Problem Set 2

Directions: Answer all questions. Clearly label all answers. Show all of your code. Compute standard errors for all parameter estimates. Turn in the following to me via your dropbox (in a folder labeled 'MatlabPS2.2') in Sakai by 11:59 p.m. on Thursday, August 2, 2012:

- m-file(s)
- .sh shell script file that executes your m-file(s)[1]
- a log file (from off the cluster)
- matsub.oXXXXX file
- pdf version of your writeup with its LaTeXsource code

Put the names of all group members at the top of your writeup (each student must turn in his/her own materials).

1. Gradient of Multinomial Logit.

   (a) Derive the gradient of the multinomial logit log likelihood function, i.e. find $\partial\ell/\partial\beta_j$ where

$$\begin{aligned}
\mathscr{L}\left(X_{it};\beta\right) &= \prod_i \prod_j \prod_t P_{ijt}^{d_{ijt}} \\
\ln\left(\mathscr{L}\left(X_{it};\beta\right)\right) &= \ln\left(\prod_i \prod_j \prod_t P_{ijt}^{d_{ijt}}\right) \\
\ell\left(X_{it};\beta\right) &= \sum_i \sum_j \sum_t d_{ijt} \ln\left(P_{ijt}\right) \\
&= \sum_i \sum_j \sum_t d_{ijt} \ln\left(\frac{\exp\left(X_{ijt}\beta_j\right)}{\sum_k \exp\left(X_{ikt}\beta_k\right)}\right).
\end{aligned}$$

   Express this derivative in matrix form. For more help, see p. 63 of Train[2].

[1] Name your job something other than "matsub," and have it send you an email upon completion.

[2] Available for free online at `http://elsa.berkeley.edu/books/choice2.html`. Note that Train expresses his formula in terms of $\partial\ell/\partial\beta$, not $\partial\ell/\partial\beta_j$.

2. Estimation of Multinomial Logit with and without gradient. Using the dataset `nlsy97m.mat`, consider the following model (same as Problem Set 1)

$$\begin{aligned} d_{ijt} &= \beta_{1j} + \beta_{2j} male_i + \beta_{3j} AFQT_i + \beta_{4j} Mhgc_i + \beta_{5j} hgc_{it} + \beta_{6j} exper_{it} + \\ &\quad \beta_{7j} Diploma_{it} + \beta_{8j} AA_{it} + \beta_{9j} BA_{it} + \varepsilon_{ijt} \end{aligned} \tag{1}$$

where $\varepsilon_{ijt} \overset{iid}{\sim}$Type I Extreme Value, and d_{ijt} is an indicator that $choice_{it} = j$ for individual i in time t. $choice_{it}$ now takes on values $1, \ldots, 6$ according to the following fashion:

- school only
- work while attending school
- work part-time (no school)
- work full-time (no school)
- active military duty
- all other activities

(a) Tabulate the choice set.

(b) Estimate β using `fminunc` and random starting values[3]. Normalize β_6 to be zero (i.e. use "other activities" as the reference group). Report $\hat{\beta}$ and its standard errors, along with the log likelihood value at convergence and number of iterations to convergence.

(c) Now estimate β using `fminunc` and random starting values, but this time supply `fminunc` with the gradient you computed in part (a) of question 1.[4] What difference does the user-supplied gradient make in terms of computational time as well as estimate precision?

3. Constrained optimization within `fminunc`. Using the same dataset as in question 2, estimate the following model:

$$\begin{aligned} \ln(wage_{it}) &= \gamma_1 + \gamma_2 male_i + \gamma_3 AFQT_i + \gamma_4 Mhgc_i + \gamma_5 hgc_{it} + \gamma_6 exper_{it} + \\ &\quad \gamma_7 Diploma_{it} + \gamma_8 AA_{it} + \gamma_9 BA_{it} + \gamma_{10} school\&work_{it} + \gamma_{11} workPT_{it} + \varepsilon_{it} \end{aligned}$$

where $\varepsilon_{it} \overset{iid}{\sim} N(0, \sigma^2)$. $school\&work_{it}$ is a dummy for whether or not the individual i chose school and work (choice 2) in time t, and $workPT_{it}$ is a dummy for whether or not individual i chose work part-time (choice 3) in time t.

(a) Estimate $\hat{\gamma}$ using `fminunc` and imposing the following constraints within the optimization[5]:

- $\hat{\sigma} > 0$

[3] As usual, set the seed to 1234.

[4] Remember that you can use the `'DerivativeCheck','on'` option in `fminunc` to compare your analytical derivative with Matlab's numerical approximation.

[5] Hint: The following constraint functions are compatible with these constraints: $\exp(x)$, $-x^2$, $.15 * \tanh(x) + .05$

- $\hat{\gamma}_{10} < 0$
- $-.1 < \hat{\gamma}_2 < .2$

Be sure to report the delta-method-corrected standard errors with your point estimates.

4. Estimation of normal MLE with and without gradient and hessian.[6]

   (a) Estimate the same model as in question 3, but this time supply the gradient of the log likelihood to `fminunc`. Compare your results with and without the user-provided gradient. Report coefficient estimates, log likelihood values, iteration counts, and standard errors of your coefficient estimates. Comment on the computational savings of a user-provided gradient.

   (b) Repeat (a), but this time supply both the gradient and the hessian.[7] For help on what this looks like, consult Hayashi (pp. 49-51) or Greene (pp. 518-519). Compare your results with and without the user-provided hessian. Report coefficient estimates, log likelihood values, iteration counts, and standard errors of your coefficient estimates. Comment on the computational savings of a user-provided hessian.

[6] For this problem, start all optimization at the same starting values (one specific draw from a $U[0,1]$ distribution).

[7] Note: Matlab does not have an analogous option to `'DerivativeCheck'` for the user-supplied hessian. However, there is a user-written script called `hessian` (the full package name is "Adaptive Robust Numerical Differentiation" which you can find on the Matlab file exchange), which will numerically calculate the hessian of an objective function. This is useful as a way to check that your differentiation is correct.