Lecture V

Numerical Optimization

Gianluca Violante

New York University

Quantitative Macroeconomics

Isomorphism I

- We describe minimization problems: to maximize f minimize $-f$
- If dealing with the minimization problem

$$\min_{x \in [a,b]} f(x)$$

  and you know it is concave, use root-finding on the FOC

$$f'(x) = 0$$

- You do not need to compute the derivative analytically, you can use a finite difference method to approximate f'.

Isomorphism II

- Note that the multivariate root-finding problem

$$\begin{aligned} f^1(x_1, x_2, ..., x_n) &= 0 \\ f^2(x_1, x_2, ..., x_n) &= 0 \\ &... \\ f^n(x_1, x_2, ..., x_n) &= 0 \end{aligned}$$

can be restated as a nonlinear least square problem

$$\min_{\{x_1, x_2, ...x_n\}} \sum_{i=1}^{n} f^i(x_1, x_2..., x_n)^2$$

that can be solved via numerical optimization methods

Basic concepts of numerical optimization

- All methods search through the space of feasible choices generating a sequence of guesses that converges to the solution
- Methods can be divided into two groups:
  - Comparison methods: compute objective at several points and pick the one yielding the smallest value
  - Gradient-based methods: use info on slope and possibly on the curvature (hessian-based): fastest, but unstable and most costly to code
- All methods discussed here yield local minima. We need multiple initializations to explore whether solution is a global minimum
- Grid search: the most primitive. It does not find an exact solution, but it conveys information about the shape of the objective function. Useful first step.

Choosing an optimization algorithm

1. Speed: the major determinant of computation time is the number of function evaluations required by the algorithm to find a local minimum. Gradient based methods generally involve fewer function evaluations than the other methods

2. Robustness to non-smoothness: Gradient based methods get stuck if the objective function is not smooth. Comparison methods better behaved.

3. Robustness to starting parameters: gradient based algorithm started at different starting parameters are more likely to lead to a different local minimum. Stochastic comparison methods more likely to find global minimum.

Bracketing method

- Most reliable for one dimensional problems
- Initialization: find $a < b < c$ such that $f(a), f(c) > f(b)$
  1. Choose $d \in (a, c)$ and compute $f(d)$
  2. Choose new (a, b, c) triplet. If $d < b$ and $f(d) > f(b)$, then there is a minimum in $[d, c]$. Update the triple (a, b, c) with (d, b, c) . If $d < b$ and $f(d) < f(b)$, then the minimum is in $[a, b]$. Update the triple (a, b, c) with (b, d, c) . Otherwise, update the triple (a, b, c) with (a, b, d) .
  3. Stop if $c - a < \delta$. If not, go back to step 1
- Golden search: more sophisticated version that features an optimal way to segment intervals

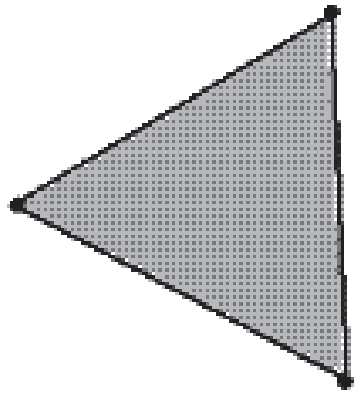
Simplex method

- Comparison methods in n dimensions –also called Nelder-Meade or polytope method. In Matlab: `fminsearch`

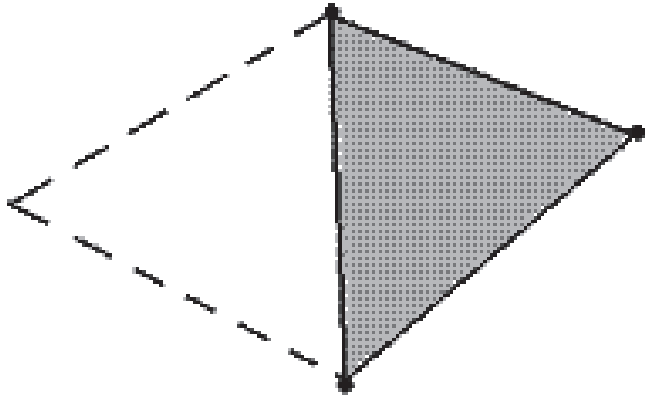
  1. Choose initial simplex $\{x_1, x_2, ..., x_n, x_{n+1}\}$ in $\mathbb{R}^n$
  2. Reorder the simplex vertices in descending order: $f(x_i) \geq f(x_{i+1}) \geq ..., \quad \forall i$
  3. Find the smallest i such that $f(x_i^R) < f(x_i)$, where x_i^R is the reflection of x_i. If exists, replace x_i with x_i^R and go back to step 2. Else, go to step 4.
  4. If width of the current simplex is $< \varepsilon$, stop. Else, go to step 5.
  5. For $i = 1, ..., n$ set $x_i^S = \frac{x_i + x_{i+1}}{2}$ to shrink the simplex. Go back to step 1.

- Other simplex methods use other operations, such as expansions and contractions of vertices around the centroid of the simplex

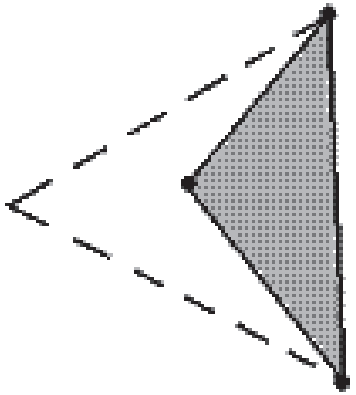
Types of Simplex Moves



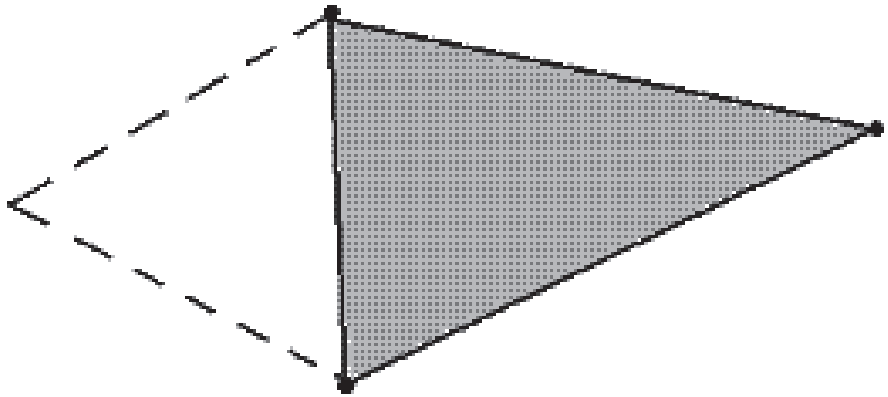
Origonal Simplex



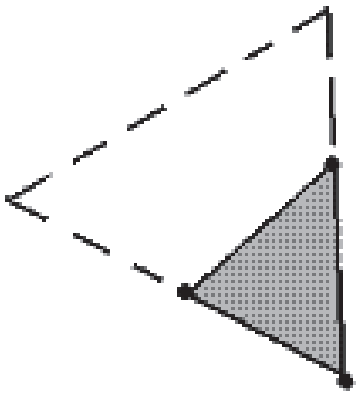
Reflection



Contraction



Reflection and Expansion



Shrinkage

Simulated Annealing: stochastic comparison method

- Best algorithm to find a global minimum in a problem plagued with lots of local ones... but excruciatingly slow

  1. Draw z from $N(0,1)$ and perturb initial guess x_0:

     $$x_1 = x_0 + z\lambda, \quad \text{with } \lambda \text{ given step length}$$

  2. If $f(x_1) < f(x_0)$, accept candidate solution and go to step 3. If $f(x_1) \geq f(x_0)$ accept stochastically, i.e., if:

$$\frac{f(x_1) - f(x_0)}{|f(x_0)|} < \tau c, \quad c \sim U[0,1], \text{ and } \tau > 0 \text{ is temperature parameter}$$

     If not, go back to step 1.

  3. Stop if $|x_1 - x_0|$ is less than tolerance level

- Along the algorithm, decrease $\tau \to 0$ to reduce randomness

Newton/Quasi-Newton methods (one dimension)

- Idea is similar to root-finding Newton methods
- Second-order Taylor approximation of f around x_n:

$$f(x) \simeq f(x^n) + f'(x^n)(x - x^n) + \frac{1}{2} f''(x^n)(x - x^n)^2$$

- Choose $x = x^{n+1}$ that minimizes the approximate function

$$\begin{aligned} f'(x^n) + f''(x^n)(x - x^n) &= 0 \\ x^{n+1} &= x^n - \frac{f'(x^n)}{f''(x^n)} \end{aligned}$$

- It needs evaluation of first and second derivative
- Idea: you move in the direction of the derivative, and if function is very concave you move little, if it is very linear you move a lot.

Multivariate versions

- Updating equation in $\mathbb{R}^n$:

$$\mathbf{x}^{n+1} = \mathbf{x}^n - H\left(\mathbf{x}^n\right)^{-1} \nabla f\left(\mathbf{x}_n\right)$$

- Note that a second-order Taylor appx. around $\mathbf{x}^{n+1}$ yields:

$$f\left(\mathbf{x}^{n+1}\right) \simeq f\left(\mathbf{x}^n\right) + \nabla f\left(\mathbf{x}^n\right)' \left(\mathbf{x}^{n+1} - \mathbf{x}^n\right) + \frac{1}{2}\left(\mathbf{x}^{n+1} - \mathbf{x}^n\right)' H\left(\mathbf{x}^n\right)\left(\mathbf{x}^{n+1} - \mathbf{x}^n\right)$$

and using the updating rule:

$$f\left(\mathbf{x}^{n+1}\right) \simeq f\left(\mathbf{x}^n\right) - \frac{1}{2}\left(\mathbf{x}^{n+1} - \mathbf{x}^n\right)' H\left(\mathbf{x}^n\right)\left(\mathbf{x}^{n+1} - \mathbf{x}^n\right)$$

therefore as long as the Hessian is approximated by a positive definite matrix, the algorithm moves in the right direction

Multivariate version

- Computation of Hessian is time consuming. Nothing guarantees that is positive definite away from the solution
- That's where quasi-Newton methods come handy
- Simplest option: set Hessian to I

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \nabla f\left(\mathbf{x}_n\right)$$

  This method is called steepest descent. Always moving in the right direction, but slow convergence rate.

- BHHH (Berndt, Hall, Hall, and Hausman) method uses the outer product of the gradient vectors to replace the Hessian
- Broyden-Fletcher-Goldfarb-Shanno (BFGS) and Davidon-Fletcher-Powell (DFP) algorithms: secant methods that approximate Hessian with symmetric positive definite matrix

Penalty-function methods

- We wish to solve the minimization problem subject to inequality and equality constraints:

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{x}) \\ &s.t. \\ &g_i(\mathbf{x}) \leq 0,\ i = 1, ..., m \\ &r_j(\mathbf{x}) = 0,\ j = 1, ..., k \end{aligned}$$

- Construct the quadratic penalty function:

$$P(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^{m} \left[\max\{0, g_i(\mathbf{x})\}\right]^2 + \frac{1}{2} \sum_{j=1}^{k} \left[r_j(\mathbf{x})\right]^2$$

which yields quadratic-augmented objective function:

$$h(c, \mathbf{x}) \equiv f(\mathbf{x}) + cP(\mathbf{x})$$

Choice of penalty-parameter

- Each unsatisfied constraint influences $\mathbf{x}$ by assessing a penalty equal to the square of the violation. These influences are summed and multiplied by c the penalty parameter
- This influence is counterbalanced by f(x). If the penalty term is small relative to the $f(\mathbf{x})$, minimization of $h(\mathbf{x})$ will almost certainly not result in an $\mathbf{x}$ feasible to the original problem.
- Starting the problem with a very high value of c yield a feasible solution but causes problems:
  1. almost impossible to tell how large c must be get solution without creating numerical difficulties in the computations
  2. problem is dynamically changing with the relative position of $\mathbf{x}$ and with the subset of the constraints violated
  3. large values of c create enormously steep valleys at the boundaries, and hence formidable convergence difficulties unless algorithm starts extremely close to the minimum

Sequential Unconstrained Minimization Technique

- Start with a relatively small value of c and an infeasible point. This way no steep valleys are present in the initial optimization of $(c, \mathbf{x})$.
- Next, solve a sequence of problems with monotonically increasing values of c chosen so that the solution to each new problem is "close" to the previous one.

  1. Initialization Step: Select a growth parameter $\eta > 1$, a stopping parameter $\varepsilon > 0$, and an initial penalty value c^0.
  2. At iteration t minimize $h\left(c^t, \mathbf{x}\right)$. Call this solution $\mathbf{x}^t$. If convergence reached, stop, if not set $c^{t+1} = (1+\eta)\, c^t$ and solve the minimization problem starting from $\mathbf{x}^t$.

- Must scale the constraints so that the penalty generated by each one is about the same magnitude
- Choose c^0 so that penalty term is same size as objective function

Proof of convergence of SUMT

- Easy to prove that $\{f\left(c^t, \mathbf{x}^t\right)\}$ decreasing sequence:

$$\begin{aligned} h\left(c^{t+1}, \mathbf{x}^{t+1}\right) = f\left(\mathbf{x}^{t+1}\right) + c^{t+1} P\left(\mathbf{x}^{t+1}\right) &\geq f\left(\mathbf{x}^{t+1}\right) + c^{t} P\left(\mathbf{x}^{t+1}\right) \\ &\geq f\left(\mathbf{x}^{t}\right) + c^{t} P\left(\mathbf{x}^{t}\right) = h\left(c^{t}, \mathbf{x}^{t}\right) \end{aligned}$$

- From this, we have

$$\begin{aligned} f\left(\mathbf{x}^{t+1}\right) + c^{t} P\left(\mathbf{x}^{t+1}\right) &\geq f\left(\mathbf{x}^{t}\right) + c^{t} P\left(\mathbf{x}^{t}\right) \\ f\left(\mathbf{x}^{t}\right) + c^{t+1} P\left(\mathbf{x}^{t}\right) &\geq f\left(\mathbf{x}^{t+1}\right) + c^{t+1} P\left(\mathbf{x}^{t+1}\right) \end{aligned}$$

  since the RHS are both minima

- Now, subtract bottom-right from top-left and bottom left from top right to prove that $P\left(\mathbf{x}^{t+1}\right) \leq P\left(\mathbf{x}^{t}\right)$

Proof of convergence of SUMT

$$
\begin{aligned}
f\left(\mathbf{x}^{t+1}\right)+c^{t}P\left(\mathbf{x}^{t+1}\right)-f\left(\mathbf{x}^{t+1}\right)-c^{t+1}P\left(\mathbf{x}^{t+1}\right) &\geq f\left(\mathbf{x}^{t}\right)+c^{t}P\left(\mathbf{x}^{t}\right)-f\left(\mathbf{x}^{t}\right) \\
&\quad -c^{t+1}P\left(\mathbf{x}^{t}\right) \\
c^{t}P\left(\mathbf{x}^{t+1}\right)-c^{t+1}P\left(\mathbf{x}^{t+1}\right) &\geq c^{t}P\left(\mathbf{x}^{t}\right)-c^{t+1}P\left(\mathbf{x}^{t}\right) \\
\left(c^{t+1}-c^{t}\right)P\left(\mathbf{x}^{t+1}\right) &\leq \left(c^{t+1}-c^{t}\right)P\left(\mathbf{x}^{t}\right) \\
P\left(\mathbf{x}^{t+1}\right) &\leq P\left(\mathbf{x}^{t}\right)
\end{aligned}
$$

- This result together with

$$
f\left(\mathbf{x}^{t+1}\right)+c^{t}P\left(\mathbf{x}^{t+1}\right) \geq f\left(\mathbf{x}^{t}\right)+c^{t}P\left(\mathbf{x}^{t}\right) \quad \rightarrow \quad f\left(\mathbf{x}^{t+1}\right)\geq f\left(\mathbf{x}^{t}\right)
$$

because we are moving from outside to inside the feasible region

$$
f\left(\mathbf{x}^{*}\right)=f\left(\mathbf{x}^{*}\right)+c^{t}P\left(\mathbf{x}^{*}\right) \geq f\left(\mathbf{x}^{t}\right)+c^{t}P\left(\mathbf{x}^{t}\right)\geq f\left(\mathbf{x}^{t}\right)
$$

- Thus, $P\left(\mathbf{x}^{t}\right)\rightarrow P\left(\mathbf{x}^{*}\right)$ and $f\left(\mathbf{x}^{t}\right)\rightarrow f\left(\mathbf{x}^{*}\right)$

Barrier methods

- The penalty method searches always outside the feasible region
- An alternative that searches always inside is the barrier method based on the augmented objective function

$$h\left(c,\mathbf{x}\right)\equiv f\left(\mathbf{x}\right)+c\sum_{i=1}^{m}\log\left[-g_{i}\left(\mathbf{x}\right)\right]$$

  where you start from a large value of c.

- At each subsequent iteration, the value of c is monotonically decreased and $\mathbf{x}$ approaches boundaries
- Penalty (exterior) methods better because barrier (interior) methods cannot handle equality constraints
- Issue with penalty method: need to define f smoothly outside the feasible region.

Karush-Kuhn-Tucker Methods

- This method directly solves the KKT system of (nonlinear) equations associated with the problem by a kind of the Newton or Quasi-Newton approach.

- Example with equality constraints:

$$\min_{\mathbf{x},\lambda} L(\mathbf{x},\lambda) = f(\mathbf{x}) + \lambda^T r(\mathbf{x})$$

- The system of KT conditions is:

$$\begin{aligned} \nabla_x L(\mathbf{x},\lambda) &= 0 \rightarrow \nabla f(\mathbf{x}) + \nabla r(\mathbf{x})^T \lambda = 0 \\ \nabla_\lambda L(\mathbf{x},\lambda) &= 0 \rightarrow r(\mathbf{x}) = 0 \end{aligned}$$

- We have a nonlinear system of $n+k$ equations in $(n+k)$ unknowns that can be solved as we studied.