Lecture 7: Numerical Optimization

ResEcon 703: Topics in Advanced Econometrics

Matt Woerman
University of Massachusetts Amherst

Agenda

Last time

- Nonlinear Regression Models
- Maximum Likelihood Estimation

Today

- Numerical Optimization
- Recap of Random Utility and Logit Models

Upcoming

- Reading for next time
  - Train textbook, Chapters 3.7–3.8
  - Bayer et al. (2009)
- Problem sets
  - Problem Set 1 solutions are posted
  - Problem Set 2 will be posted soon, due October 17

Maximum Likelihood Recap

The probability density function (PDF) for a random variable, y, conditioned on a set of parameters, θ, is

$$f(y \mid \theta)$$

The log-likelihood function for θ conditional on observed data is

$$\ln L(\theta \mid y) = \sum_{i=1}^{n} \ln f(y_i \mid \theta)$$

The maximum likelihood estimator (MLE) is the value(s) of θ that maximizes this function

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \ln L(\theta \mid y)$$

Numerical Optimization

Numerical Optimization

Most structural estimation requires maximizing (or minimizing) an objective function

- For MLE, we want to maximize the log-likelihood function

In theory, this is a relatively simple proposition

- Some optimization problems have a closed-form expression
- For only one or two parameters, a grid search may suffice

In practice, finding the correct parameters in an efficient way can be challenging

- Especially when you are optimizing over a vector of many parameters and using a complex objective function
- Numerical optimization algorithms can solve this problem

Numerical Optimization Steps

We want to find the set of K parameters, $\hat{\beta}$, that maximize the objective function, $\ell(\beta)$

1. Begin with some initial parameter values, β_0
2. Check if you can "walk up" to a higher value
3. If so, take a step in the right direction to β_{t+1}
4. Repeat (2) and (3) until you are at the maximum

But which direction should you step and how big of a step should you take from β_t to β_{t+1}?

- If your steps are too small, optimization can take too long
- If your steps are too big, you may never converge to a solution

Gradient and Hessian

The gradient tells us which direction to step

$$g_t = \left(\frac{\partial \ell(\beta)}{\partial \beta} \right)_{\beta_t}$$

- The gradient is a $K \times 1$ vector tells us which direction to move each parameter to increase the objective function

The Hessian tells us how far to step

$$H_t = \left(\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta'} \right)_{\beta_t}$$

- The Hessian is a $K \times K$ matrix that gives us information about the "curvature" of the objective function in all dimensions

Newton-Raphson Method

The Newton-Raphson method is based on the second-order Taylor's approximation of $\ell(\beta_{t+1})$ around $\ell(\beta_t)$

$$\ell(\beta_{t+1}) = \ell(\beta_t) + (\beta_{t+1} - \beta_t)' g_t + \frac{1}{2}(\beta_{t+1} - \beta_t)' H_t (\beta_{t+1} - \beta_t)$$

We step to the value of β_{t+1} that maximizes this approximation

$$\frac{\partial \ell(\beta_{t+1})}{\partial \beta_{t+1}} = 0 \quad \Rightarrow \quad \beta_{t+1} = \beta_t + \lambda(-H_t)^{-1} g_t$$

This method steps to what would be the maximizing vector of parameters if the objective function was quadratic

- If the objective function is not close to quadratic, steps can be too small or too large
  - You can iteratively scale the step size to be larger or smaller using λ
- Steps can go in the wrong direction if the objective function is not globally concave

Score

When we are maximizing a log-likelihood function, we can speed up optimization by exploiting the fact that we are maximizing a sum of individual-specific terms

To do this, we calculate the score for each individual

$$s_n(\beta_t) = \left(\frac{\partial \ln L_n(\beta)}{\partial \beta}\right)_{\beta_t}$$

If we think of maximizing the average log-likelihood

$$LL(\beta) = \frac{\sum_{n=1}^{N} \ln L_n(\beta)}{N}$$

then the gradient is equal to the average score

$$g_t = \frac{\sum_{n=1}^{N} s_n(\beta_t)}{N}$$

BHHH (Berndt-Hall-Hall-Hausman) Method

The BHHH method uses the the average outer product of scores, which is related to the variance and covariance of scores, to calculate step size

$$B_t = \frac{\sum_{n=1}^{N} s_n(\beta_t) s_n(\beta_t)'}{N}$$

The BHHH method uses this average outer product in place of the Hessian

$$\beta_{t+1} = \beta_t + \lambda B_t^{-1} g_t$$

Advantages of BHHH over NR

- B_t is faster to calculate than H_t
- B_t is always positive definite, so no concavity problems

Other Methods

- BHHH-2
- Steepest ascent
- DFP (Davidson-Fletcher-Powell)
- BFGS (Broyden-Fletcher-Goldfarb-Shanno)
- Nelder-Mead
- Conjugate gradients
- Limited-memory BFGS
- Simulated annealing

Convergence Criterion

When do we stop taking steps?

- In theory, when the gradient vector equals zero
- In practice, you will never hit the precise vector of parameters (down to the 15th decimial point) that yields a gradient of zero
- So we stop taking steps when we get "close enough"

How do we know when we are "close enough?"

- Calculate a statistic, m_t, to evaluate convergence

$$m_t = g_t'(-H_t^{-1})g_t$$

- Stop when this statistic gets sufficiently small

$$m_t < \breve{m} = 0.0001$$

Global or Local Maximum

Global maximum

- The largest value of the objective function over all possible sets of parameter values
- This is the maximum you want to converge to
- When the objective function is globally concave (as in the logit model with linear utility), you will always hit the global maximum

Local maximum

- The largest value of the objective function within a range of parameter values, but not the global maximum
- Optimization algorithms will sometimes converge to a local maximum instead of the global maximum
- More complex objective functions have local maxima

Try different starting values to ensure you have converged to the global maximum, not a local maximum

Recap of Random Utility and Logit Models

Random Utility Model

Discrete choice from the perspective of the decision maker

- Decision maker, n, faces a choice among J alternatives
- Alternative j provides utility U_{nj}
- The decision maker chooses the alternative with the greatest utility

Discrete choice from the perspective of the econometrician

- We do not observe utility, but we do observe
  - The choice
  - Data about the alternatives
  - Data about the decision makers
- We model the utility of alternative j as $U_{nj} = V_{nj} + \varepsilon_{nj}$
  - V_{nj} is the component of utility from observed factors
  - ε_{nj} is the component of utility we do not observe
- We do not observe ε_{nj}, so we cannot model the choice with certainty, but we can treat ε_{nj} as random and form probabilities

Representative Utility Example

Let's write down the representative utility for each of the four commute options from problem 3(a) on problem set 1

- Intercept term for all but one alternative
- Cost divided by income with a common coefficient
- Time with alternative-specific coefficients

$$
\begin{aligned}
V_{nk} &= \beta_1 \frac{C_{nk}}{I_n} + \beta_2 T_{nk} & \qquad U_{nk} &= V_{nk} + \varepsilon_{nk} \\
V_{nb} &= \alpha_b + \beta_1 \frac{C_{nb}}{I_n} + \beta_3 T_{nb} & \qquad U_{nb} &= V_{nb} + \varepsilon_{nb} \\
V_{nc} &= \alpha_c + \beta_1 \frac{C_{nc}}{I_n} + \beta_4 T_{nc} & \qquad U_{nc} &= V_{nc} + \varepsilon_{nc} \\
V_{nw} &= \alpha_w + \beta_1 \frac{C_{nw}}{I_n} + \beta_5 T_{nw} & \qquad U_{nw} &= V_{nw} + \varepsilon_{nw}
\end{aligned}
$$

Choice Probabilities

The decision maker chooses the alternative that gives the greatest utility

- Decision maker n chooses alternative i if and only if $U_{ni} > U_{nj} \; \forall j \neq i$
- But we treat ε_{nj} as random, so we model the probability that a given alternative is chosen

The probability that decision maker n chooses alternative i is

$$\begin{aligned} P_{ni} &= \Pr(U_{ni} > U_{nj} \; \forall j \neq i) \\ &= \int_{\varepsilon} I(\varepsilon_{nj} - \varepsilon_{ni} < V_{ni} - V_{nj} \; \forall j \neq i) f(\varepsilon_n) d\varepsilon_n \end{aligned}$$

We make assumptions about the joint density of unobserved components, $f(\varepsilon_n)$, to make this integral more tractable

- Different assumptions about $f(\varepsilon_n)$ yield different discrete choice models

Choice Probabilities Example

The probability that decision maker n chooses to drive a car to campus is

$$\begin{aligned} P_{nc} &= \Pr(U_{nc} > U_{nk}, U_{nc} > U_{nb}, U_{nc} > U_{nw}) \\ &= \int_{\varepsilon} I(\varepsilon_{nk} - \varepsilon_{nc} < V_{nc} - V_{nk}, \\ &\qquad \varepsilon_{nb} - \varepsilon_{nc} < V_{nc} - V_{nb}, \\ &\qquad \varepsilon_{nw} - \varepsilon_{nc} < V_{nc} - V_{nw}) f(\varepsilon_n) d\varepsilon_n \end{aligned}$$

Choice probabilities for the other three alternatives are defined similarly

Logit Model

Logit assumption about the joint distribution of unobserved utility

$$\varepsilon_{nj} \sim \text{i.i.d. type I extreme value (Gumbel) with } Var(\varepsilon_{nj}) = \frac{\pi^2}{6}$$

Logit choice probabilities have a closed-form expression

$$P_{ni} = \frac{e^{V_{ni}}}{\sum_j e^{V_{nj}}}$$

Logit Choice Probabilities Example

The logit choice probability for each commute mode is

$$P_{nk} = \frac{e^{V_{nk}}}{\sum_j e^{V_{nj}}} = \frac{e^{\beta_1 \frac{C_{nk}}{I_n} + \beta_2 T_{nk}}}{e^{V_{nk}} + e^{V_{nb}} + e^{V_{nc}} + e^{V_{nw}}}$$

$$P_{nb} = \frac{e^{V_{nb}}}{\sum_j e^{V_{nj}}} = \frac{e^{\alpha_b + \beta_1 \frac{C_{nb}}{I_n} + \beta_3 T_{nb}}}{e^{V_{nk}} + e^{V_{nb}} + e^{V_{nc}} + e^{V_{nw}}}$$

$$P_{nc} = \frac{e^{V_{nc}}}{\sum_j e^{V_{nj}}} = \frac{e^{\alpha_c + \beta_1 \frac{C_{nc}}{I_n} + \beta_4 T_{nc}}}{e^{V_{nk}} + e^{V_{nb}} + e^{V_{nc}} + e^{V_{nw}}}$$

$$P_{nw} = \frac{e^{V_{nw}}}{\sum_j e^{V_{nj}}} = \frac{e^{\alpha_w + \beta_1 \frac{C_{nw}}{I_n} + \beta_5 T_{nw}}}{e^{V_{nk}} + e^{V_{nb}} + e^{V_{nc}} + e^{V_{nw}}}$$

Estimating the Logit Model

We estimate the logit model by finding the set of parameters that best fit our data

- Parameters: α_b, α_c, α_w, β_1, β_2, β_3, β_4, and β_5
- What set of parameters makes it most likely to observe to the data that we do observe?

Two options for estimation

1. Code up the maximum likelihood estimator...next time!
2. Let `mlogit()` find the MLE for us

`mlogit()` in R

Two steps to estimating a multinomial logit model in R with `mlogit()`

1. Use `mlogit.data()` to organize your dataset in a way that `mlogit()` will understand
    - See previous R code examples for how to do this
    - Not trivial, but once you get it figured out, not too hard
2. Use `mlogit()` to estimate model parameters
    - Tricky part is specifying the model formula correctly

```
mlogit(formula = y ~ a | b | c)
```

- a: Variables with common coefficients
- b: Individual-specific variables with alternative-specific coefficients
- c: Alternative-specific variables with alternative-specific coefficients

`mlogit` (and other packages) have vignettes that can be very helpful

- `cran.r-project.org/web/packages/mlogit/index.html`

`mlogit()` Example in R

We want to specify a model where the representative utility is

$$V_{nj} = \alpha_j + \beta_1 \frac{C_{nj}}{I_n} + \beta_j T_{nj}$$

`mlogit(formula = y ~ a | b | c)`

- `a`: Variables with common coefficients
- `b`: Individual-specific variables with alternative-specific coefficients
- `c`: Alternative-specific variables with alternative-specific coefficients

```r
## Model choice as multinomial logit with common cost/income coefficient,
## alternative intercepts, and alternative-specific time coefficients
model_mlogit <- data_mlogit %>%
  mlogit(mode ~ I(cost / income) | 1 | time, data = .)
```

mlogit() Results in R

```
## Summarize model results
model_mlogit %>%
  summary()
##
## Call:
## mlogit(formula = mode ~ I(cost/income) | 1 | time, data = .,
##     method = "nr")
##
## Frequencies of alternatives:
##  bike   bus   car  walk
## 0.103 0.290 0.465 0.142
##
## nr method
## 11 iterations, 0h:0m:0s
## g'(-H)^-1g = 7.46E-06
## successive function values within tolerance limits
##
## Coefficients :
##                   Estimate Std. Error  z-value  Pr(>|z|)
## bus:(intercept)    3.392381   0.279235  12.1488 < 2.2e-16 ***
## car:(intercept)    6.650311   0.464905  14.3047 < 2.2e-16 ***
## walk:(intercept)   3.711605   0.338860  10.9532 < 2.2e-16 ***
## I(cost/income)   -76.745551   5.590969 -13.7267 < 2.2e-16 ***
## bike:time         -0.362973   0.026469 -13.7129 < 2.2e-16 ***
## bus:time          -0.232457   0.023900  -9.7263 < 2.2e-16 ***
## car:time          -0.475974   0.045422 -10.4790 < 2.2e-16 ***
## walk:time         -0.401968   0.034372 -11.6946 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -760.7
## McFadden R^2:  0.3797
## Likelihood ratio test : chisq = 931.28 (p.value = < 2.22e-16)
```

Announcements

Reading for next time

- Train textbook, Chapters 3.7–3.8
- Bayer et al. (2009)

Upcoming

- Problem Set 2 will be posted soon, due October 17