Applied Statistics Review

(Some things you should already know)

1. Hypothesis testing and large-sample likelihood ratio tests.
2. Basics of S under unix.
3. Normal linear model
4. Generalized linear models, esp. logistic regression
5. Basic Bootstrapping

1. Hypothesis testing and likelihood ratio tests

We will adopt the following model for observed data. The distribution of $\mathbf{Y} = (Y_1, ..., Y_n)$ is considered known except for some **parameter** θ, which may be a vector $\theta = (\theta_1, ..., \theta_k)$; $\theta \in \Theta$, the **parameter space**. The parameter space will usually be an open set. If $\mathbf{Y}$ is a continuous random variable, its **probability density function** (pdf) will de denoted $f(\mathbf{y};\theta)$. If $\mathbf{Y}$ is discrete then $f(\mathbf{y};\theta)$ represents the **probability mass function** (pmf); $f(\mathbf{y};\theta) = P_\theta(\mathbf{Y}=\mathbf{y})$.

A **statistical hypothesis** is a statement about the value of θ. We are interested in testing the null hypothesis H_0: $\theta \in \Theta_0$ versus the alternative hypothesis H_1: $\theta \in \Theta_1$. Where Θ_0 and $\Theta_1 \subset \Theta$. Naturally $\Theta_0 \cap \Theta_1 = \varnothing$, but we need not have $\Theta_0 \cup \Theta_1 = \Theta$. A **hypothesis test** is a procedure for deciding between H_0 and H_1 based on the sample data. It is equivalent to a **critical region**: a critical region is a set $C \subset \mathbb{R}^n$ such that if $\mathbf{y} = (y_1, ..., y_n) \in C$, H_0 is rejected. Typically C is expressed in terms of the value of some **test statistic**, a function of the sample data. For example, we might have $C = \{(y_1, ..., y_n)$: $\frac{\bar{y} - \mu_0}{s / \sqrt{n}} \geq 3.324\}$. The number 3.324 here is called a **critical value** of the test statistic $\frac{\bar{Y} - \mu_0}{S / \sqrt{n}}$.

If $\mathbf{y} \in C$ but $\theta \in \Theta_0$, we have committed a Type I error. If $y \notin C$ but $\theta \in \Theta_1$, we have committed a Type II error. The ideal hypothesis test would simultaneously minimize the probabilities of both types of error, but this turns out to be impossible in principle. So what we do is to select a **significance level** $\alpha = \underset{\theta \in \Theta_0}{\text{Max}} P_\theta(Y \in C)$ to be a small number; $\alpha = 0.05$ and $\alpha = 0.01$ are traditional. Then a "good" test is one with a small probability of Type II error, among all possible tests with significance level α. When $\mathbf{y} \in C$ for a test of level α (that is, H_0 is rejected), we say the results are **statistically significant** at level α.

For a particular set of data, the smallest significance level that leads to the rejection of H_0 is called the **p–value**. H_0 is rejected if and only if $p \leq \alpha$.

$P_\theta(Y \in C)$ can be viewed as a function of θ. If $\theta \in \Theta_1$, we refer to this quantity as the **power** of the test C. For any good test, we will have $P_\theta(Y \in C) \uparrow 1$ as $n \rightarrow \infty$ for each $\theta \in \Theta_1$. This provides a way to choose sample size. For a fixed value $\theta \in \Theta_1$ that is of scientific interest, choose n large enough so that the probability of rejecting H_0 is acceptably high.

How do we construct good hypothesis tests? Usually it is hard to beat the **likelihood ratio tests**, which are defined as follows. $C = \{\mathbf{y}: \lambda = \dfrac{\underset{\theta \in \Theta_0}{\text{Max}} L(\theta;\mathbf{y})}{\underset{\theta \in \Theta}{\text{Max}} L(\theta;\mathbf{y})} \leq k\}$. The value of k ($0<k<1$) varies from problem to problem. It is chosen so that the test will have significance level ("size") α.

Notice that the denominator of λ is just the likelihood function evaluated at the MLE. The numerator is the likelihood function evaluated at a sort of restricted MLE, in which θ is forced to stay in the set Θ_0. Also notice that when $\lambda=0$, H_0 is always rejected, and when $\lambda=1$, H_0 is never rejected.

There are two main versions of the likelihood ratio approach. One version leads to exact tests, and the other leads to large–sample approximations. The **exact likelihood ratio tests** are obtained by working on the critical region C, and re–expressing it in terms of the value of some test statistic whose distribution (given that H_0 is true) is known exactly. This is where we get most of the standard statistical tests, including t–tests and F–tests in regression and the analysis of variance.

Sometimes, after the critical region C has been re–expressed in terms of some seemingly convenient test statistic, nobody can figure out its distribution under H_0. This means we are unable to choose $k \in (0,1)$ so that the test has size α. And if the MLE has to be approximated numerically, there is little hope of an exact test. In such cases we resort to **large–sample likelihood ratio tests**; we will use the following result.

Let $\theta = (\theta_1, ..., \theta_p)$; we want to test H_0: $\theta \in \Theta_0$ versus H_1: $\theta \in \Theta_1$, where $\Theta_0 \cup \Theta_1 = \Theta$. Let $r \leq p$ be the number of parameters θ_j whose values are restricted by H_0. Then under some smoothness conditions, $G = -2 \log(\lambda)$ has (for large n) an approximate χ^2 distribution with r degrees of freedom. We reject H_0 when G is greater than the critical value of the χ^2 distribution. If g is the observed value of the statistic G calculated from the sample data, the p–value is $1-\gamma(g)$, where γ is the cumulative distribution function of a χ^2 distribution with r degrees of freedom.

Generalized Linear Models, esp. Logistic Regression

According to the logistic regression model, $Y_1, ..., Y_n$ are independent Bernoulli $B(1,\theta(\mathbf{x}_i))$ random variables, with a linear model for the log odds — that is, $\log(\frac{\theta(\mathbf{x}_i)}{1-\theta(\mathbf{x}_i)}) = \mathbf{x}_i' \boldsymbol{\beta} =$ $\beta_0 + \beta_1 x_{i,1} + ... \beta_p x_{i,p} \Leftrightarrow \theta(\mathbf{x}_i) = \frac{e^{\mathbf{x}_i'\boldsymbol{\beta}}}{1+e^{\mathbf{x}_i'\boldsymbol{\beta}}}$. We estimate $\boldsymbol{\beta}$ using maximum likelihood, and test hypotheses about $\boldsymbol{\beta}$ using large–sample likelihood ratio tests. The interpretation of β is that if x_j is increased by c units with all other independent variables being held constant, the odds of Y=1 are multiplied by $e^{c\beta_j}$. The quantity $e^{c\beta_j}$ is called an *odds ratio.* Categorical variables are represented by dummy variables and interactions are represented by product terms, as in ordinary regression. That's basically all there is to it.

Splus glm Lesson 1

```
S-PLUS : Copyright (c) 1988, 1992 Statistical Sciences, Inc.
S : Copyright AT&T.
Version 3.1 Release 1 for Sun SPARC, SunOS 4.x : 1992
Working data will be in .Data
> ! rm .Data/*
> data <- scan("bweight.dat",list(id=0, low=0, age=0, lwt=0, race=0, smoke=0,
+ ptl=0, ht=0, ui=0, ftv=0, bwt=0))
> low <- data$low ; age <- data$age ; lwt <- data$lwt ; race <- data$race
> > > > ftv <- data$ftv
> racefac <- factor(race,levels=c(1,2,3),label=c("White","Black","Other"))
> # Makes a FACTOR corresponding to race, like declaring it categorical.
> # The levels parameter can be omitted.  So can labels, but labels are helpful.
> redmod <- glm( low ~ lwt + racefac, family=binomial ) # Reduced model
> summary(redmod)

Call: glm(formula = low ~ lwt + racefac, family = binomial)
Deviance Residuals:
       Min         1Q     Median       3Q     Max
 -1.349029 -0.8918671 -0.7196572 1.252624 2.09923

Coefficients:
                  Value  Std. Error    t value
(Intercept)  1.32592752 0.839660487  1.5791234
        lwt -0.01521982 0.006395764 -2.3796713
   racefac1  0.54050127 0.243490714  2.2198024
   racefac2 -0.01997056 0.121672546 -0.1641337

(Dispersion Parameter for Binomial family taken to be 1 )

    Null Deviance: 234.672 on 188 degrees of freedom

Residual Deviance: 223.2591 on 185 degrees of freedom

Number of Fisher Scoring Iterations: 3

Correlation of Coefficients:
         (Intercept)        lwt   racefac1
     lwt -0.9766815
racefac1  0.2903805  -0.2052574
racefac2 -0.3348159   0.2890645 -0.3695642

>  # How are dummy variables being set up?
> contrasts(racefac)
      [,1] [,2]
White   -1   -1
Black    1   -1
Other    0    2
> # Default setup for dummy vars is Helmert contrasts.
> # Indicator dummy vars are called "treatment" contrasts; assigned with cap C
> racefac <- C(racefac,treatment)
> redmod <- glm( low ~ lwt + racefac, family=binomial ) # Writes over old one
> summary(redmod)

Call: glm(formula = low ~ lwt + racefac, family = binomial)
Deviance Residuals:
       Min         1Q     Median       3Q     Max
 -1.349029 -0.8918671 -0.7196572 1.252624 2.09923



Coefficients:
                   Value  Std. Error    t value
 (Intercept)  0.80539681 0.840773379  0.9579238
         lwt -0.01521982 0.006395764 -2.3796713
racefacBlack  1.08100255 0.486981428  2.2198024
racefacOther  0.48058959 0.356136276  1.3494542

(Dispersion Parameter for Binomial family taken to be 1 )

    Null Deviance: 234.672 on 188 degrees of freedom
Residual Deviance: 223.2591 on 185 degrees of freedom
Number of Fisher Scoring Iterations: 3

Correlation of Coefficients:
             (Intercept)        lwt racefacBlack
         lwt -0.9577774
racefacBlack  0.0538743  -0.2052574
racefacOther -0.3445053   0.1559385  0.3049205

> 234.672-223.2591  #  This is G for both vars
[1] 11.4129
> fullmod <- update(redmod,. ~ . + age + ftv)  #  Enter 2 more vars
> summary(fullmod)

Call: glm(formula = low ~ lwt + racefac + age + ftv, family = binomial)
Deviance Residuals:
       Min         1Q     Median       3Q      Max
 -1.416224 -0.8931506 -0.7113406 1.245424 2.075383

Coefficients:
                   Value Std. Error    t value
 (Intercept)  1.29495356 1.06680088  1.2138662
         lwt -0.01424125 0.00649708 -2.1919464
racefacBlack  1.00382939 0.49676062  2.0207507
racefacOther  0.43310052 0.36161349  1.1976890
         age -0.02382119 0.03363897 -0.7081426
         ftv -0.04929883 0.16668883 -0.2957537

(Dispersion Parameter for Binomial family taken to be 1 )
    Null Deviance: 234.672 on 188 degrees of freedom
Residual Deviance: 222.5729 on 183 degrees of freedom

Number of Fisher Scoring Iterations: 3
```

```
Correlation of Coefficients:
             (Intercept)        lwt racefacBlack racefacOther        age
         lwt -0.6475215
racefacBlack -0.0713275  -0.2318403
racefacOther -0.3457582   0.1337497  0.3161780
         age -0.6088156  -0.1520083  0.1860453    0.1047949
         ftv  0.0310256  -0.0497851 -0.0030391    0.0955343   -0.1687550

> G <- redmod$deviance - fullmod$deviance  #  See  help(glm.object)
> p <- 1-pchisq(G,df=2)
> G ; p
[1] 0.6861841
> [1] 0.7095729
> # Predicted prob of lbw baby for a 125 lb white woman aged 27, 3 visits
> xb <- sum( coefficients(fullmod)*c(1,125,0,0,27,3) )
> exp(xb)/(1+exp(xb))
[1] 0.2181856
```

To get the job done, we could have used just the following:

```
data <- scan("bweight.dat",list(id=0, low=0, age=0, lwt=0, race=0, smoke=0,
ptl=0, ht=0, ui=0, ftv=0, bwt=0))
low <- data$low ; age <- data$age ; lwt <- data$lwt
race <- data$race ; ftv <- data$ftv
racefac <- C(factor(race,label=c("White","Black","Other")),treatment)
redmod <- glm( low ~ lwt + racefac, family=binomial ) # Reduced model
fullmod <- update(redmod,. ~ . + age + ftv)  #  Enter 2 more vars
summary(redmod); summary(fullmod)
G <- redmod$deviance - fullmod$deviance ; p <- 1-pchisq(G,df=2)
G ; p ; xb <- sum( coefficients(fullmod)*c(1,125,0,0,27,3) )
exp(xb)/(1+exp(xb))
```

I would put it in a file (for example, called fname) and use

```
      source("fname")    and
      ! emacs fname
```

repeatedly until I was satisfied with the results. (Of course if you are using X-windows, it makes sense to have Splus and your text editor running in separate windows, and just go back and forth using the mouse.)

Exponential families and Generalized Linear Models

A random variable is said to belong to the exponential family provided it has a density or probability mass function of the form $f(y;\theta,\phi) = \mathrm{Exp}\{\frac{y\theta - b(\theta)}{a(\phi)} - c(y,\phi)\}I_A(y)$, where a, b, and c are functions that depend only on the arguments shown, and the support set A does not depend on either θ or ϕ; θ is called the *natural parameter*, and ϕ is called the *dispersion parameter*. If we denote E[Y] by μ, the function g relating θ to μ by $\theta = g(\mu)$ is called the *natural link function*. If we denote Var[Y] by σ^2, we can write $\sigma^2 = \phi V(\mu)$; the function $V(\mu)$ is called the *variance function*.

The basic idea behind a generalized linear model is to adopt a linear model for $\theta = g(\mu)$, and do estimation by maximum likelihood and testing by large–sample likelihood ratio tests. It is possible to adopt a linear model for some other function of μ, besides g. That is, we can employ a link function other than the "natural" one.

Remarkably, the standard algorithm for estimating the βs (iteratively re–weighted least squares) depends only on the link function and the variance function, not on the exact form of $f(y;\theta,\phi)$. Consequently, one can analyze data using a generalized linear model by specifying ONLY the link function and the variance function.

Poisson Regression with the S glm function

In this example, a national (U.S.) chain of home renovation supply stores (something like Home Depot) obtains data for a sample of U.S. census tracts. For each census tract, they measure

- Number of housing units
- Average income in dollars
- Average housing unit age, in years
- Average distance to competitor's nearest store, in miles
- Average distance to our nearest store, in miles
- Number of customers during 2-week period

The dependent variable is number of customers. Management wants to know whether distance to our nearest store and distance to competitor's nearest store (considered jointly) makes any difference once we allow for number of housing units, average income and average housing unit age -- including the intereraction between age and income.

```
# Poisson Regression
renodat <-
scan("data/lumber.dat",list(ctract=0,income=0,age=0,cdist=0,dist=0,customers=0))
income <- renodat$income
age <- renodat$age
cdist <- renodat$cdist
dist <- renodat$dist
customers <- renodat$customers
ageinc <- age*income # Interaction term
# First, always look at the raw data. Then do descriptive statistics. NEVER
# assume that the data are okay.
X11()
# postscript("describe.ps",horizontal=F) # Portait mode
hist(income) ; hist(age) ; hist(dist) ; hist(cdist) ; hist(customers)
boxplot(dist,cdist)
pairs(cbind(income,age,dist,cdist,customers)) # Scatterplot matrix
cor(cbind(income,age,dist,cdist,customers))

> cor(cbind(income,age,dist,cdist,customers))

              income         age        dist       cdist   customers
income     1.0000000  0.62324747  0.25723238  0.12689036 -0.26448803
age        0.6232475  1.00000000  0.35720237 -0.01655543 -0.41444964
dist       0.2572324  0.35720237  1.00000000  0.04625029 -0.59498958
cdist      0.1268904 -0.01655543  0.04625029  1.00000000 -0.06448814
customers -0.2644880 -0.41444964 -0.59498958 -0.06448814  1.00000000
```

Note: Take a look at pairs.
Discuss histogram of customers. Compare with

```
mean(customers)
[1] 6.831727
>
> simclus <- rpois(120,6.831727)
> postscript("sim.ps",horizontal=F)
> hist(simclus)
> q()

# Now the Poisson regression
redmod <- glm(customers ~ age+income+ageinc , family=poisson)
fullmod <- update(redmod,. ~ . + dist + cdist)  #  Enter 2 more vars
summary(redmod) ; summary(fullmod)
anova(redmod,fullmod)

Call:
glm(formula = customers ~ age + income + ageinc, family = poisson)

Deviance Residuals:
      Min         1Q     Median         3Q        Max
-2.582719  -0.538975   0.007298   0.689726   1.831947
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.082e+00  2.913e-01   7.145 9.02e-13 ***
age         -2.489e-06  7.157e-06  -0.348    0.728
income       3.025e-04  3.942e-04   0.767    0.443
ageinc      -7.110e-09  8.226e-09  -0.864    0.387
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 97.031  on 109  degrees of freedom
Residual deviance: 80.947  on 106  degrees of freedom
AIC: 496.72

Number of Fisher Scoring iterations: 4

Call:
glm(formula = customers ~ age + income + ageinc + dist + cdist,
    family = poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9166  -0.5580  -0.1817   0.4695   1.6223

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.463e+00  3.086e-01   7.980 1.46e-15 ***
age         -3.906e-06  7.219e-06  -0.541    0.588
income       1.311e-04  4.016e-04   0.326    0.744
ageinc      -1.978e-09  8.353e-09  -0.237    0.813
dist        -1.183e-01  2.721e-02  -4.347 1.38e-05 ***
cdist       -1.071e-03  2.220e-03  -0.482    0.630
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '  1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 97.031  on 109  degrees of freedom
Residual deviance: 61.361  on 104  degrees of freedom
AIC: 481.13
```

```
> anova(redmod,fullmod)
Analysis of Deviance Table

Response: customers


                                     Resid. Df Resid. Dev  Df Deviance
age + income + ageinc                      106     80.947
age + income + ageinc + dist + cdist       104     61.361   2   19.586
```

Elementary Bootstrapping

The basic idea behind the bootstrap is this. For a large sample, the sample distribution is a good approximation of the population distribution. Therefore, we can approximate the sampling distribution of any statistic pretty well by sampling repeatedly with replacement from the sample data, and computing the statistic of interest each time.

```
\/dos/brunner/applied > cat boot1.s
#  Boot1.s:  First bootstrap demo
typodat <- scan("typo.dat",list(id=0,group=0,typos=0))
group <- typodat$group ; typos <- typodat$typos
sum1 <- sum(typos[group==0]) ; n1 <- length(typos[group==0])
sum2 <- sum(typos[group==1]) ; n2 <- length(typos[group==1])
xbar1 <- sum1/n1  ; xbar2 <- sum2/n2 ; xbar <- mean(typos)

# Central Limit Theorem Approximation, just to check
margin <- 1.96 * sqrt(xbar1/n1 + xbar2/n2 )
cltlow <- xbar1-xbar2 - margin
cltup <- xbar1-xbar2 + margin
cat(" 95% clt CI is ",cltlow," to ",cltup,"\n")

mdiff <- numeric(1000)
howlong <- unix.time(
      for(i in 1:1000) mdiff[i] <- rpois(1,sum1)/n1-rpois(1,sum2)/n2 )
cat("User, system & elapsed time = ",howlong[1:3],"\n")
mdiff <- sort(mdiff)
lower <- (mdiff[26]+mdiff[25])/2
upper <- (mdiff[976]+mdiff[975])/2
cat(" 95% parametric bootstrap CI is ",lower," to ",upper,"\n")

# Now a non-parametric bootstrap
typo1 <- typos[group==0] ; typo2 <- typos[group==1]
for(i in 1:1000)
   {
   mdiff[i] <-
      mean(sample(typo1,replace=TRUE))-mean(sample(typo2,replace=TRUE))
   }
mdiff <- sort(mdiff)
lower <- (mdiff[26]+mdiff[25])/2
upper <- (mdiff[976]+mdiff[975])/2
cat(" 95% non-parametric bootstrap CI is ",lower," to ",upper,"\n")

#  Now significance tests

#  First H0: identical distributions
for(i in 1:1000)
   {
   mdiff[i] <-
      mean(sample(typos,size=n1,replace=T)) -
      mean(sample(typos,size=n2,replace=T))
   }
p1 <- length(mdiff[mdiff>abs(xbar1-xbar2)])/1000
cat(" P-value for nonpar = dist bootstrap test is ",p1,"\n")



# Now mean shift H0
shift1 <- typo1-xbar1+xbar ; shift2 <- typo2-xbar2+xbar
for(i in 1:1000)
   {
   mdiff[i] <- mean(sample(shift1,replace=T)) -
mean(sample(shift2,replace=T))
   }
p2 <- length(mdiff[mdiff>abs(xbar1-xbar2)])/1000
cat(" P-value for nonpar shifted mean bootstrap test is ",p2,"\n")

/dos/brunner/applied > rm .Data/*
/dos/brunner/applied > S Sd
Warning: Cannot open audit file
> source("boot1.s")
 95% clt CI is  -0.314813931009298  to  0.793085239532705
 User, system & elapsed time =  2.47000004351139 0.269999980926514 3
 95% parametric bootstrap CI is  -0.380232092837135  to  0.797198879551821
 95% non-parametric bootstrap CI is  -0.308323329331733  to  0.816926770708283
 P-value for nonpar = dist bootstrap test is  0.181
 P-value for nonpar shifted mean bootstrap test is  0.176
```