Lecture 6: Estimation and Optimizers

Su, Danyang

Estimation:

- Generally non-linear (or you should use Stata)
- Time consuming
- Need to deal with a lot of technical issues

GMM:

- Essentially it's about matching moments.
- $\min_\delta ||\hat{M} - M(\delta)||$, normally l_2 or l_∞
- Most of the time you won't be bothered by over-identification.

GMM:

- Average winning bid for μ and variance of bid for σ
- Maximal number of auctions for m
- $\frac{E(X-n)}{E(X-w)} = \gamma$
- Minimal payment for r
- If we don't have bidder id, can we still identify n?

Estimation with Matlab:

- Matlab provides many powerful estimation toolboxes
- (add examples, local vs global)

Before We Jump In:

- Think hard to simplify the problem
- Simplification may depends heavily on problem at hand.
- For complicated method totally worth it
  - Can we simply play with moments directly?
  - No. Because of r.
- Value iteration, contraction or matrix inversion?

Technical Issues:

- What is the maximal grid range?
- How to deal with values outside the box?
- Curse of dimension
- Keep same group of sample.
- Need burn-in period
- Global vs. local

Matlab Optimizers

- Local Optimizers:
  - Gradient-based optimizers: fminsearch
  - Non-gradient-based optimizers: fminunc/fmincon
- Global optimizers:
  - Genetic Algorithm
  - Multistart
- Matlab offers many configurations

Example:

- Find $\arg\max_x \sum -(x-2)^2$ where x is a 4-by-1 vector

```
obj = @(x) sum((x-2).^2)
options = optimset('Display','iter','TolX',1e-8);
[x fval] = fminsearch(obj,randn(4,1),options);
```

- optimize = minimize, so remember to change sign if looking for max

Function Handles:

- f = @(x,y) x^2+y^2; f(2,3);
- f 1 = @(x,y) [f(x,y);f(x,y)+1];
- Optimizer only takes one inputs
- obj = @(x) my_func(x,parameters)

NLopt:

- Much more powerful than Matlab build-in optimizer toolbox
- try $\min_x (x-20)^2$ when x is a 1-by-40 vector.
- Runs on C++, so it's faster.
- Installation is subtle (for non-CS people)

Installation of NLopt on Windows:

- Install skd7/visual studio
- run cmd mode in skd7/visual studio
- in the folder where .dll is stored, type lib /def:libnlopt-0.def
- in Matlab, type: mex -setup c++
- mex nlopt_optimize.c path\nlopt\libnlopt-0.lib -Ipath\nlopt
- copy the dll file to the folder where mexw64 is stored

Installation of NLopt on Mac:

- Install Xcode
- Follow the instruction of the patch page
- Rename nlopt_optimize.c as nlopt_optimize-mex.c in source code (if necessary)
- in terminal run: ./configure –enable-shared MATLAB= /Applications/MATLAB_R2014b.app MEX=/Applications/MATLAB_R2014b.app/bin/mex –prefix= /Users/ds293/nlopt MEX_INSTALL_DIR=/Users/ds293/nlopt
- make and make install
- In matlab, addpath('/Users/ds293/nlopt')

Installation of NLopt on Mac:

- Install Xcode
- Follow the instruction of the patch page
- Rename nlopt_optimize.c as nlopt_optimize-mex.c in source code (if necessary)
- in terminal run: ./configure –enable-shared MATLAB= /Applications/MATLAB_R2014b.app MEX=/Applications/MATLAB_R2014b.app/bin/mex –prefix= /Users/ds293/nlopt MEX_INSTALL_DIR=/Users/ds293/nlopt
- make and make install
- In matlab, addpath('/Users/ds293/nlopt')
- Check your Matlab Version and where the mex is stored.
- You can choose the target folder where you want to install the program

Use NLopt:

- Example:

```
obj = @(x) sum((x-20).^2);
opt.algorithm = NLOPT_LN_BOBYQA;
opt.min_objective = obj;
opt.verbose = 1;
xopt = nlopt_optimize(opt,randn(40,1));
```

- For More information: see NLopt wiki page.