat{\text{Var}}(\hat{\mu}) = \frac{1}{\sum w_j}$ where $w_j = \frac{1}{\hat{\sigma}_j^2 + \hat{\tau}^2}$.

However, in practice, this estimate is biased. That is, the estimate of the variance underestimates the true variance, on average. And further, the normal distribution, while asymptotically valid, is rarely a good approximation in practice since the number of studies is often small.

Nominal Coverage is 95%			
plan	τ^2	Achieved Coverage for the DerSimonian Laird Confidence Interval	Achieved Coverage for Another Confidence Interval
A B C D E	0		
A B C D E	0.1		
A B C D E	1.0		
A B C D E	10.0		

Here, the plans need to be described, perhaps in their own table:

plan	Study Design						
A	sample sizes	10	20	30	10	20	30
	variances	1	1	1	1	1	1
B	sample sizes	10	20	30	10	20	30
	variances	1	3	5	1	3	5
C	sample sizes	10	20	30	10	20	30
	variances	5	3	1	5	3	1
D	sample sizes	20	20	20	20	20	20
	variances	1	1	1	1	1	1
E	sample sizes	10	20	30	10	20	30
	variances	1	3	5	1	3	5

How do statisticians determine such things? It is easy to guess that the variance estimate for $\hat{\mu}$ will underestimate the true variance because the estimate for τ^2 will obviously tend to overestimate the heterogeneity variance since it is a truncated estimator. After that, one typical procedure is to set up a simulation study to demonstrate the bias and its magnitude. For example, one might fill in a table like the one above giving various study designs as in the table that follows it.

The code for filling in the table might be similar to the following:

```
tau_DSL <- function(sample_sizes, sample_variances, sample_means) {
t = 0
weights <- 1/(sample_variances/sample_sizes)
sum_weights = sum(weights)
sum_weights_squared = sum(weights*weights)
mean_estimate = sum(weights*sample_means)/sum(weights)
Dt = sum(weights*(sample_means - mean_estimate)*(sample_means - mean_estimate))
Dt = Dt - (length(sample_sizes) - 1)
Dt = Dt/(sum_weights - sum_weights_squared/sum_weights)
t = t + Dt
if(t<0) t = 0
t
}
k = 6
C_DSL = array(0, dim=c(5))
true_sample_sizes <- array(0, dim=c(5,k))
true_variances <- array(0, dim=c(5,k))
true_sample_sizes[1,] <- c(10, 20, 30, 10, 20, 30)
true_sample_sizes[2,] <-c(10, 20, 30, 10, 20, 30)
true_sample_sizes[3,] <- c(10, 20, 30, 10, 20, 30)
true_sample_sizes[4,] <- c(20, 20, 20, 20, 20, 20)
true_sample_sizes[5,] <- c(20, 20, 20, 20, 20, 20)
true_variances[1,] <- c(1, 1, 1, 1, 1, 1)
```

```
true_variances[2,] <- c(1, 3, 5, 1, 3, 5)
true_variances[3,] <- c(1, 3, 5, 1, 3, 5)
true_variances[4,] <- c(1, 1, 1, 1, 1, 1)
true_variances[5,] <- c(1, 3, 5, 1, 3, 5)
sample_sizes <- array(0, dim=c(k))
sample_variances <- array(0, dim=c(k))
sample_means <- array(0, dim=c(k))

runs = 10000
for(outer_loop in 1:5) {
sample_sizes <- true_sample_sizes[outer_loop,]

tau = 0

for(i in 1:runs){

# this code efficiently generates the needed summary data
for(j in 1:k){

sample_variances[j] = rchisq(1, sample_sizes[j] -1)*true_variances[outer_loop,j]/
                                        ((sample_sizes[j]-1))
sample_means[j] = rnorm(1, 0.5,
(true_variances[outer_loop,j]/sample_sizes[j] + tau)^.5 )
}

#this code evaluates the DSL estimator
t <- tau_DSL(sample_sizes, sample_variances, sample_means)
weights <- 1/(sample_variances/sample_sizes + t)
sum_weights = sum(weights)
sum_weights_squared = sum(weights*weights)
mean_estimate = sum(weights*sample_means)/sum(weights)
mean_error = 1/sum(weights)
multiplier = qnorm(.975)
if(mean_estimate - multiplier*(mean_error^.5) < 0.5
&& 0.5 < mean_estimate + multiplier*(mean_error^.5))
C_DSL[outer_loop] = C_DSL[outer_loop] + 1
}
}
C_DSL = C_DSL/runs
print(tau)
print(C_DSL)
```

One would change the value for "tau" to match the entries in the table and re-run the code to fill in the table. More efficient possibilities are available as well.

So there are various possible "fixes" to this problem. One thing to note is that when the correct weights are known, $Q(\mathbf{w})$ and $\hat{\mu}$ for weights $w_i = 1/(\sigma_i^2+\tau^2)$ are independent and thus the distribution of

$$\frac{\sqrt{\sum w_i}(\hat{\mu}-\mu)}{\sqrt{Q(\mathbf{w})/(k-1)}}$$

is t with $k-1$ degrees of freedom. Thus a better confidence interval might be:

$$\hat{\mu} - \pm t^* \sqrt{Q(\mathbf{w})/(k-1)}/\sqrt{\sum w_i}$$

where the multiplier t^* comes from a t-distribution with $k-1$ degrees of freedom.

The weights above are theoretical. We have to estimate them in practice. If we estimate $\hat{\tau}^2$ as above, using the t-distribution improves but does not completely correct the anti-conservative nature of the confidence interval.

There are other possibilities for estimating $\hat{\tau}^2$. One is to notice that when the weights are "right", $EQ(\mathbf{w}) = k-1$ and to solve the equation $Q(\mathbf{w}) = k-1$ for τ^2 using the within-sample estimates of the variances σ_i^2 where $w_i = 1/(\sigma_i^2+\tau^2)$. (That was my idea.) One nice feature of this idea is that the standard error in $\hat{\mu}$ is what one expects since the strange multiplier, $Q(\mathbf{w})/(k-1)$, is 1 when τ^2 is estimated this way.

Another idea due to Hartung and Makambi works better:

Assume the weights are $w_i = 1/\sigma_i^2$. Note that

$$EQ(\mathbf{w}) = (k-1) + \tau^2 \sum \left(w_i - \frac{w_i^2}{\sum w_j} \right)$$

For the sake of notation, let A be the constant $\sum \left(w_i - \frac{w_i^2}{\sum w_j} \right)$. Take $Q(\mathbf{w})/A$ as a biased estimate of τ^2. We create a multiplicative correction, λ. for this bias and let

$$f(Q(\mathbf{w})) = \frac{\lambda Q(\mathbf{w})}{A}$$

We want $Ef(Q(\mathbf{w})) = \tau^2$. Using a first order Taylor series expansion, we see that

$$\tau^2 = Ef(Q(\mathbf{w})) \approx \frac{\lambda EQ(\mathbf{w})}{A} + \frac{\lambda}{A}\left(Q(\mathbf{w}) - EQ(\mathbf{w})\right) = \frac{\lambda}{A}\left(A\tau^2 + 2(k-1)\right)$$

Thus, the estimator

$$\frac{\tau^2}{2(k-1) + \tau^2 \sum \left(w_i - \frac{w_i^2}{\sum w_j} \right)} Q(\mathbf{w})$$

approximately has expected value τ^2. Now, in the above equation, replace τ^2 with its estimate

$$Q(\mathbf{w}) / \sum \left(w_i - \frac{w_i^2}{\sum w_j} \right)$$

where the weights are, in practice, $w_i = 1/\hat{\sigma}_i^2$. The new estimator for τ^2 is then:

$$\hat{\tau}^2 = \frac{\left(\sum w_i(Y_i - \hat{\mu})^2\right)^2}{\left(2(k-1) + \sum w_i(Y_i - \hat{\mu})^2\right)\left(\sum w_i - w_i^2/\sum w_j\right)}$$

where the weights are $w_i = 1/\hat{\sigma}_i^2$ and $\hat{\mu} = \sum w_i Y_i / \sum w_i$.

After estimating τ^2, re-estimate the mean $\hat{\mu}$, re-estimate $Q(\mathbf{w})$ using the weights $w_i = 1/(\hat{\sigma}_i^2 + \hat{\tau}^2)$, and form the confidence interval as

$$\hat{\mu} \pm t^* \sqrt{Q(\mathbf{w})/(k-1)} / \sqrt{\sum w_i}$$

where, again, t^* is taken from the t-distribution with $k-1$ degrees of freedom.

References and Readings

[1] P J Bickel, E A Hammel, and J W O'Connell. Sex bias in graduate admissions: Data from berkeley. *Science*, 187:398–403, 1975.

[2] R DerSimonian and Laird N. Meta-analysis in clinical trials. *Controlled Clinical Trials*, 7:177–188, 1986.

[3] J Hartung. An alternative method for meta-analysis. *Biometrical Journal*, 41:901–916, 1999.

[4] J Hartung and Makambi K. Reducing the number of unjustied signicant results in meta-analysis. *Communications in Statistics: Simulation and Computation*, 32:1179–1190, 2003.

[5] L Hooper, Thompson R L, Harrison R A, Summerbell C D, Ness A R, Moore H J, Worthington H V, Durrington P N, Higgins J P, Capps N E, Riemersma R A, Ebrahim S B, and Davey Smith G. Risks and benefits of omega 3 fats for mortality, cardiovascular disease, and cancer: systematic review. *BMJ*, 332:752–760, 2006.

Conceptual Exercises for Section 4.5

1.

♢ 4.6 Projects

Instructions

All projects should be written up as short research papers. You should present the data, the question, your methods, and your conclusions so that a bright researcher who is not necessarily a statistician can understand them.

Project 1: Design a simulation study to show that Hartung and Makambi's method for forming confidence intervals for the mean effect in the presence of heterogeneity variance is better than the original DerSimonian and Laird method. What are the confidence intervals provided by both methods for the effect of Omega-3 fatty acid on mortality given the results of the 3 studies presented in Section 4.5?

References and Readings

Chapter 5

Data Mining

♢ 5.1 Corrections for Conducting Too Many Tests

♢ 5.2 Benjamini and Hochberg's False Discovery Rate

♢ 5.3 Storey's Positive False Discovery Rate

♢ 5.4 Projects

Appendix A

R Software

♢ A.1 Obtaining, Installing, and Using R

R is freely available for Linux, MACs, and Windows from the R project WEB page :

http://www.r-project.org/

Go to the CRAN download page, elect a mirror site, and download the appropriate pre-compiled binary code. These steps will place the R software on your computer.

R is also available on the IU research computers (LIBRA) and in most university computer labs.

You are not required to use R for this course, but I will use it myself whenever I want to present code. The other natural statistics software to use for this course is SAS which is commonly used in the US government and frequently used in industry and academia. You can access SAS by getting a research computer account on LIBRA at IU, on Windows-based computers in the university computer labs, or by paying the Stat-Math center $110 to give you a user-license good for one year. SAS is available for Windows (and perhaps for Unix) this way.

Other statistics software packages you might be familiar with, including SPSS and Minitab, are not ideal for this course. While you may want to use some pre-programmed statistics algorithms such as drawing random samples for certain distributions or decomposing matrices into their eigenvalues and eigenvectors without writing the code to do this from scratch, most of what you will do in this course will involve writing your own code. While Minitab and SPSS allow for this, they are either slow or not very friendly about it.

If you choose to program in Java, Fortran, or C/C++, you will probably have faster code but you will have to work a little harder because some routines that are single commands in SAS and R will have to be programmed by you (or possibly imported from other sources.)

There are help manuals. The manual, "An Introduction to R," will probably suffice for this course. If you get serious about writing statistical software and want to be able to contribute packages to the R project, you should read the "Writing R Extensions" manual.

References and Readings

[1] The R project for statistical computing [http://www.r-project.org].
[2] W N Venables, Smith D M, and the R Development Core Team. *An Introduction to R.* The R Project, http://www.r-project.org, 2005.

♢ A.2 Basic R Commands and Sample R Code

One of the first hurdles to overcome is figuring out how to get data into R. When dealing with real problems, data sets are often large and in a file, such as an Excel file or a text file. To get them into R, you can use the "read.table" command. You should know whether your data file has a header line (to give the names of the variables) or not. You should also know where your file is located on your computer.

Say we have the following data in a file located at '∼/Desktop/data.txt':

```
x y
1 2
3 4
5 2
7 1
3 6
```

When you type the following into R, the variable "data" stores all of the data in the file.

```
data <- read.table("~/Desktop/data.txt", header=TRUE)
```

To assign the "x" data to a variable called x, type

```
x <- data[,1]
```

Similarly, to assign the "y" data to a variable called y, type

```
y <- data[,2]
```

To see the data x, simply type

```
x
```

Note that in R, " <- " means "assign." You can assign in either direction (that is, "->" works too.)

One can write to a file using similar language:

```
write.table(data, "~/Desktop/data_out.txt", col.names=TRUE, row.names=FALSE)
```

And you can get help on this command (and, similarly, on other R commnds) and see its list of options with a description of what the options do by typing

```
help(write.table)
```

You can get all the usual summary statistics for a vector of values by typing in the appropriate, obvious, command (mean(x), var(x), sd(x), median(x), min(x), max(x) and so forth.) You can also sort the vector using the sort(x) command.

You can also create your own functions in R. The following is how you would create a function called "stdev" to reproduce the "sd" function in R. You tell R that "stdev" will take one argument, x. The s alone on the last line inside the code for the function tells R to return the value of s when "stdev" is called.

```
stdev <- function(x){
n <- length(x)
s <- sum(x*x) - (sum(x)*sum(x)/n)
s <-  s/(n-1)
s <- sqrt(s)
s
}
```

You can type "stdev(x)" and "sd(x)" to be sure that you get the same result from both functions as you should.

The list of probability distributions built into R is given in the Introduction manual. The prefixes are "r" for random, "d" for the density, "p" for the cumulative distribution, and "q" for the quantile functions (the inverse CDF). When using "r" for random, the first argument is the sample size you want.

To demonstrate the "for" loop, we will put random deviates from a Normal distribution with mean 5 and standard deviation 2 into an array of length 10. It is silly to write such a "for" loop because the command

```
x <- rnorm(10, 5, 2)
```

does it all in one swoop. Nonetheless, here is the "for" loop routine:

```
x <- array(0, dim=c(10))
for(i in 1:10){
x[i] = rnorm(1, 5, 2)
}
```

There is a similar "while(conditional)" loop. Conditionals in R include

```
<       less than
<=      less than or equal to
```

```
>       greater than
>=      greater than or equal to
==      exactly equal
!=      not equal
&&      and
||      or
```

The following code uses the while construction and and if/else construction to count the how far x is above y by 2's.

```
check = 0
distance = 0
while(check==0){
if(x>y){
x = x - 2
distance = distance + 1}
else
{check=1}
}
print(distance)
```

For multivariate analyses, we will need to work with matrices. In R, matrices and arrays can be manipulated entry-by-entry or by matrix operations and the language has to distinguish between these possibilities.

To get a 2×2 matrix of 1's we can use the command

```
A <- array(1, dim=c(2,2))
```

If we create a 2×2 matrix by giving individual values, such as:

```
B = array(c(2, 3, 1, 2), dim=c(2,2))
```

then we find that the first column contains the 2 and 3. To get these numbers to be the first row instead, we can take the transpose:

```
B = t(B)
```

Or we could use the matrix command and fill in the data by rows:

```
B = matrix(c(2, 3, 1, 2), 2,2, byrow=TRUE)
```

If we let C be

```
C=A*B
```

we get pointwise multiplication so that $C = B$. If we want the usual matrix multiplication we have to use

```
C=A%*%B
```

instead. To get the Kronecker product, the command is

```
C=A%x%B
```

Matrix inversion should be done with care and uses the "solve" command. In general, when using an matrix inverse in a quadratic form, such as when computing $\mathbf{x}'\mathbf{B}^{-1}\mathbf{x}$, you would use "solve(B,x)" for the $\mathbf{B}^{-1}\mathbf{x}$ part. For example:

```
x = c(1, 1)
quadform=x%*%solve(B,x)
```

If you need the inverse of a matrix, it is given simply by "solve(matrix_name)".

R also has a powerful plotting capabilities. Suppose you define two functions f and g as follows:

```
f <- function(t){
response = 2 + t + t*t
response
}
g<- function(t){
response = 1 - t + t*t
response
}
```

You can overlay the plots of f and g using the following series of commands which sets wide enough limits on the horizontal, x, and vertical, y, ranges in the first plot to allow all of the function g in the second plot to be shown. If the labels (lab) for the y axis is set to nothing in the first plot and no labels are shown at all in the second plot (otherwise, the plots would overwrite each other's labels and that would be ugly.)

```
plot(f, -3, 3, ylab="", col="red", xlim=c(-4,4), ylim=c(0,20))
par(new=TRUE)
plot(g, -4, 4, ann=FALSE, axes=FALSE, col="blue")
```

You can also create fancy graphical layouts of several plots (up to 500 plots). For instance, say you want to create one histogram pointing up (for girls, say) and one pointing down (for boys, say) with the same horizontal scale. You first divide the output "device" (the screen on which the plot

appears) into a 4×4 grid, (this is the "2,2" part of the matrix command - you have 2 rows and 2 columns). You tell R to plot the first plot on the upper part of the screen and the second plot on the lower part of the screen (this is the c(1, 1, 2, 2) and byrow=TRUE part of the command - in the matrix the first plot (1) is placed in the top 2 columns, and the second plot (2) is placed lower two columns. To see the layout, we use the "layout.show" command.

```
nf <- layout(matrix(c(1,1, 2, 2),2,2,byrow=TRUE), respect=TRUE)
layout.show(nf)
```

Since we don't have data for boys and girls, let's create some:

```
boys <- rnorm(100, 1000, 140)
boyhist <- hist(boys, breaks=seq(600,1600,50),  plot=FALSE)
girls <- rnorm(100, 1000, 100)
girlhist <- hist(girls, breaks=seq(600,1600,50),  plot=FALSE)
```

Since we used the same breaks, the horizontal scale for boys and girls will be the same. We need to change the natural margins of the graphs to get the boy and girl histograms close together. The margins are mar=c(bottom, left, top, right) and the normal default it mar=c(5, 4, 4, 2) + 0.1. We need to ensure that the margins of the two graphs remain the same in order not to mess up the alignment and scales. To make the boy histogram pointing down, we will barplot the negative counts for boys.

```
par(mar=c(2,4,1,2))
barplot(girlhist$counts, width=1, space=0, ylim=c(0,25), xlim=c(0,20),
       axes=FALSE, main="girls")
axis(2, at=c(0, 5, 10, 15, 20, 25), labels =c("0", "5", "10", "15", "20", "25"))
axis(1, at=seq(0.5, 19.5, 2), labels =c("625", "725", "825", "925",
        "1025", "1125", "1225", "1325", "1425", "1525"))
par(mar=c(2,4,1,2))
barplot(-boyhist$counts, width=1, space=0, axes=FALSE,
              ylim=c(-25,0), xlim=c(0,20))
axis(2, at=c(0, -5, -10, -15, -20, -25), labels =c("0", "5", "10", "15",
          "20", "25"))
axis(3, at=seq(0.5, 19.5, 2),
        labels=c(" "," "," "," "," "," "," "," "," "," "," "))
title(main="boys", line=-13)
```

References and Readings

[1] W N Venables, Smith D M, and the R Development Core Team. *An Introduction to R.* The R Project, http://www.r-project.org, 2005.

Conceptual Exercises for Section A.2

1. What happens when you omit "row.names = FALSE" when writing the matrix data to an output file?
2. What happens when you use "write.csv" instead of "write.table"?
3. What is the outer product of the matrices A and B as in the notes in this section? How do you find it in R?
4. Why is "solve(B, x)" better than "solve(B)%*% x" ($B^{-1}x$)?

Appendix B

Useful Facts from Linear Algebra and Statistics

Multivariate Normal Transformations: If $\mathbf{y}$ is a multivariate normal vector with mean vector $\boldsymbol{\mu}$ and variance-covariance matrix $\mathbf{V}$ then the linear transformation $\mathbf{Ay}$ is multivariate normal with mean $\mathbf{A}\boldsymbol{\mu}$ and variance-covariance matrix $\mathbf{AVA}'$ where $\mathbf{A}'$ is the transpose of $\mathbf{A}$.

Multivariate Normal Density and Likelihood: The probability density for an n-dimensional vector $y \sim N(\boldsymbol{\mu}, \mathbf{V})$ is

$$f(\mathbf{y}) = \frac{1}{(2\pi)^{n/2}|\mathbf{V}|^{1/2}} e^{(\mathbf{y}-\boldsymbol{\mu})'\mathbf{V}^{-1}(\mathbf{y}-\boldsymbol{\mu})/2}$$

so the log likelohood function is

$$l(\mathbf{y}|\boldsymbol{\mu}, \mathbf{V}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{V}| - \frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})'\mathbf{V}^{-1}(\mathbf{y}-\boldsymbol{\mu})$$

Symmetric Positive Definite Matrices: Variance-covariance matrices are symmetric positive definite matrices. Such matrices have several useful decompositions. One is the Singular Value Decomposition that allows the factorization: $\mathbf{V} = \mathbf{PDP}'$ where $\mathbf{P}$ is the matrix of eigenvectors and $\mathbf{D}$ is a diagonal matrix with the eigenvalues along the diagonal. Since $\mathbf{P}' = \mathbf{P}^{-1}$, we see that $\mathbf{V}^{-1} = \mathbf{PD}^{-1}\mathbf{P}'$ and the square root of $\mathbf{V}$ can be defined as $\mathbf{PD}^{1/2}\mathbf{P}'$ where the square root of the diagonal matrix is just the diagonal matrix with the square root of the original entries along the diagonal.

Another is the Cholesky factorization that allows the factorization into an upper and a lower matrix: $\mathbf{V} = \mathbf{LL}'$ where $\mathbf{L}$ is a lower triangular matrix. We can use the Cholesky factorization to efficiently and robustly solve $\mathbf{Ax} = \mathbf{b}$ for $\mathbf{x}$ by first solving $\mathbf{Ly} = \mathbf{b}$ for $\mathbf{y}$ and then solving $\mathbf{L}'\mathbf{x} = \mathbf{y}$ for $\mathbf{x}$.

REML Estimates: REML stands for Restricted Maximum Likelihood. REML procedures remove the fixed effects from the data so that the data have mean zero. The degrees of freedom is dropped correspondingly. For instance, if $\mathbf{y}$ has mean $\mu\mathbf{1}$ and consists on n entries, we find a $(n-1) \times n$ matrix $\mathbf{K}$ so that $\mathbf{K1} = \mathbf{0}$ where $\mathbf{0}$ has $n-1$ entries. Then $\mathbf{Ky}$ has mean zero. Such matrices are not unique, there are many possible choices for $\mathbf{K}$ above. If the entries in $\mathbf{y}$ were independent, we would want to choose $\mathbf{K}$ so that $\mathbf{KK}' = \mathbf{I}$ where $\mathbf{I}$ is the $(n-1) \times (n-1)$ identity matrix.

The multivariate normal log likelihood function for the REML estimates when K has dimensions $(n-p) \times n$ is then

$$l(\mathbf{y}|\mathbf{V}, \mathbf{K}) = -\frac{n-p}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{KVK}'| - \frac{1}{2}(\mathbf{Ky})'(\mathbf{KVK}')^{-1}(\mathbf{Ky})$$

where p is the number of fixed effects removed from the data.

A standard and general way to transform the data to have mean zero is to use the design matrix $\mathbf{X}$ which indicates which fixed effects contribute to each response. Above $\mathbf{X} = \mathbf{1}$. We calculate

$$\mathbf{K} = \mathbf{I} - \mathbf{X}\text{ginv}(\mathbf{X}'\mathbf{X})\mathbf{X}'$$

where ginv is the generalized inverse. (The generalized inverse of a matrix satisfies $\mathbf{A}\text{ginv}(\mathbf{A})\mathbf{A} = \mathbf{A}$ and is often used in finding regression and ANOVA estimates.) Note that, here, $\mathbf{K}$ is an $n \times n$ matrix but it does not have full rank.

In this framework, the multivariate normal log likelihood function for the REML estimates is

$$l(\mathbf{y}|\mathbf{V}, \mathbf{X}) = -\frac{n-p}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{V}| - \frac{1}{2}\log|\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}| - \frac{1}{2}\mathbf{r}'(\mathbf{V})^{-1}\mathbf{r}$$

where $\mathbf{r} = \mathbf{y} - \mathbf{X}\text{ginv}(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})\mathbf{X}'\mathbf{V}^{-1}\mathbf{y}$ and p is the number of fixed effects removed (the rank of $\mathbf{X}$).

Likelihood Ratio Tests: Let $\omega \subset \Omega$ be regions in parameter space and $\boldsymbol{\theta}$ as points in parameter space. We think of ω as being the null hypothesis that some restriction on the parameter space is acceptable and Ω as the alternative hypothesis. The likelihood ratio test considers

$$\Lambda = 2\log\frac{\max\limits_{\Omega} P(\text{data}|\theta)}{\max\limits_{\omega} P(\text{data}|\theta)}$$

Under certain regularity conditions, Λ is Chi-Square distributed with degrees of freedom equal to the difference in the dimensions of the two spaces. The regularity conditions require that the likelihood function be twice-differentiable and that ω be interior to Ω.

References and Readings

[1] J Fan, Hung H-N, and Wong W-H. Geometric understanding of likelihood ratio statistics. *Journal of the American Statistical Association*, 95:836–841, 2000.

Appendix C

R Routines and Code for the Figures

The Golden Section Algorithm

```
GS <- function(funct, a, b, tol){
r = (sqrt(5) -1)/2 # find the golden section ratio
x = b - r*(b-a) # determine the first 2 points
y = a + r*(b-a)
fx = funct(x)
fy = funct(y)
steps = ceiling(log(2*tol/(b-a))/log(r)) # determine how many iterations are needed
for(i in 1:steps){
if(fy>fx){ # if f(y)>f(x) then the new interval is [a, y]
b=y
y=x
fy=fx
x = b - r*(b-a)
fx = funct(x)
}
else # the new interval is [x, b]
{
a=x
x=y
fx=fy
y = a + r*(b-a)
fy=funct(y)
}
} # end for
if(fx<fy)
{position = x
value = fx}
else
{position=y
value=fy}
response=list(position=position, value=value)
response
}
```

The Bracket Algorithm

```
bracket <- function(f, a, c, limit){
brace = c(0, 0, 0)
values=c(0,0,0)
fa = f(a)
fc = f(c)
if (f(c)>f(a)){ # reverse the roles of a and c so that we are going downhill
temp = a
a = c
c = temp
temp = fa
fa = fc
fc = temp
}
#Pick a point further out to try. The scaling factor is a bit arbitrary.
b = c + 2*(c-a)
fb = f(b)
#If this point works, then we are done. Let test indicate when we stop
test=0
if(fc < fb) test=1
iterations = 0
while(test==0)
{       # otherwise we fit a parabola through these points and
         # let u denote its minimum
denominator = (c-a)*(fc-fb) + (b-c)*(fc-fa)
# try not to divide by 0
if (abs(denominator) > 10**(-10)) {
numerator = (c-a)*(c-a)*(fc-fb) - (b-c)*(b-c)*(fc-fa)
u = c - .5*numerator/denominator}
else {u = b + 100*(b-c)} # if the denominator is very small,
                                          # look for a point far away.
# if u is not too far away, then determine where it is
u_bound = b + 100*(b-c) # u is either between c and b,
                                                # b and u_bound, or greater than u_bound
if ((c-u)*(u-b) > 0){ # if u is between c and b
fu = f(u)
if(fu < fb) { # then update the interval to be (a,c,b)= (c, u, b)
a = c
fa = fc
c = u
fc = fu
test = 1
} # end if (fu< fb)
else
```

```
{
if (fu > fc) {# then update the interval to be (a,c,b)= (a, c, u)
b = u
fb = fu
print("Warning: the function appears to decrease past an endpoint")
test = 1
}  # end if (fu > fc)
else
{ # the parabolic fit didn't work in the interval a to b
u = c + 2*(b-c)  # so try a scaling attempt
fu = f(u) # and update the interval to be (a, c, b) = (c, b, u)
a=c
fa=fc
c=b
fc=fb
b=u
fb = fu
if (fc < fb) test=1
} # end else
} # end else
}  # end if(c<u<b)
else             # if u is too far away, use the large scaling factor instead
{   # if u > c see if it is less than u_bound
if ((b-u)*(u-u_bound) > 0) # else if u < u_bound
{fu = f(u)       # use the parabolic estimate
a=c # and update the interval to be (a, c, b) = (c, b, u)
fa=fc
c=b
fc=fb
b=u
fb = fu
if (fc < fb) test=1
}
else
{
u = u_bound
fu = f(u)
a=c   # and update the interval to be (a, c, b) = (c, b, u)
fa=fc
c=b
fc=fb
b=u
fb = fu
if (fc < fb) test=1
} # end else
```

```
} # end if
iterations = iterations + 1
if (iterations> limit && test==0)
{ test = 1
print("No bracketing interval was found within the number of iterations allowed")
}
}  # end while

brace = c(a, c, b)
values = c(fa, fc, fb)
o <- order(brace)
brace = brace[o]
values = values[o]
response=list(brace=brace, values=values)
response
} # end bracket routine
```

Brent's Algorithm

```
Brent <- function(f, brace, tolerance=.Machine$double.eps^.25){
# INITIALIZATIONS
tolerance=max(tolerance, .Machine$double.eps)
eps = sqrt(.Machine$double.eps)
values = brace$values
brace = brace$brace
o = order(brace)
values = values[o]
brace = brace[o]
r = (sqrt(5) - 1)/2
a = brace[1]
c= brace[2]
b= brace[3]
x = c # the point with the least function value so far
w = x # the point with the second least function value
v=w #previous value of w
e = 0.0 #distance moved on the step before last
d = 0.0
fx = f(x)
fw = fx
fv = fx
error = (b-a)/2
xm = (a + b)/2
tol = eps*abs(x) + tolerance/3
```

```
while ( error > 2*tol - .5*(b-a) ){
# find the numerator and denominator for the parabolic point
numerator = (x-v)*(x-v)*(fx-fw) - (x-w)*(x-w)*(fx-fv)
denominator = (x-v)*(fx-fw) + (w-x)*(fx-fv)
if (denominator > 0) numerator = (-1)* numerator
denominator = abs(denominator)
etemp = e
e = d
# use these values to determine whether to take a golden section step instead
if( (abs(numerator) >= abs(.5*denominator*etemp))
|| (numerator <= denominator*(a-x))
|| (numerator>=denominator*(b-x)) ){
  # if you don't move enough  or if you move outside (a,b)
 # take golden section step
 if(x > xm) e = a - x
 else e = b-x
 d = (1-r)*e
}
else{
d = numerator/denominator
if( (x + d -a < 2*tol) || (b-x-d < 2*tol) ) d = sign(d)*tol
} # end else take parabolic step
if (abs(d)> tol) u = x + d
else u = x + sign(d)*tol
fu = f(u)
if (fu <= fx) { #bracket (a, x, b) becomes either (x, u, b) or (a, u, x)
 if(u>=x) a = x
 else b=x
 v = w
 fv = fw
 w = x
 fw = fx
 x = u
 fx = fu
}#end if fu < fx
else
{if(u<x) a = u #bracket (a,x, b) becomes either (u, x, b) or (a, x, u)
else b = u
if (fu <=fw || w==x){
v = w
fv = fw
w = u
fw = fu
}
else{
```

```
if (fu <= fv || v == x || v == w){
v = u
fv = fu
}
}# end else
}# end else if fu > fx
xm = (a + b)/2
error = x - xm
tol = eps*abs(x) + tolerance
}# end while
response = list(position=x, value=fx)
response
}#end Brent
```

Powell's Algorithm

The Simplex Method

The EM Algorithm (applied to a mixture of 2 normal distributions)

Figure 1.4

```
f <- function(t){
t = t**1.2
t <- (3*t-2.6)/2
response = 2 + t - t*t + t*t*t*t
response
}
r = (sqrt(5) - 1)/2
a = 0
b = 1
limx = c(0,1)
limy=c(0, 2.2)
plot(f, a, b, col="black", ylab="", xlab="",axes=FALSE,
        xlim=limx, ylim=limy, lwd=2)
```

```
x = a + r*(b-a)
y = f(x)
points( x, y, type = "p", pch = 21,
       col = "black", bg = NA, cex = 2)
x = b - r*(b-a)
y = f(x)
points( x, y, type = "p", pch = 21,
       col = "black", bg = NA, cex = 2)

a = 0
b = r
par(new=TRUE)
plot(f, a, b, ylab="", xlab="", col="red", axes=FALSE,
          xlim=limx, ylim=limy, lty=3, lwd=2)
x = a + r*(b-a)
y = f(x)
points( x, y, type = "p", pch = 22,
       col = "red", bg = NA, cex = 3)
x = b - r*(b-a)
y = f(x)
points( x, y, type = "p", pch = 22,
       col = "red", bg = NA, cex = 3)

a = x
b = b
par(new=TRUE)
plot(f, a, b, ylab="", xlab = "", col="blue", axes=FALSE,
            xlim=limx, ylim=limy, lty=2, lwd=3)
x = a + r*(b-a)
y = f(x)
points( x, y, type = "p", pch = 24,
       col = "blue", bg = NA, cex = 3)
x = b - r*(b-a)
y = f(x)
points( x, y, type = "p", pch = 24,
       col = "blue", bg = NA, cex = 3)
axis(1, at=c(0, r*r, r, 1), col="black",
         labels=c("a", "x", "y", "b"), pos=.6, padj=-1)
axis(1, at=c(0, r*r*r, r*r, r), col="red",
         labels=c("a", "x", "y", "b"), pos=.3, padj=-1)
axis(1, at=c(r*r*r, r*r, r*(r-r*r*r) + r*r*r, r), col="blue",
         labels=c("a", "x", "y", "b"), pos=0, padj=-1)
axis(2, at=c(.6, 2.3), labels=FALSE, pos=0)
```

Figure 1.5 (a) and (b)

```
#a
x <- y <- seq(-5, 5, len = 100)
r <- outer(x^2, 5*y^2, "+")
limx = c(-5, 5)
limy = c(-5,5)
contour(x, y, r, drawlabels = TRUE, levels=c(.25, 2.5, 5, 10, 15),
                axes=FALSE, frame=FALSE, xlim=limx, ylim=limy, lwd=3, labcex=1)
axis(1, col="black", pos=0, labels=FALSE, tck=0)
axis(2, col="black", pos=0, labels=FALSE, tck=0)
arrows(2, 3*sqrt(1/5), 2, 0, col="red", lwd=2, length=.1)
arrows(2, 0, 0, 0, col="blue",lwd=2, length=.1)

#b
x <- y <- seq(-5, 5, len = 100)
r <- array(0, dim=c(100,100))
for (i in 1:100){
for (j in 1:100){
r[i,j] = (x[i]+y[j])^2/2 +5*(x[i]-y[j])^2/2
}}
limx = c(-6, 6)
limy = c(-6,6)
contour(x, y, r, drawlabels = FALSE, levels=c(.25, 2.5, 5, 10, 15),
                axes=FALSE, frame=FALSE, xlim=limx, ylim=limy, lwd=3, labcex=1)
axis(1, col="black", pos=0, labels=FALSE, tck=0)
axis(2, col="black", pos=0, labels=FALSE, tck=0)
arrows(2, 3, 2, 4/3, col="red", lwd=2, length=.1)
arrows(2, 4/3, 8/9, 4/3, col="blue",lwd=2, length=.1)
arrows(8/9, 4/3, 8/9, 3/4, col="red", lwd=2, length=.1)
```

Figure 1.6 (a) and (b)

```
#a
x <- y <- seq(-5, 5, len = 100)
r <- array(0, dim=c(100,100))
for (i in 1:100){
for (j in 1:100){
r[i,j] = (x[i]+y[j])^2/2 +5*(x[i]-y[j])^2/2
}}
limx = c(-6, 6)
limy = c(-6,6)
contour(x, y, r, drawlabels = FALSE, levels=c(.25, 2.5, 5, 10, 15),
                axes=FALSE, frame=FALSE, xlim=limx, ylim=limy, lwd=3, labcex=1)
axis(1, col="black", pos=0, labels=FALSE, tck=0)
axis(2, col="black", pos=0, labels=FALSE, tck=0)
```

```
arrows(2, 3, 2, 4/3, col="red", lwd=2, length=.1)
arrows(2, 4/3, 8/9, 4/3, col="blue",lwd=2, length=.1)
arrows(2, 3, 0,  0, col="yellow3",lwd=2, length=.1)

#b
limx = c(0, 6)
limy = c(0,6)
plot(0, 0, xlim=limx, ylim=limy, xlab="", ylab="", axes=FALSE, cex=1)
points(5.5, 5.5, type="p", cex=2)
text(5.6, 5.8, expression(P[0]))
text(5.9, 5, expression(u[1]))
text(6, 4, expression(P[1]))
text(5, 3.6, expression(u[2]))
text(4.2, 4, expression(P[2]))
text(3.9, 2.4, expression(P[0]))
text(3.3, 2.8, expression(u[1]))
text(2.6, 2.4, expression(u[2]))
axis(1, at=c(-6, 6), col="black", pos=0, labels=FALSE, tck=0)
axis(2, at=c(-6, 6), col="black", pos=0, labels=FALSE, tck=0)
arrows(5.5, 5.5, 5.5, 4, col="red", lwd=2, length=.1)
arrows(5.5, 4, 4.5, 4, col="blue",lwd=2, length=.1)
arrows(5.5, 5.5, 5.5 - 2, 5.5 - 2*1.5, col="darkolivegreen",lwd=2, length=.1)
arrows(3.5, 2.5, 3, 2.5,  col="blue",lwd=2, length=.1)
arrows(3, 2.5, 3-.5, 2.5 - 1.5/2, col="darkolivegreen",lwd=2, length=.1)
arrows(3.5, 2.5, 3.5 - 2, 2.5 - 1.5, col="yellow3",lwd=2, length=.1)
arrows( 1.5, 1,  1.5, .5, col="red",lwd=2, length=.1)
arrows(1.5, .5, 1, .5, col="blue",lwd=2, length=.1)
```

Figure 1.4

```
nf<-layout(matrix(c(1,2,3,4), 2, 2, byrow=TRUE),
       respect=TRUE, widths=c(lcm(15), lcm(15)), heights=c(lcm(15), lcm(15)) )

layout.show(nf)

nf<-layout(matrix(c(1,2,3,4), 2, 2, byrow=TRUE), respect=TRUE)
# Reflection
limx = c(-6, 6)
limy = c(-6,6)
x=c(-5, -2, 5, -5)
y = c(0, 3, 0, 0)
plot(x, y, type="l", xlab="", ylab="", axes=FALSE, frame=FALSE,
        xlim=limx, ylim=limy, lwd=2)
par(new=TRUE)
```

```
plot(0, 0, type="p", xlab="", ylab="", axes=FALSE, frame=FALSE,
         xlim=limx, ylim=limy, cex=2)
par(new=TRUE)
plot(c(-2, 2),  c(3, -3), type="l", xlab="", ylab="", axes=FALSE, frame=FALSE,
          xlim=limx, ylim=limy, lwd=2, lty=2)
par(new=TRUE)
x=c(-5, 2, 5, -5)
y = c(0, -3, 0, 0)
plot(x, y, type="l", xlab="", ylab="", axes=FALSE, frame=FALSE,
         xlim=limx, ylim=limy, lwd=2, lty=3)
text(5.5, -.7, "best")
text(5.6, .1, expression(f(x[1])))
text(-5.6, -.7, "second")
text(-5.8, -1.3, "worst")
text(-5.6, .1, expression(f(x[2])))
text(-2, 4.5, "worst")
text(-2, 3.6, expression(f(x[3])))
title(main="Reflection")
#Expansion
limx = c(-6, 6)
limy = c(-6,6)
x=c(-5, -2, 5, -5)
y = c(0, 3, 0, 0)
plot(x, y, type="l", xlab="", ylab="", axes=FALSE, frame=FALSE,
         xlim=limx, ylim=limy, lwd=2)
par(new=TRUE)
plot(0, 0, type="p", xlab="", ylab="", axes=FALSE, frame=FALSE,
        xlim=limx, ylim=limy, cex=2)
par(new=TRUE)
plot(c(-2, 4),  c(3, -6), type="l", xlab="", ylab="", axes=FALSE, frame=FALSE,
        xlim=limx, ylim=limy, lwd=2, lty=2)
x=c(-5, 4, 5, -5)
y = c(0, -6, 0, 0)
par(new=TRUE)
plot(x, y, type="l", xlab="", ylab="", axes=FALSE, frame=FALSE,
       xlim=limx, ylim=limy, lwd=2, lty=5)
text(5.6, -.7, "best")
text(5.6, .1, expression(f(x[1])))
text(-5.6, -.7, "second")
text(-5.8, -1.3, "worst")
text(-5.6, .1, expression(f(x[2])))
text(-2, 4.5, "worst")
text(-2, 3.6, expression(f(x[3])))
title(main="Expansion")
#Contraction
```

```
limx = c(-6, 6)
limy = c(-6,6)
x=c(-5, -2, 5, -5)
y = c(0, 3, 0, 0)
plot(x, y, type="l", xlab="", ylab="", axes=FALSE, frame=FALSE,
        xlim=limx, ylim=limy, lwd=2)
par(new=TRUE)
plot(0, 0, type="p", xlab="", ylab="", axes=FALSE, frame=FALSE,
        xlim=limx, ylim=limy, cex=2)
par(new=TRUE)
plot(c(-2, -1),  c(3, 1.5), type="l", xlab="", ylab="", axes=FALSE, frame=FALSE,
        xlim=limx, ylim=limy, lwd=2, lty=2)
par(new=TRUE)
x=c(-5, -1, 5, -5)
y = c(0, 1.5, 0, 0)
plot(x, y, type="l", xlab="", ylab="", axes=FALSE, frame=FALSE,
         xlim=limx, ylim=limy, lwd=3, lty=3)
text(5.5, -.7, "best")
text(5.6, .1, expression(f(x[1])))
text(-5.6, -.7, "second")
text(-5.8, -1.3, "worst")
text(-5.6, .1, expression(f(x[2])))
text(-2, 4.5, "worst")
text(-2, 3.6, expression(f(x[3])))
title(main="Contraction")
#Shrinkage
limx = c(-6, 6)
limy = c(-6,6)
x=c(-5, -2, 5, -5)
y = c(0, 3, 0, 0)
plot(x, y, type="l", xlab="", ylab="", axes=FALSE, frame=FALSE,
        xlim=limx, ylim=limy, lwd=2)
par(new=TRUE)
plot(0, 0, type="p", xlab="", ylab="", axes=FALSE, frame=FALSE,
        xlim=limx, ylim=limy, cex=2)
par(new=TRUE)
plot(5, 0, type="p", xlab="", ylab="", axes=FALSE, frame=FALSE,
         xlim=limx, ylim=limy, cex=2)
par(new=TRUE)
plot(0, 0, type="p", xlab="", ylab="", axes=FALSE, frame=FALSE,
         xlim=limx, ylim=limy, cex=2)
par(new=TRUE)
plot(1.5, 1.5, type="p", xlab="", ylab="", axes=FALSE, frame=FALSE,
        xlim=limx, ylim=limy, cex=2)
par(new=TRUE)
```

```
x=c(0, 1.5, 5, 0)
y = c(0, 1.5, 0, 0)
plot(x, y, type="l", xlab="", ylab="", axes=FALSE, frame=FALSE,
        xlim=limx, ylim=limy, lwd=3, lty=3)
text(5.5, -.7, "best")
text(5.8, .1, expression(f(x[1])))
text(-5.6, -.7, "second")
text(-5.8, -1.3, "worst")
text(-5.6, .1, expression(f(x[2])))
text(-2, 4.5, "worst")
text(-2, 3.6, expression(f(x[3])))
title(main="Shrinkage")
```

Index

Jöreskog, K. G. , 8

Keesling, J. W., 8

Mixed Models, 1
    Pedigree Analysis, 4
    Phylogenetic Analysis, 5

R software
    Conditionals, 49
    For, 49
    If/else, 50
    Input, 48
    Kronecker product, 51
    Matrices, 50
    Matrix inversion, 51
    Output, 48
    While, 49
Relationship Matrix
    Example, 4

Structural Equation Models
    Mathematical formulation, 8

Wheaton study, 8
Wiley, D. E., 8