Convex Optimization II

Lecture 7: Gradient Methods

Hamed Shah-Mansouri

Department of Electrical Engineering

Sharif University of Technology

1400-2

Reference

[1] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[2] D. P. Bertsekas, *Nonlinear Programming*, second edition, Athena Scientific, 1999.

Most of the materials and figures in this lecture are from Chapter 9 in [1]. A few figures are from Chapter 1 in [2].

Thanks to Professors Stephen Boyd and Dimitri Bertsekas for the slides used in this lecture.

How to Solve a Convex Optimization Problem?

- Using CVX, MOSEK, MATLAB, CPLEX, and other efficient optimization software, you do not really need to implement the solver yourself.
- Your main job is to formulate your problem as a convex optimization problem and actually prove that it is indeed a convex problem.
- However, it is still good to have some idea on how you may numerically solve a convex optimization problem.
- For the rest of this lecture, we look at some numerical techniques to solve unconstrained and constrained convex optimization problems.

Unconstrained Minimization Problems

$$\text{minimize } f(x) \tag{1}$$

- $f : \mathbf{R}^n \to \mathbf{R}$ is convex, twice continuously differentiable (hence $\mathbf{dom} f$ is open)
- Optimal value $p^\star = \inf_x f(x)$ is attained (and finite)
- A necessary and sufficient condition for $x^\star$ to be optimal is

$$\nabla f(x^\star) = 0. \tag{2}$$

- In some special cases, we can find a solution to the problem (1) by analytically solving equation (2).
- Usually, it has to be solved by an iterative algorithm, which computes a sequence of points $x^{(0)}, x^{(1)}, \ldots \in \mathbf{dom} f$ with $f(x^{(k)}) \to p^\star$ as $k \to \infty$.
- The algorithm is terminated when $f(x^{(k)}) - p^\star \leq \epsilon$, where $\epsilon > 0$ is some specified tolerance.

Bisection Method for Single Variable Case

$$\text{minimize } f(x)$$

- $f : \mathbf{R} \to \mathbf{R}$ is convex and differentiable.
- Idea: Whether the slope (derivative) is positive or negative at a trial solution definitely indicates whether improvement lies immediately to the left or right, respectively.

Notations:

- x' is the current trial solution
- $\underline{x}$ is the lower bound on $x^\star$
- $\overline{x}$ is the upper bound on $x^\star$
- ϵ is the error tolerance on $x^\star$

Bisection Method for Single Variable Case

- Initialization: Select ϵ. Find an initial $\underline{x}$ and $\overline{x}$ by inspection (or by respectively finding any value of x at which the derivative is negative and then positive).
- Select an initial trial solution $x' = \frac{\underline{x}+\overline{x}}{2}$.
- Iteration
  1. Evaluate $\frac{df(x)}{dx}$ at $x = x'$.
  2. If $\frac{df(x)}{dx} \leq 0$, reset $\underline{x} = x'$.
  3. If $\frac{df(x)}{dx} \geq 0$, reset $\overline{x} = x'$.
  4. Select a new $x' = \frac{\underline{x}+\overline{x}}{2}$.
- Stopping Rule: If $\overline{x} - \underline{x} \leq 2\epsilon$, so that the new x' must be within ϵ of $x^\star$, stop. Otherwise, perform another iteration.

Newton's Method for Single Variable Case

$$\text{minimize } f(x)$$

- $f : \mathbf{R} \to \mathbf{R}$ is convex and twice differentiable.
- Idea: Approximate $f(x)$ within the neighborhood of the current trial solution by a quadratic function and then minimize the approximate function exactly to obtain the new trial solution to start the next iteration.

$$f(x^{(k+1)}) \approx f(x^{(k)}) + f'(x^{(k)})(x^{(k+1)} - x^{(k)}) + \frac{f''(x^{(k)})}{2}(x^{(k+1)} - x^{(k)})^2$$

- Having fixed $x^{(k)}$ at the beginning of iteration k, functions $f(x^{(k)})$, $f'(x^{(k)})$, and $f''(x^{(k)})$ are also fixed.
- The first derivative is

$$f'(x^{(k+1)}) \approx f'(x^{(k)}) + f''(x^{(k)})(x^{(k+1)} - x^{(k)})$$

- Setting the first derivative to zero yields

$$x^{(k+1)} = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})}$$

Newton's Method for Single Variable Case

- Initialization: Select ϵ. Set $k = 1$. Find an initial $x^{(1)}$ by inspection.
- Iteration k

  (1) Calculate $f'(x^{(k)})$ and $f''(x^{(k)})$.

  (2) Set $x^{(k+1)} = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})}$.
- Stopping Rule: If $|x^{(k+1)} - x^{(k)}| \leq \epsilon$, stop. Otherwise, set $k := k + 1$ and perform another iteration.
- Another stopping criterion can be $|f(x^{(k+1)}) - f(x^{(k)})|$ is sufficiently small.

Descent Methods

- Consider the problem of unconstrained minimization of a continuously differentiable convex function $f : \mathbf{R}^n \rightarrow \mathbf{R}$.

- Iterative Descent: Start at some point $x^{(0)}$ (an initial guess) and successively generate vectors $x^{(1)}, x^{(2)}, \ldots$, such that f is decreased at each iteration

$$f(x^{(k+1)}) < f(x^{(k)}), \quad k = 0, 1, \ldots,$$

  except when $x^{(k)}$ is optimal.

- We successively improve our current solution estimate and we hope to decrease f all the way to its minimum.

Descent Methods

$$x^{(k+1)} = x^{(k)} + t^{(k)}\Delta x^{(k)} \quad \text{with } f(x^{(k+1)}) < f(x^{(k)})$$

- other notations: $x^+ = x + t\Delta x$, $x := x + t\Delta x$
- Δx is the *step*, or *search direction*; t is the *step size*, or *step length*
- from convexity, $f(x^+) < f(x)$ implies $\nabla f(x)^T \Delta x < 0$ ($i.e.$, Δx is a *descent direction*)

General descent method.

given a starting point $x \in \textbf{dom}\, f$.
repeat

1. Determine a descent direction Δx.
2. *Line search.* Choose a step size $t > 0$.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

Line Search Types

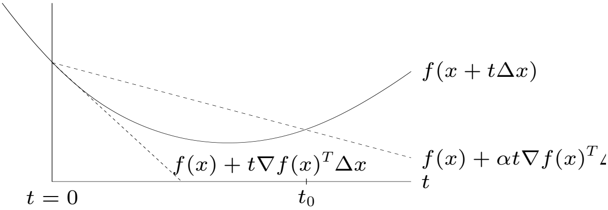
exact line search: $t = \text{argmin}_{t>0} f(x + t\Delta x)$

backtracking line search (with parameters $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$)

- starting at $t = 1$, repeat $t := \beta t$ until

$$f(x + t\Delta x) < f(x) + \alpha t \nabla f(x)^T \Delta x$$

- graphical interpretation: backtrack until $t \leq t_0$



- Typical value of α: between 0.01 and 0.3
- Typical value of β: between 0.1 and 0.8

Gradient Descent Method

general descent method with $\Delta x = -\nabla f(x)$

---

given a starting point $x \in \mathbf{dom}\, f$.
repeat
1. $\Delta x := -\nabla f(x)$.
2. *Line search.* Choose step size t via exact or backtracking line search.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

- stopping criterion usually of the form $\|\nabla f(x)\|_2 \le \epsilon$

Constrained Optimization

- What if the optimization problem is constrained? That is, what if we need to solve the following problem

$$\text{minimize}_{x \in \mathcal{X}} \;\; f(x),$$

where $\mathcal{X}$ is the feasible set.

- Even if we start inside the feasible set $\mathcal{X}$, an update can take us outside that set.

- A simple way to tackle this problem is to project back to the set $\mathcal{X}$ whenever such a situation arises.

PROJECTION

- We use the notation $[x]^+$ to denote the orthogonal projection (with respect to the Euclidean norm) of a vector x onto the convex set $\mathcal{X}$.
- In particular, $[x]^+$ is defined as

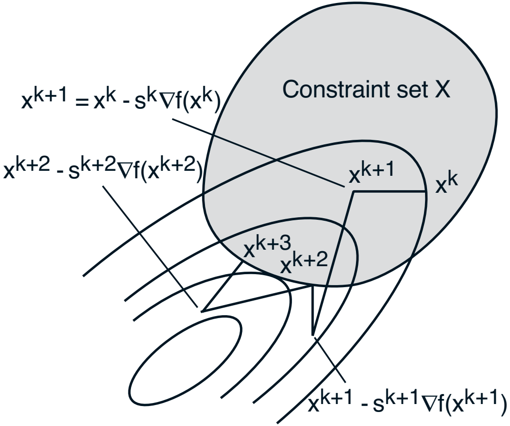
$$[x]^+ = \arg\min_{z \in \mathcal{X}} \|z - x\|_2.$$

- For every $x \in \mathbf{R}^n$, there exists a unique $z \in \mathcal{X}$ that minimizes $\|z - x\|_2$ over all $z \in \mathcal{X}$, and will be denoted by $[x]^+$.
- Example: consider the box constraints

$$\mathcal{X} = \{\mathbf{x} \mid \alpha_i \leq x_i \leq \beta_i, \;\; i = 1, \ldots, n\}$$

  What is $[x]_i^+$?

Gradient Projection



Gradient Projection Method

- The gradient projection algorithm generalizes the gradient algorithm to the case where there are constraints, and is described by the following equation:

$$x^{(k+1)} = \left[x^{(k)} - \gamma \nabla f(x^{(k)})\right]^+,$$

where γ is a positive stepsize.

CONVERGENCE

- In this lecture, we did not prove the convergence of the studied algorithms.
- We will have a lecture on convergence analysis techniques soon. In that lecture, we will study the convergence of the gradient and gradient projection methods.

Summary

- Unconstrained Minimization
- Descent Methods
- Line Search
- Gradient Descent Method
- Gradient Projection Method
- Reading: Sections $9.2 - 9.4$ in Boyd and Vandenberghe.