Maximum Likelihood 1

Jacob M. Montgomery

2017

Maximum likelihood estimation

Overview

- Last time we talked about
  - What is an MLE estimate?
  - What are the properties of these estimators?
- Today we are are going to talk about some numerical methods that are used to estimate maximum likelihood models

Problems with maximum likelihood models

Problem 1: What if the likelihod function is flat around the mode?

- This is an indeterminant (and fortunately rare) occurance.
- We say that the model is "non-identified" because the likelihood function cannot discriminate between alternative MLE values
- usually this comes from a model specification that has ambiguities.

Problem 2: What if the the likelihood function has more than one mode?

- Then i is difficult to choose one, even if we had perfect knowledge about the shape of the function.
- This model is idenfied provided that there is some criteria for picking a mode.
- Usually comes from complex model specifications.

Algorithms for maximizing log-likelihood functions

- Computers guess at critical points using techniques like the **Newton-Raphson** (NR) algorithm
- They use the NR algorithm to "climb the hill": to find the peak of the log-likelihood function.

Assume that we are maximizing the negative log-likelihood of a model with one parameter.

- First we choose an arbitrary starting point θ_0.

-The next θ is in the chain is always given by:

$$\theta_{n+1} = \theta_n - \frac{\mathcal{L}'(\theta_n)}{\mathcal{L}''(\theta_n)}$$

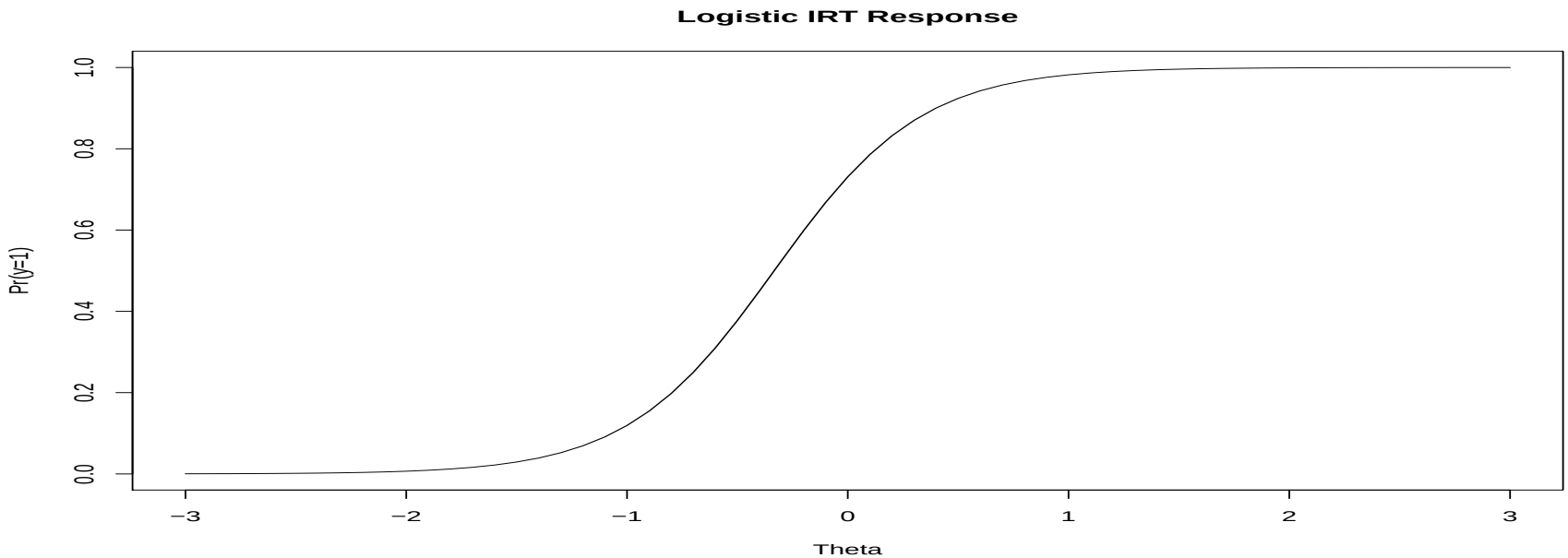
- Continue until each θ is basically equivalent to the previous θ in the sequence. The last θ_n in the sequence is the critical point.

Example

As an example we are going to consider the problem of estimate the "ability" parameter for individuals in a logistic item response model:

$$P_{ij} = 1 - Q_i j \equiv Pr(y_{ij} = 1|\theta_j) = \frac{\exp(a_i + b_i\theta_j)}{1 + \exp(a_i + b_i\theta_j)}$$

```r
a=1; b=3
x<-seq(-3, 3, by=.1)
y<-exp(a+b*x)/(1+exp(a+b*x))
plot(x, y, main="Logistic IRT Response", type="l", xlab="Theta", ylab="Pr(y=1)")
```



Likelihood function:

$$L(\theta_j|\mathbf{y}_j) = \prod_{i=1}^{J} \Big(p_i(\theta_j)^{y_{ij}} q_i(\theta_j)^{(1-y_{ij})} \Big)$$

$$= \exp\Big[\sum_{i=1}^{J} \Big(y_{ij} \log(p_i(\theta_j)) + (1 - y_{ij}) \log(q_i(\theta_j)) \Big) \Big]$$

First and second derivatives of the likelihood function

Take the derivative in terms of θ

$$L_\theta = \sum_{i=1}^{n} b_i(y_{ij} - P_{ij}) \tag{1}$$

$$\lambda_{\theta\theta} = -\sum_{i=1}^{n} b_i^2 \Big(P_{ij}\Big)^2 \frac{Q_{ij}}{P_{ij}} \tag{2}$$

Class activity: Visualize

normalsize

```
library(catSurv)
data(ltm_cat)
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))
# Your job is to plot the likelihood and
# the derivatives of the first and second derivatives
likelihood(ltm_cat, theta = 1)
d1LL(ltm_cat, theta = 1, use_prior = FALSE)
d2LL(ltm_cat, theta = 1, use_prior = FALSE)
```

Class activity: Write the algorithm

$$\theta_{n+1} = \theta_n - \frac{\mathcal{L}'(\theta_n)}{\mathcal{L}''(\theta_n)}$$

1. Write a function with your group the execute the algorithm
2. Add points to the likelihood function in your previous plot showing each iteraction.

Problems

NR-algorithm and other gradient ("hill-climber") methods may have the following problems:

1. Does not distinguish betwee a max and min
2. Does not distinguish between local and global
3. Get stucks, or moves slowly when the data is flat (saddle point).

These problems are called convergence problems since the hill-climber did not converge to the critical point.

Gradient decent

A closely related method is gradient decent, follows the same basic idea but just follows down the negative gradient.

$$\theta_{n+1} = \theta_n - \gamma \nabla \mathcal{L}(\theta)$$

where γ is some small number. When the change from one iteration to another is "small", then we can stop

Example:

Using the same example stated above implement a gradient decent algorithm. How many iterations does it take to converge? Is that more or less than the NR algorithm above?

Iterated generalized method of moments

- Last lecture we established that method of moments estimators are asymptotically consistent.

Iterated generalized method of moments

- Last lecture we established that method of moments estimators are asymptotically consistent.
- But they are often very inefficient.

Iterated generalized method of moments

- Last lecture we established that method of moments estimators are asymptotically consistent.
- But they are often very inefficient.
- However, it is possible to add a weights matrix to improve the efficiency of the estimator.
- So we are going to try and add a weights matrix W to alter the variance/covariance matrix.
- Relaxes the iid assumption a bit
- Very popular among economists and their cousins in political science and especially in time series data.

Example

$$y_i = \mathbf{x}_i'\beta + \epsilon_i$$

Note that $E(\epsilon_i \mathbf{x}_i) = 0$, which implies that

$$E[(y_i - x_i'\beta)\mathbf{x}] = 0$$

GMM estimators try to find the value of β where this equation is exactly zero.

Discussion of GMM

- In many cases we do not get one simple equation, but rather a system of equations that cannot be solved directly.
- Further, some equations will contain the first moment and another could contain the fourth. When trying to minimize the "orthogonality conditions" of the model, we want to give lower moments more weight.

1. We take the first derivatives of the moment conditions, which gives us a vector $\mathbf{g}$.
2. We want to minimize $\mathbf{g}'Wg$ with some W that weights the equations.
3. The optimal weights matrix turns out to be $E(gg')^{-1}$, which we can approximate as

$$S = \left(\frac{1}{N} \sum_{i=1}^{N} g_i g_i' \right)^{-1}$$

4. A two-step procedure is to start off with $W = I$ to estimate θ and then using this formula. This is referred to as a two-step or feasible GMM.
5. Another approach is to do this iteratively until the value of θ stops changing (the difference between each iteration falls below some threshold.)