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

Upcoming

- Reading for next time
  - Train textbook, Chapters 3.7–3.8
  - Bayer et al. (2009)
- Problem sets
  - Problem Set 1 was due at 10 am today
  - 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

Announcements

Reading for next time

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

Office hours

- Reminder: Tuesdays at 2:00-3:00 in 218 Stockbridge

Upcoming

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