Econ 21410 - Problem Set V

Paper Replication*

May 20, 2014

This homework should be done in LaTeX The homework will be graded on correctness, but will also heavily weight clarity and professionalism. Its better to not do some of the harder parts than to turn in an incomprehensible document. Your R script as well as a log-file should be submitted. Alternatively, use knitr to print your code inline in your latex document.

SUBMISSION: The homework must be emailed to Oliver and myself by 2p.m. Monday, May 12th. The email must include a pdf with the filename lastname_pset6.pdf and R code called lastname_pset6_code.R where "lastname" should be replaced with your last name. The subject of your email should be [ECON 21410: pset6 submission]

If you are struggling, please use the github page to ask for help!* Remember that asking and answering questions on our github page, coming to office hours to ask questions, and contributing to the class wiki are all worth participation credit, which is 10% of your grade in this class.

In this homework we will be replicating a classic paper from Industrial Organization: Bresnahan and Reiss (1991) "Entry and Competition in Concentrated Markets". It can be downloaded here: http://pages.stern.nyu.edu/ acollard/bresnahan-reiss.pdf

The data from the paper can be downloaded from my site or github.

Replicating Bresnahan and Reiss 1991

Using the provided data, replicate Table 4, Table 5-A, and Figure 4 from the paper but only for tire dealers. Describe how you performed the replication, what your calculations were, how they were implemented, the optimizer you used, etc. Your write-up should be written as if you were performing this replication as part of your research, or part of your Job. Include clear details of what you did, but do not be overly verbose.

To calculate your standard errors, use the "hessian()" command in the "numDeriv" library. This command will give a reasonably accurate numerical approximation to the matrix of second derivatives. Recall that your maximum likelihood standard errors will be the the square-root of the diagonal terms of the inverted Hessian evaluated at your estimated parameter values.

Some hints and suggestions:

*Please email johneric@uchicago.edu and obrowne@uchicago.edu if you have questions.

1. Solve the problem using maximum likelihood. Make a function that takes data and parameter values as inputs and returns the negative log-likelihood.

2. To test your log-likelihood function, try using the parameter values reported in their paper. You should get roughly the same log-likelihood (differences larger than 1 may indicate a problem in your code).

3. In the paper, the α and γ parameters are constrained to be greater than or equal to 0 (except γ_L). These constraints are "Bound Constraints" and should not be too difficult to impose (but will force you to use a particular optimizer in optim()).

4. This problem is solvable with optim(), but you will need to give it smart starting values. You can make sure it works by starting with the parameters from the paper, but then you should find alternative starting values that may be what you would have used if writing this paper yourself and did not know the true solutions.

5. Your function will most likely produce some values of "-Inf". These can cause problems and produce NaNs in later steps. You may wish to replace -Inf values with large negative numbers like -1000000000000000000 to avoid this problem.

6. It should be possible to write a relatively short log-likelihood function. Without comments or trying to make my code particularly compact, mine in 19 lines long and references two other functions which are both 9 lines long (without comments).

```r
# Section 1: setup ===================

V <- function(br, n, alpha = rep(1, 5), bet = rep(1,
    4)) {
    profit <- alpha[1] + bet[1] * br$ELD + bet[2] *
        br$PINC + bet[3] * br$LNHDD + bet[4] * br$FFRAC
    if (n >= 2) {
        for (i in 2:n) profit <- profit - alpha[i]
    }
    return(profit)
}

F <- function(br, n, gam = rep(1, 6)) {
    F <- gam[1] + gam[6] * br$LANDV
    if (n >= 2) {
        for (i in 2:n) F <- F + gam[i]
    }
    return(F)
}

# theta values from their paper
theta.true <- c(-0.53, 2.25, 0.34, 0.23, -0.49, -0.03,
    0.004, -0.02, 0.86, 0.03, 0.15, 0, 0.08, 0.53,
    0.76, 0.46, 0.6, 0.12, -0.74)

# Liklihood function
lLik <- function(theta = rep(1, 19), br = br) {
    lam <- theta[1:4]
```

```
    bet <- theta[5:8]
    alpha <- theta[9:13]
    gam <- theta[14:19]

    S <- br$TPOP + lam[1] * br$OPOP + lam[2] * br$NGRW +
        lam[3] * br$PGRW + lam[4] * br$OCTY
    P <- list()
    P[[1]] <- log(1 - pnorm(S * V(br, 1, alpha = alpha,
        bet = bet) - F(br, 1, gam = gam)))
    P[[2]] <- log(pnorm(S * V(br, 1, alpha = alpha,
        bet = bet) - F(br, 1, gam = gam)) - pnorm(S *
        V(br, 2, alpha = alpha, bet = bet) - F(br,
        2, gam = gam)))
    P[[3]] <- log(pnorm(S * V(br, 2, alpha = alpha,
        bet = bet) - F(br, 2, gam = gam)) - pnorm(S *
        V(br, 3, alpha = alpha, bet = bet) - F(br,
        3, gam = gam)))
    P[[4]] <- log(pnorm(S * V(br, 3, alpha = alpha,
        bet = bet) - F(br, 3, gam = gam)) - pnorm(S *
        V(br, 4, alpha = alpha, bet = bet) - F(br,
        4, gam = gam)))
    P[[5]] <- log(pnorm(S * V(br, 4, alpha = alpha,
        bet = bet) - F(br, 4, gam = gam)) - pnorm(S *
        V(br, 5, alpha = alpha, bet = bet) - F(br,
        5, gam = gam)))
    P[[6]] <- log(pnorm(S * V(br, 5, alpha = alpha,
        bet = bet) - F(br, 5, gam = gam)))
    for (i in 1:6) P[[i]][P[[i]] == -Inf] <- -1e+08

    lLik <- sum(P[[1]][br$TIRE == 0]) + sum(P[[2]][br$TIRE ==
        1]) + sum(P[[3]][br$TIRE == 2]) + sum(P[[4]][br$TIRE ==
        3]) + sum(P[[5]][br$TIRE == 4]) + sum(P[[6]][br$TIRE >=
        5])
    return(-1 * lLik)
}

# ==================
```

```
# Section 2: numerical optimization
# ====================

# What I want them to do
out1 <- optim(par = rep(0.1, 19), lLik, br = br, lower = c(rep(-Inf,
    8), rep(0, 10), -Inf), upper = rep(Inf, 19), method = "L-BFGS-B",
    control = list(maxit = 500))

# ==================
```

One Additional step

The optimizer called by "optim()" is approximating the gradient with finite-differences. The library numDeriv offers an improved way to numerically approximate derivatives that tend to be more accurate. Using the "grad()" function in the numDeriv library, write a new function that returns the numeric gradient.

Use this function in the "gr" option of "optim()" to have optim() use this function rather than the built in finite-differences method. Does this change your solutions? Are they closer to the ones reported in the paper? Use "system.time" to time both methods, report and discuss (1-2 sentences).

```
# Section 3: using numerical gradient
# ====================

# Defining a function that returns the numeric
# gradient.
lLik.grad <- function(theta = rep(1, 19), br = br) {
    grad(x = theta, lLik, br = br)
}

out2 <- optim(par = rep(0.1, 19), lLik, gr = lLik.grad,
    br = br, lower = c(rep(-Inf, 8), rep(0, 10), -Inf),
    upper = rep(Inf, 19), method = "L-BFGS-B")

# Calculating Standard errors using hessian (not a
# great method ofcourse since we are inverting a
# numerical hessian etc etc.)
out.grad <- grad(lLik, x = out2$par, br = br)
out.hessian <- hessian(lLik, x = out2$par, br = br)
out.se <- sqrt(diag(solve(out.hessian)))

cbind(out2$par, out1$par, theta.true, out2$par - theta.true,
    out1$par - theta.true)

results <- cbind(out2$par, out.se)
colnames(results) <- c("estimate", "se")
rownames(results) <- c("lambda1", "lambda2", "lambda3",
    "lambda4", "beta1", "beta2", "beta3", "beta4",
    "alpha1", "alpha2", "alpha3", "alpha4", "alpha5",
    "gamma1", "gamma2", "gamma3", "gamma4", "gamma5",
    "gamma6")
stargazer(results, out = "n6_paperReplication/results.tex")

# ====================
```

```
# Section 4: Calculate thresholds
# ====================

calc.S <- function(br, theta) {
```

```
    lam <- theta[1:4]
    bet <- theta[5:8]
    alpha <- theta[9:13]
    gam <- theta[14:19]
    br.m <- data.frame(t(colMeans(br)))

    S <- rep(0, 5)
    for (i in 1:5) S[i] <- F(br.m, i, gam)/V(br.m,
        i, alpha, bet)

    return(S)
}

S <- calc.S(br, out2$par)
SN.S5ratio <- (S[5] * (1:5))/(S * 5)
qplot(1:5, SN.S5ratio)
ggsave(file = "SNS5ratio.pdf")

# ==================
```

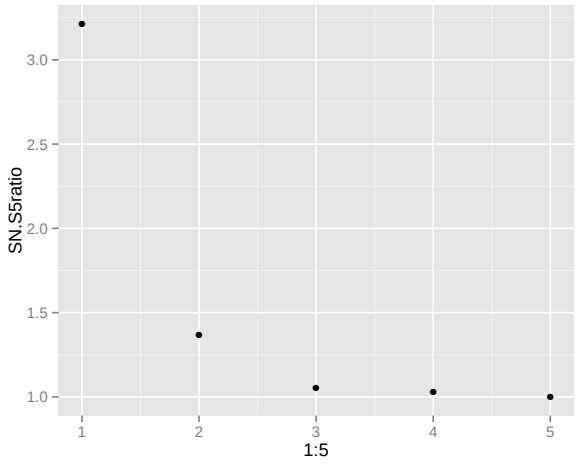
Table 1:

	estimate	se
lambda1	-0.579	0.485
lambda2	0.616	1.546
lambda3	0.623	0.795
lambda4	0.081	0.441
beta1	-0.269	0.644
beta2	-0.014	0.031
beta3	-0.038	0.059
beta4	-0.012	0.078
alpha1	0.969	0.459
alpha2	0.007	0.107
alpha3	0.071	0.090
alpha4	0	0.079
alpha5	0.082	0.050
gamma1	0.452	0.230
gamma2	0.748	0.187
gamma3	0.642	0.218
gamma4	0.591	0.250
gamma5	0.099	0.172
gamma6	-0.836	0.412

Side Projects

- Write a function above that returns the analytic gradient. Use it in the step above and report the timing (up to 2.5 points).

Figure 1: Relative profits



- Program some or all of this assignment in C++ using Rcpp or in Julia (up to 6 points).
- Program some or all of assignment 2 or 3 in C++ using Rcpp or in Julia (up to 6 points).
- Write down 3 research ideas. For each explain the idea in 4-10 sentences. Write 1-4 sentences on why it is interesting, write 1-4 sentences on what data you would use, write 1-4 sentences on what methods it would use. These ideas can be very rough. If you would like feedback on one or all of them, let us know (up to 1 point each, whole assignment can be done twice)
- Find a paper that looks at market entry and write a 1-3 page report on it (up to 3.5 points).
- Find up to 4 articles in the news and generate a related research ideas. Explain the idea in 1-4 sentences and how you might execute the research idea (1-3 sentences) (up to 1 point per idea).
- Read any published academic article by by Gary Becker and write a 2-3 page summary and response (up to 3 points).
- Commonly for grants or applications you will be required to write research proposals. Write a research proposal that follows the guidlines set out by the NSF Graduate Research Fellowship. Under "Graduate Research Statement" here:

  http://www.nsfgrfp.org/how_to_apply/application_materials#proposed. (up to 5 points).