University of Illinois
Spring 2013

Hotelling Tubes
Economics 508

Department of Economics
Professor Roger Koenker

These notes are based on a "take-home" final exam question I wrote for my 574 course in 2012. Further background is available in the my "Some Notes on Hotelling Tubes" available at `http://bit.ly/IAuBZa` where I've tried to sketch an elementary exposition of the 1939 Hotelling approach. The question was structured as follows: consider a simple model with one Box-Cox transformed covariate,

$$y_i = \beta \frac{x_i^\lambda - 1}{\lambda} + u_i \quad i = 1, \cdots, n,$$

with $\{u_i\}$ iid $\mathcal{N}(0, \sigma^2)$. Suppose we have reason to restrict the Box Cox parameter $\lambda \in [-1, 1]$. We would like to test the hypothesis that β is zero, versus the general alternative that it is non-zero. The Hotelling formulation is based on the likelihood ratio statistic, assuming known σ,

$$T_n = \inf_{\lambda \in [-1,1]} \left\{ \sum_{i=1}^n (y_i - \hat{\beta}(\lambda)(x_i^\lambda - 1)/\lambda)^2 / \sum_{i=1}^n y_i^2 \right\}.$$

1. Explain the form of T_n, and explain how you would reduce the testing problem to this T_n even if you had additional linear covariates in the model for the conditional mean. (Note that $\hat{\beta}(\lambda)$ in the above expression is just the usual least squares estimate, which obviously depends on λ.)
2. Following the recipe in the "Notes" implement the test in R and conduct a small Monte Carlo experiment to validate its performance. (For the sake of comparison, use $m = 500$ for the discretization of the grid for $\|\gamma\|$.)

The basic problem here is that under the null the Box-Cox parameter, λ, isn't identified so we need some sort of special trick to evaluate the distribution of the likelihood ratio statistic. Provided we know σ^2, in our simple model the usual twice log likelihood ratio statistic would be

$$LR = \ell_n(\hat{\beta}(\lambda)) - \ell_n(0)$$

but here we find it convenient to divide through by $\ell_n(0)$ and ignore the superfluous -1. It is worth stressing at this point that any monotone transformation of the LR statistic like the one we have just made has the same rejection region as the original one so we are free to choose a transformation that makes computing the null distribution convenient. Hotelling's strategy is appealing in this way. When there are other covariates we can simply follow the usual Gauss-Frisch-Waugh approach and project the response and the Box-Cox'd covariate onto the space orthogonal to these covariates and proceed as before.

Following the recipe in my "Notes" I ran a small simulation experiment with $\lambda \in \{-0.5, 0, 0.5\}$ and $n \in \{20, 50, 100, 500, 1000\}$ and local alternatives $\beta_n = \beta_0/\sqrt{n}$. Results of the experiment can be seen in the table below. (R code is provided in the appendices.) It will be seen that the nominal 0.05 level of the test based on the Hotelling critical values is quite well maintained, and power is also quite respectible, somewhat better than for $\lambda = 0$. I used standard Gaussian errors and standard lognormal x_i's, but it might be interesting to know how results would differ with other choices.

Table 1: Rejection frequencies for the Hotelling likelihood ratio test for a simple Box-Cox example. Tests are nominal level $\alpha = 0.05$. Local alternatives are employed of the form: $\beta_n = \beta_0/\sqrt{n}$.

	$\beta_0 = \mathbf{0}$			$\beta_0 = \mathbf{1}$			$\beta_0 = \mathbf{2}$		
	λ = -0.5	$\lambda = 0$	$\lambda = 0.5$	λ = -0.5	$\lambda = 0$	$\lambda = 0.5$	λ = -0.5	$\lambda = 0$	$\lambda = 0.5$
n = 20	0.056	0.058	0.049	0.313	0.193	0.182	0.781	0.459	0.380
n = 50	0.049	0.051	0.057	0.275	0.225	0.342	0.639	0.577	0.782
n = 100	0.063	0.048	0.056	0.350	0.261	0.281	0.840	0.637	0.704
n = 500	0.048	0.052	0.055	0.298	0.243	0.288	0.747	0.612	0.735
n = 1000	0.063	0.046	0.047	0.299	0.218	0.250	0.724	0.549	0.667

A Simulation Code

```
# Simulation exercise for Q2 of 2012 574 exam

# Warning: the following function shouldn't be evaluated at lambda == 0
# This is safe for the simulation below with even m.
lam <- function(lambda,x) (x^lambda - 1)/lambda
enorm <- function(x) sqrt(sum(x^2))
Gam <- function(lambda,x){
        Gam <- outer(lambda,x,lam)
        Gam/sqrt(apply(Gam^2,1,sum))
        }
gdotnorm <- function(x, m = 500){
        lambda <- seq(-1,1,length=m)
        G <- Gam(lambda,x)
        dG <- G[-1,] - G[-m,]
        sum(sqrt(apply(dG^2,1,sum)))
        }
phot <- function(w,x,m = 500, method = "JJ") {
# Two equivalent ways to get probability for "caps"
# The default seems to be the standard version as in my "Notes"
# and the Johansen and Johnstone paper cited there.  The
# Student t version is from Knowles and Siegmund (ISR, 1989)
        n <- length(x)
        kappa <- gdotnorm(x, m = m)
        if(method == "KS")
        return(kappa * ((1 - w^2)^((n-2)/2))/(2*pi) +
                (1 - pt(w * sqrt((n-1)/(1-w^2)), n-1)))
        else
        return(kappa * ((1 - w^2)^((n-2)/2))/(2*pi) +
                0.5 * (1 - pbeta(w^2, 0.5, (n-1)/2)))
        }
critval <- function(x, alpha = 0.05){
# Hotelling tube critical value for simple Box-Cox test
   tube <- function(w, x, alpha) phot(w, x) -  alpha
   uniroot(tube,c(0.05,0.5),x = x, alpha = alpha)$root
   }
Whot <- function(x,y,m = 500){
        y <- y/enorm(y)
        lambda <- seq(-1,1,length = m)
        G <- Gam(lambda,x)
```

```
        #plot(lambda, G %*%y, type = "l")
        max(G %*% y)
        }
BoxCox <- function(x,lambda) {
        if(lambda == 0) return(log(x))
        else (x^lambda - 1)/lambda
        }

# The next two lines enable the simulation to run on multiple cores
require(doMC)
registerDoMC(8)
date()
sessionInfo()
set.seed(1939)
R <- 1000
lambdas <- c(-0.5,0,0.5)
cvs <- ns <- c(20, 50, 100, 500, 1000)
betas <- c(0,1,2)
xs <- list()
for(i in 1:length(ns)) xs[[i]] <- exp(rnorm(ns[i])) # one x vector per n
for(i in 1:length(ns)) cvs[i] <- critval(xs[[i]])
A <- array(0,c(length(ns),length(betas),R))
W <- rep(0,R)
ptime <- system.time({
   AA <- foreach(i = 1:length(lambdas)) %dopar% {
      for(j in 1:length(ns)){
         x <- xs[[j]]
         n <- ns[j]
         for(k in 1:length(betas)){
            for(r in 1:R){
               y <-  betas[k]/sqrt(n) * BoxCox(x,lambdas[i]) + rnorm(n)
               A[j,k,r] <- Whot(x,y)
               }
            }
         }
      A
   }
})
```

B Analysis of Simulation Code

```
# Analysis of the q2 simulation
require(Hmisc)
load("q2.Rda")
A <- array(unlist(AA),c(length(ns),length(betas),R,length(lambdas)))
tab <- apply(A > cvs, c(1,2,4),mean)
tab <- aperm(tab,c(1,3,2))
tab <- matrix(tab,length(ns))
cgrp <- paste("$\\beta_0 = $", 0:2)
rnames <- paste("n = ", ns)
cnames <- rep(paste("$\\lambda = $", lambdas),3)
dimnames(tab) <- list(rnames,cnames)
cap <- paste("Rejection frequencies for the Hotelling likelihood ratio test
for a simple Box-Cox example. Tests are nominal level $\\alpha = 0.05$.
Local alternatives are employed of the form: $\\beta_n = \\beta_0 / \\sqrt{n}.$")
latex(tab, file = "q2.tex", rowlabel = "", cgroup = cgrp,
        where = "!htbp", caption = cap)
```