Introduction to Structural Econometrics

Lesson 6: Numerical Optimization

Guillaume Wilemme

Aix-Marseille School of Economics - PhD course

Nov 2017

Table of contents

Plan

1. General questions
2. Reminder of the 5 previous sessions
   - What is structural econometrics?
   - A type of model: dynamic programming
   - Basics for another type of estimation: Monte-Carlo methods
   - Simulation-based methods
   - Another type of model: Kalman filter
3. About numerical optimization
   - Gradient-based optimization
   - Derivative-free optimization

Schedule

- October 3rd, room 23, 10-12am
  What is Structural Econometrics?
- October 10th, room 23, 10-12am
  Dynamic Programming and Recursive Models
- October 24th, room 23, 10-12am
  Monte-Carlo Methods
- November 7th, room 23, 10-12am
  Simulation-based Estimation
- November 14th, room 15, 10-12am
  State-Space Representation and Latent Variables
- November 21st, room 16, 10-12am
  Numerical tools

Plan

1. General questions

2. Reminder of the 5 previous sessions
   - What is structural econometrics?
   - A type of model: dynamic programming
   - Basics for another type of estimation: Monte-Carlo methods
   - Simulation-based methods
   - Another type of model: Kalman filter

3. About numerical optimization
   - Gradient-based optimization
   - Derivative-free optimization

Approach based on a model

- Take as given a model $\mathcal{M}(\theta)$
- Hypo. 1 (not testable): the observed data $\mathcal{D}^{obs}$ is a draw from $\mathcal{M}(\theta_0)$
  - The model should be able to produce the dataset, that's why we need stochasticity in general
- looking for $\theta_0^I \subseteq \theta_0$
  - for θ_0^{NI} in $\theta_0 = (\theta_0^I, \theta_0^{NI})$, they are taken as given
  - typically "exogenous" variables X
  - more structural models: interest rates, discount factor, minimum wage, ...
- example linear model and BBPR

What some people call "structural econometrics"

- To model (repeated) choices in a dynamic environment
- Continuous or discrete choices
- Focus on Rust (1987), who observes data on mileage and replacement decisions
  - structure enables him to eventually estimate a demand function for replacement
  - methodological contribution : assumptions required on error terms, estimation strategy (nested partial likelihood)
- Difficult to estimate with standard methods (OLS, GMM, ML)

Monte-Carlo methods

- A way to compute integrals by only using integrands (the functions within the integrals)
  - alternative to quadrature methods
  - more precise, faster
  - drop a deterministic numerical error in quadrature methods for a stochastic error
- Simple idea based on law of large numbers
- In practice, refined methods
  - Importance sampling (a famous example is GHK)
  - Markov Chain Monte-Carlo

From the MC methods to a new way of estimating models

- When estimation is based on expectation (ie integrals) hard to compute or even to formulate
- Generalize the MC method by simulating data $\mathcal{D}^1, ..., \mathcal{D}^S$
- Indirect inference
  - Match simulated data and observed data through the lens of an auxiliary model
  - extend ML and GMM methods as particular cases (called MSL and MSM)
- Some "art" in structural econometrics
  - by opposition to science
  - which method to use, which auxiliary model, which parameters to estimate, back to model crafting

Plan

1. General questions
2. Reminder of the 5 previous sessions
   - What is structural econometrics?
   - A type of model: dynamic programming
   - Basics for another type of estimation: Monte-Carlo methods
   - Simulation-based methods
   - Another type of model: Kalman filter
3. About numerical optimization
   - Gradient-based optimization
   - Derivative-free optimization

Last step: estimate the model on computers

- We have a model
- We have an estimation strategy (ML or indirect inference)
  - $\hat{\theta}^I = \text{argmin}\, Obj(\theta^I; \mathcal{D}^o bs)$
  - *Obj* is "numerically known"
  - There are possibly some constraints on θ^I
- How to compute numerically $\hat{\theta}^I$?

Optimization algorithms

- Another piece of "art" in structural econometrics
- There are many different algorithms
  - some algorithms are not available with some softwares
  - an algorithm can be coded differently on two softwares (and produce different results)
- I present two main families of algorithms to find *local* optimum

Gradient-based optimization

- take an initial guess θ^{l}_{init}, then $Obj(\theta^{l}_{0}) = \min_{\theta^{l}} Obj(\theta^{l}) = \min_{d} Obj(\theta^{l}_{init} + d)$
- finding θ^{l} is finding d
- The idea of the Newton method

The idea of the Newton method

- Local approximation:
  $Obj(\theta^l_{init} + d) = Obj(\theta^l_{init}) + d' G(\theta^l_{init}) + \frac{1}{2} d' H(\theta^l_{init}) d$
- with G the gradient, and H the hessian
- find d^* that minimizes the RHS given θ^l_{init}
- repeat starting from the point $\theta^l_{init} + d^*$
- "line search"
- One dimensional example:
  `https://de.wikipedia.org/wiki/Datei:NewtonIteration_Ani.gif`

Extensions in practice

- In practice, many methods are based on this basic algorithm
- They differ in the way to compute d^*
  - if G and H are not provided by the user, they have to be approximated too
  - high dimensionality, because $dim(H) = dim(\theta^I)^2$
  - need to account for possible constraints

Derivative-free optimization

- Sometimes the objective is not differentiable
  - it can be the case with simulation methods due to stochastic errors
  - avoid non-differentiability if possible
  - the GHK simulator is differentiable, not the naive MC simulator for instance

Nelder-Mead algorithm

- Take a simplex ($n+1$ distinct points in a n-dimensional space)
- Algorithm that modifies the simplex to converge towards an optimum
- Example: `https://en.wikipedia.org/wiki/Nelder%E2%80%93Mead_method`