Lecture 6: Optimization

Yuya Takahashi

April 23, 2019

1 Introduction

- Optimization is the central behavioral assumption in economics. But it is rare that we have a closed-form solution for optimality. Thus, we need to find a minimum/maximum by computation.

- There are many different methods: derivative-free methods (e.g., Nelder-Mead), direction set methods (e.g., Newton), stochastic search (simulated annealing). There is no method that performs the best in all situations.

- In practice, we often combine more than one methods to find the optimum.

- We focus on unconstrained optimization.

2 Derivative-Free Methods

- A standard problem is

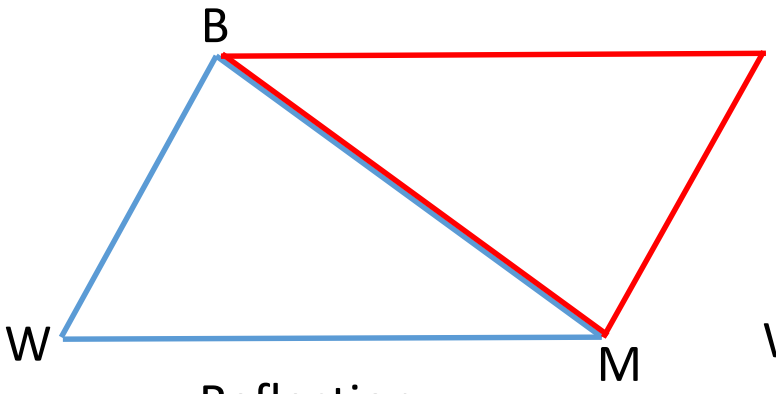
$$\min_x f(x) \tag{1}$$

  where x could be either a scaler or a vector.

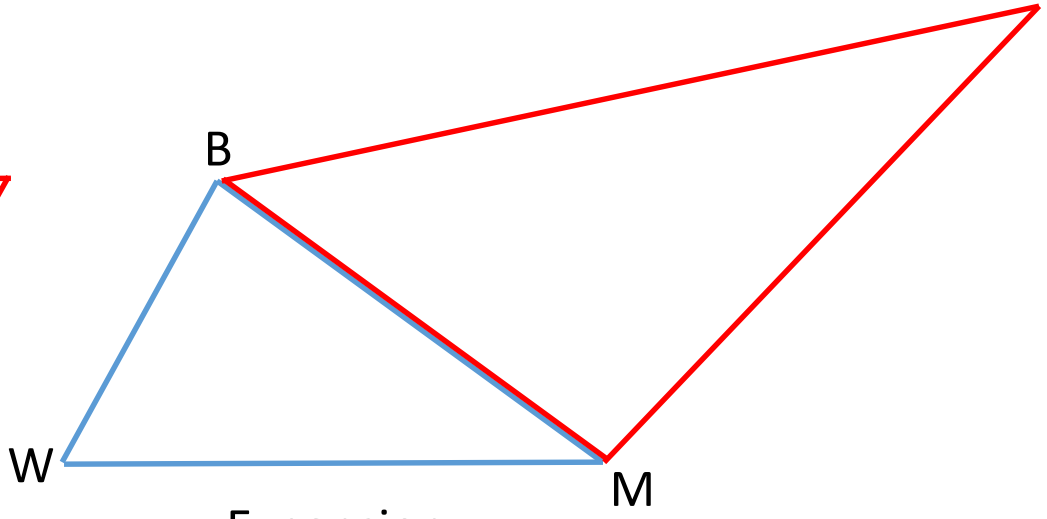
- Maximization problems can be written in this form.

- Problems are often accompanied by equality or inequality constraints

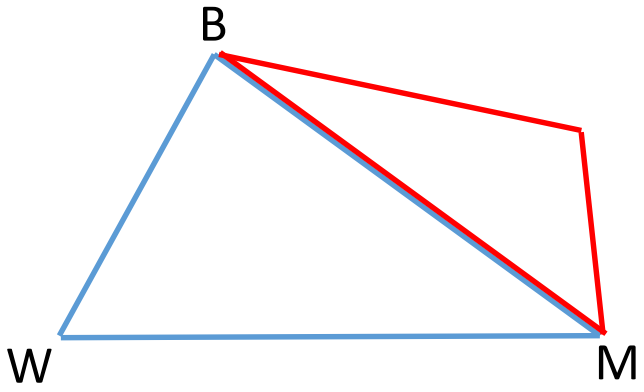
- One of the widely used methods is Nelder-Mead method. To fix the idea, consider an example where x is of two dimensions. Assume we have three points B (best), M (middle), and W (worst).



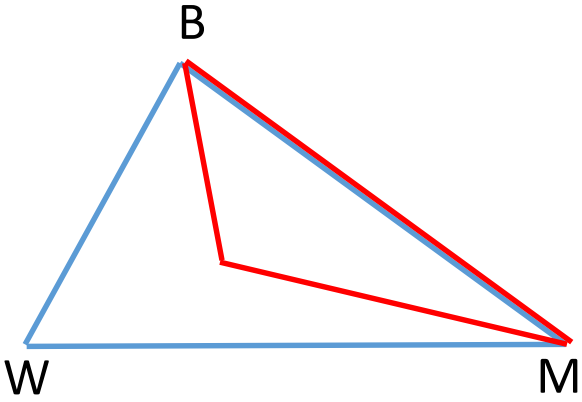
Reflection



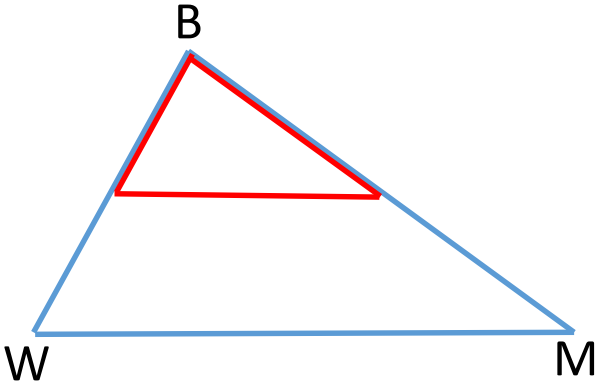
Expansion



Outside Contraction



Inside Contraction



Shrinkage

- In the standard Nelder-Mead method, we keep track of $n+1$ points of interest in R^n, whose convex full forms a simplex.

- Denote $n+1$ vertices of the current simplex by $\{x_1, x_2, ..., x_{n+1}\}$ where the ordering is such that

$$f(x_1) \leq f(x_2) \leq \cdots \leq f(x_{n+1}).$$

- The basic idea is to remove the vertex with the "worst" function value and replace it with another point with a better value.

- The centroid of the best n points

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

- Points along the line that joins $\bar{x}$ and the worst vertex x_{n+1} are denoted by

$$\bar{x}(t) = \bar{x} + t(x_{n+1} - \bar{x}).$$

- In the previous example, reflection, expansion, outside contraction, inside contraction corresponds to -1, -2, $-\frac{1}{2}$, and $\frac{1}{2}$, respectively.

- **The Nelder-Mead Simplex Algorithm**

Compute the reflection point $\bar{x}(-1)$ and evaluate $f_{-1} = f(\bar{x}(-1))$.

- If $f(x_1) \leq f_{-1} < f(x_n)$,

  - replace x_{n+1} by $\bar{x}(-1)$ and go to next iteration;

- Else if $f_{-1} < f(x_1)$

- Compute the expansion point $\bar{x}(-2)$ and evaluate $f_{-2} = f(\bar{x}(-2))$.
  - If $f_{-2} < f_{-1}$
    - replace x_{n+1} by x_{-2} and go to the next iteration.
  - else
    - replace x_{n+1} by x_{-1} and go to the next iteration.

- else if $f_{-1} \geq f(x_n)$
  - if $f(x_n) \leq f_{-1} < f(x_{n+1})$
    - evaluate $f_{-1/2} = f(\bar{x}(-1/2))$;
      - if $f_{-1/2} \leq f_{-1}$

        replace x_{n+1} by $x_{-1/2}$ and go to next iteration;
  - else

- evaluate $f_{1/2} = f\left(\bar{x}\left(1/2\right)\right)$;

  - if $f_{1/2} \leq f_{n+1}$

    replace x_{n+1} by $x_{1/2}$ and go to next iteration;

- If neither outside nor inside contraction was acceptable, shrink the simplex toward x_1

  - $x_i \leftarrow (1/2)(x + x_i)$ for $i = 2, 3, ..., n + 1$.

Remarks • Unless a shrinkage is performed, the average function value decreases at each step

- This algorithm works reasonably well in many situations:

  – where the function is not continuous

  – where the function has a lot of kinks

- But stagnation often occurs at nonoptimal points
- This method is local in nature
- Convergence is slow
- One way to choose initial vertices: for $(a_1, ..., a_n)$, we can construct additional n points by

$$
\begin{aligned}
&(a_1 + \frac{1}{2}|a_1|, a_2, ..., a_n) \\
&(a_1, a_2 + \frac{1}{2}|a_2|, ..., a_n) \\
&\vdots \\
&(a_1, a_2, ..., a_n + \frac{1}{2}|a_n|)
\end{aligned}
$$

3 Newton Methods

- Let

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x), ..., \frac{\partial f}{\partial x_n}(x) \right)$$

and

$$H(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(x) \end{pmatrix}$$

- If $f(x) = \frac{1}{2} x^T A x + b^T x$ where A is symmetric positive definite, the minimum is at

$$x^* = -A^{-1} b. \tag{2}$$

- If f is convex but not quadratic, we generally cannot directly solve for its minimum as above.

- But we can replace f locally with a quadratic approximation and apply the solution (2).

- At x^k, we define

$$g(x) = f\left(x^k\right) + \nabla f(x^k)(x - x^k) + \frac{1}{2}\left(x - x^k\right)^T H\left(x^k\right)\left(x - x^k\right)$$

- If f is convex, $H(x^k)$ is positive definite, and $g(x)$ has a minimum at $x^k - H\left(x^k\right)^{-1}\left(\nabla f(x^k)\right)^T$. Naturally, Newton's method iterates based on this:

$$x^{k+1} = x^k - H\left(x^k\right)^{-1}\left(\nabla f(x^k)\right)^T.$$

- Newton's method applies this update even when $H(x^k)$ is not positive definite.

- Newton's method is a local method

- Implementation can be expensive because computing and storing the Hessian is time- and space-consuming

- In practice, we never calculate $H\left(x^k\right)^{-1}$. Instead, we solve the linear problem $H(x^k)s^k = -\left(\nabla f(x^k)\right)^T$ for s^k and set $x^{k+1} = x^k + s^k$.

- Even with a positive-definite Hessian, x^k could be moving uphill.

- Convergence is faster than simplex methods.

- After convergence, we should check if $H(x)$ satisfies the second order condition.

4 Direction Set Methods

- The basic idea is to find a sequence of directions. In each direction, we repeat one-dimensional minimization.

- Introduce *line search*: choose a direction p_k and searches along this direction from the current iterate x^k for new iterate with a lower function value

- The distance to move along p_k can be found by solving

$$\min_{\alpha>0} f\left(x^k + \alpha p_k\right)$$

- Note this is a one-dimensional minimization problem.

- Suppose for now that we know a sequence of directions s^k.

- Generic Direction Method Algorithm:

1. Choose initial guess x^0 and $\delta, \varepsilon > 0$.
2. Compute a search direction s^k.
3. Solve $\lambda_k = \arg\min_\lambda f\left(x^k + \lambda s^k\right)$.
4. Set $x^{k+1} = x^k + \lambda_k s^k$.
5. If $||x^k - x^{k+1}|| \leq \varepsilon\left(1 + ||x^k||\right)$, go to the next step. Otherwise, go back to step 2.
6. If $||\nabla f\left(x^k\right)|| \leq \delta\left(1 + f(x^k)\right)$, stop and declare "success". Otherwise, stop and report "converge to a nonoptimal point".

- The question is, "how do we choose a direction?"

- *Coordinate Directions*: cycle through the n coordinate directions $e_1, e_2, ..., e_n$, obtaining new iterates by performing a line search along each direction in turn.

- *Steepest Descent Directions*: take the direction along which f falls most rapidly per unit length.

- The search direction is

$$s^{k+1} = -(\nabla f\left(x^k\right))^T$$

- These are generally a slow algorithm. For example, in the steepest descent method the sequence zigzags into the solution; "lack of multivariate view".

- Is there any way to find better directions?

- *Newton's Method with Line Search*: the newton method is a special case of a direction set method with

$$s^k = -H\left(x^k\right)^{-1} \nabla \left(f(x^k)\right)^T$$

and $\lambda_k = 1$.

- Basic idea: Newton method gives good directions, but the quality of approximation gets worse as we move away from x^k. Thus, we take the Newton step direction and use line search to find the best point in that direction.

- Newton's method ($\lambda_k = 1$) may overshoot the best point in the Newton direction and push the iteration uphill. On the other hand, if f is smooth, Newton's method with line search guarantees convergence to a local minimum.

- Newton's method ($\lambda_k = 1$) has quadratic convergence. Thus, as we approach a solution, it is better to switch to Newton's method.

5 Quasi-Newton Methods

- Take Newton method with line search and replaces the Hessian with another matrix of the same size, which generates the search direction as Hessian did.

- The approximate Hessian should be:

    1. positive semidefinite

    2. easier to compute and invert

- One well known method to construct H_k is called Broyden-Fletcher-Goldfarb-Shanno (BFGS) updates.

- BFGS Algorithm:

    1. Choose initial guess x^0, initial Hessian guess H^0, and $\delta, \varepsilon > 0$.

2. Solve $H_k s^k = -(\nabla f\left(x^k\right))^T$ to find the search direction s^k.

3. Solve $\lambda_k = \arg\min_\lambda f\left(x^k + \lambda s^k\right)$.

4. Set $x^{k+1} = x^k + \lambda s^k$.

5. Let

$$z_k = x^{k+1} - x^k$$
$$y_k = (\nabla f\left(x^{k+1}\right))^T - (\nabla f\left(x^k\right))^T$$

and update H_{k+1} by

$$H_{k+1} = H_k - \frac{H_k z_k z_k^T H_k}{z_k^T H_k z_k} + \frac{y_k y_k^T}{y_k^T z_k}$$

6. If $||x^k - x^{k+1}|| \leq \varepsilon\left(1 + ||x^k||\right)$, go to the next step. Otherwise, go back to step 2.

7. If $||\nabla f\left(x^k\right)|| \leq \delta\left(1 + f(x^k)\right)$, stop and declare "success". Otherwise, stop and report "converge to a nonoptimal point".

- H_k is positive definite and symmetric as long as H_0 is. This implies we can apply Cholesky factorization in step 2.

- H_k sequence does not necessarily converge to the true Hessian at the solution.

- We want to switch to Newton's method near the solution. This is necessary especially when we use H_k for inference purpose (e.g., MLE).

6 Stochastic Search

- Simulated Annealing is a method that uses the objective function to create a nonhomogeneous Markov chain process that will asymptotically converge to the minimum of $f(x)$.

- Construct a Markov process in the following way:

  1. For given x^k, draw a $y \in V_x$ where V_x is the set of points in neighborhood x.

  2. If $f(y) < f(x_k)$, then $x_{k+1} = y$

  3. If $f(y) \geq f(x_k)$, then

$$\begin{cases} \Pr\{x_{k+1} = y\} = e^{-(f(y)-f(x_k))/T_k} \\ \Pr\{x_{k+1} = x_k\} = 1 - \Pr\{x_{k+1} = y\}. \end{cases}$$

- In words, if $f(y) < f(x_k)$, we jump to y. But if $f(y) \geq f(x_k)$, we do so only probabilistically.

- T_k is called the temperature, and controls the probability of jump.

- This is a **global** search.

- Slow in conventional optimization problems (say f is continuous), but works well where conventional methods do not apply