SDSC6015 - Semester A, 2025

Stochastic Optimization for Machine Learning

4. Gradient Descent

1 Basics

Definition 1 (Big O notation). *$f(x) = \mathcal{O}(g(x))$ as $x \to x_0$ ($x_0 = 0$ without specification) if there exist positive numbers δ and M such that*

$$|f(x)| \leq Mg(x) \quad when \quad 0 < |x - x_0| < \delta.$$

Definition 2 (Little o notation). *$f(x) = o(g(x))$ as $x \to x_0$ ($x_0 = 0$ without specification) if*

$$\lim_{x \to x_0} \frac{f(x)}{g(x)} = 0$$

Theorem 1 (Taylor Expansion - $\mathcal{C}^2$). *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable (i.e. f, ∇f, $\nabla^2 f$ all exist and continuous), then*

$$f(\boldsymbol{x}) = f(\boldsymbol{x}_0) + \nabla f(\boldsymbol{x}_0)^\top (\boldsymbol{x} - \boldsymbol{x}_0) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_0)^\top \nabla^2 f(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0) + o(\|\boldsymbol{x} - \boldsymbol{x}_0\|_2^2)$$

2 Optimality Conditions

Theorem 2 (First Order Necessary Condition for Local Optimality). *Consider a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$. If $\boldsymbol{x}^\star \in \mathbb{R}^n$ is an unconstrained local minimizer, then*

$$\nabla f(\boldsymbol{x}^\star) = \boldsymbol{0}$$

Note:

- This condition is necessary but *NOT sufficient* for local optimality.

- If f is convex: this is sufficient and for global optimality.

- A point $\boldsymbol{x}$ that satisfies $\nabla f(\boldsymbol{x}) = 0$ is called the *stationary point or a critical point.*

Theorem 3 (Second Order Necessary Condition for Local Optimality)**.** *Consider a twice continuously differentiable function* $f : \mathbb{R}^n \to \mathbb{R}$*. If* $\boldsymbol{x}^\star \in \mathbb{R}^n$ *is an unconstrained local minimizer, then* $\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$ *and*

$$\nabla^2 f(\boldsymbol{x}^\star) \succeq \mathbf{0}$$

Theorem 4 (Second Order Sufficient Condition for Local Optimality)**.** *Consider a twice continuously differentiable function* $f : \mathbb{R}^n \to \mathbb{R}$*. If* $\boldsymbol{x}^\star \in \mathbb{R}^n$ *where* $\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$ *and*

$$\nabla^2 f(\boldsymbol{x}^\star) \succ \mathbf{0},$$

then $\boldsymbol{x}^\star$ *is (strict) local minimizer of* f*.*

Note:

- Necessary is not sufficient!

- Sufficient is not necessary!

3 Bisection Algorithm

3.1 1D Case

Consider a function $f : \mathbb{R} \to \mathbb{R}$ is continuously differentiable convex function. We denote $x^\star$ to its minimizer, and suppose we know that $x^\star \in [x_{\text{low}}, x_{\text{up}}]$.

Algorithm 1: Bisection Algorithm for 1D Case

Input: Search range $[x_{\text{low}}, x_{\text{up}}]$. User-specified tolerance $\epsilon > 0$.

1. Set $x = (x_{\text{low}} + x_{\text{up}})/2$,
   a. $f'(x) = 0$ then return x as the solution and stop.
   b. $f'(x) > 0$ then set $x_{\text{up}} = x$.
   c. $f'(x) < 0$ then set $x_{\text{low}} = x$.
2. If $x_{\text{up}} - x_{\text{low}} \leq \epsilon$, stop, else back to Step 1.

Note:

- the length of the interval is half at each iteration
- To reduce the interval with length $\epsilon > 0$, it takes $\mathcal{O}\left(\log\left(\frac{x_{\text{up}} - x_{\text{low}}}{\epsilon}\right)\right)$ iterations.

- this is actually a root finding algorithm

Bisection in Optimization

Consider the epigraph formulation of an optimization problem

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n, t\in\mathbb{R}}{\text{minimize}} \quad & t \\ \text{subject to} \quad & f_0(\boldsymbol{x}) \leq t \\ & f_i(\boldsymbol{x}) \leq 0, \quad \forall i \in [m] \end{aligned}$$

where $f_0 : \mathbb{R}^n \to \mathbb{R}$, $f_i : \mathbb{R}^n \to \mathbb{R}$ for $i \in [m]$. We assume that for any fixed $t \in \mathbb{R}$, we can answer the following question

Feasibility Question: *Can we find a $\boldsymbol{x} \in \mathbb{R}^n$ where $f_0(\boldsymbol{x}) \leq t$ and $f_i(\boldsymbol{x}) \leq 0$, $\forall j \in [m]$? If so, return $\boldsymbol{x}$. If not, tell us that this is not possible.*

Algorithm 2a: General Version

1. Start with some $t \in \mathbb{R}$.
2. Answer "Feasibility Question" with t
3. If yes from "Feasibility Question", decrease t.
   If no from "Feasibility Question", increase t.

Note:

- We assume there is a black box to answer the "Feasibility Question".
- Whether we can solve the problem efficiently depends on the complexity (computational difficulty) of answering the "Feasibility Question".

Algorithm 2b: Bisection Version

0. Assume the optimal value is within $[t_{\text{low}}, t_{\text{up}}]$. We have user-specified tolerance $\epsilon > 0$
1. Start with $t = (t_{\text{up}} + t_{\text{low}})/2$
2. Answer "Feasibility Question" with t
   - 2a. If yes, set $t_{\text{up}} = t$.
   - 2b. If no, set $t_{\text{low}} = t$.
3. If $t_{\text{up}} - t_{\text{low}} \leq \epsilon$, stop.

4 Line Search Method

Consider an unconstrained, smooth convex optimization problem

$$\underset{\boldsymbol{x} \in \mathbb{R}^n}{\text{minimize}}\ f(\boldsymbol{x}),$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable with $\mathbf{dom}(f) = \mathbb{R}^n$.

In line search method, for each iteration, we do

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \cdot \boldsymbol{d}_k$$

where

- $k \in \mathbb{Z}_+$: the index of iterations
- $\boldsymbol{x}_k$: solution at the k^{th} iteration
- $\boldsymbol{d}_k$: direction at the k^{th} iteration
- α_k: stepsize at the k^{th} iteration

It's about designing α_k's and $\boldsymbol{d}_k$'s!

4.1 Descent Direction

Definition 3 (descent direction)**.** *Given $\boldsymbol{x} \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$, a direction $\boldsymbol{d} \in \mathbb{R}^n$ is called a descent direction if there exists $\alpha_0 > 0$ such that*

$$f(\boldsymbol{x} + \alpha \cdot \boldsymbol{d}) < f(\boldsymbol{x}), \qquad \forall \alpha \in (0, \alpha_0).$$

Lemma 1. *Given $\boldsymbol{x} \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$, among all directions from $\boldsymbol{x}$, the direction $\boldsymbol{d} = -\nabla f(\boldsymbol{x})$ gives the maximum rate of decrease in terms of the value of f.*

Lemma 2. *Given $\boldsymbol{x} \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$, any direction $\boldsymbol{d} \in \mathbb{R}^n$ satisfying*

$$\langle \boldsymbol{d}, \nabla f(\boldsymbol{x}) \rangle < 0$$

is a descent direction at $\boldsymbol{x}$.

Lemma 3. *Given $\boldsymbol{x} \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$, $\boldsymbol{S} \in \mathbb{S}^n_{++}$, the direction $-\boldsymbol{S}\nabla f(\boldsymbol{x}) \in \mathbb{R}^n$ is a descent direction at $\boldsymbol{x}$.*

Note: Now, we can restrict the algorithm to

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \cdot \boldsymbol{S}_k \nabla f(\boldsymbol{x}_k)$$

where $\boldsymbol{S}_k \in \mathbb{S}^n_{++}$, for $k \in \mathbb{Z}_+$.

4.2 Stepsize Strategy

When picking stepsize α_k, below are the common strategies:

- **Constant stepsize:** Picking $\alpha_k = \alpha$ for all k. The is the easiest rule to implement; however, If α is too large, the algorithm might not converge. If α is too small, the algorithm will converge slowly.

- **Diminishing stepsize:** Picking the stepsizes $\{\alpha_k\}$ such that $\alpha_k \to 0$ as $k \to \infty$ and $\sum_{k=1}^{\infty} \alpha_k = \infty$ (guarantee progress not too slow).

- **Exact line search:** At each iteration k, pick the stepsize $\alpha_k = \arg\min_{\alpha \geq 0} f(\boldsymbol{x} + \alpha \cdot \boldsymbol{d}_k)$.

  Question: How do we solve this?

- **Inexact line search:** At each iteration k, pick the stepsize $\alpha_k = \arg\min_{0 \leq \alpha \leq \overline{\alpha}} f(\boldsymbol{x} + \alpha \cdot \boldsymbol{d}_k)$.

- **Backtracking line search:** An algorithm to search for a small enough stepsize for convergence, but not solving the *line search* problem exactly. For example, using Armijo condition, where we pick a stepsize α_k that satisfies

  $$f(\boldsymbol{x}_k + \alpha_k \cdot \boldsymbol{d}_k) \leq f(\boldsymbol{x}_k) + c \cdot \alpha_k \cdot \nabla f(\boldsymbol{x}_k)^\top \boldsymbol{d}_k,$$

  where $c \in (0, 0.5]$ is a user-specified parameter.

Algorithm 3: Backtracking Line Search (Armijo) at k^{th} iteration

0. User-specified parameters: $\alpha_0 > 0$, $c \in (0, 0.5]$, $\rho \in (0, 1)$

1. Set $\alpha = \alpha_0$

2. Check if $f(\boldsymbol{x}_k + \alpha \cdot \boldsymbol{d}_k) \leq f(\boldsymbol{x}_k) + c \cdot \alpha \cdot \nabla f(\boldsymbol{x}_k)^\top \boldsymbol{d}_k$

   2a. If yes, return $\alpha_k = \alpha$.

   2b. If no, set $\alpha = \rho \cdot \alpha$. Back to Step 2.

4.3 Stopping Criteria

As we don't know $f^\star = \min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x})$, when to stop the algorithm? Below are the common choices:

- When $\|\nabla f(\boldsymbol{x}_k)\| \leq \epsilon$, where $\epsilon > 0$ is a user-specified parameter. (Idea from first condition optimality condition)

- When $|f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)| \leq \epsilon$, where $\epsilon > 0$ is a user-specified parameter. That is, we stop when not making much progress.

- When $\dfrac{|f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)|}{\max\{1, |f(\boldsymbol{x}_k)|\}} \leq \epsilon$, where $\epsilon > 0$ is a user-specified parameter. Normalized version of the above condition.

- When $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\| \leq \epsilon$, where $\epsilon > 0$ is a user-specified parameter. That is, we stop when the solution is not changing much.

- When $\dfrac{\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|}{\max\{1, \|\boldsymbol{x}_k\|\}} \leq \epsilon$, where $\epsilon > 0$ is a user-specified parameter. Normalized version of the above condition.

Note: The ϵ above has different meanings in different conditions!!

5 Gradient Descent

Recall that after we found that $-\nabla f(\boldsymbol{x})$ is a descent direction at $\boldsymbol{x}$, we restrict to the following algorithm:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \cdot \boldsymbol{S}_k \nabla f(\boldsymbol{x}_k)$$

where $\boldsymbol{S}_k \in \mathbb{S}^n_{++}$, for $k \in \mathbb{Z}_+$. **Gradient descent method** is essentially choosing $\boldsymbol{S}_k = \boldsymbol{I}$ for all k, *i.e.*

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \cdot \nabla f(\boldsymbol{x}_k).$$

5.1 Gradient Descent Interpretation

Gradient descent method can be viewed from another angle via Taylor expansion:

$$f(\boldsymbol{x}) = f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^\top (\boldsymbol{x} - \boldsymbol{x}_k) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_k)^\top \nabla^2 f(\boldsymbol{x}_k)(\boldsymbol{x} - \boldsymbol{x}_k) + o(\|\boldsymbol{x} - \boldsymbol{x}_k\|_2^2)$$

We approximate the above $f(\boldsymbol{x})$ as

$$f(\boldsymbol{x}) \approx f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^\top (\boldsymbol{x} - \boldsymbol{x}_k) + \frac{1}{2\alpha_k}(\boldsymbol{x} - \boldsymbol{x}_k)^\top \boldsymbol{I}(\boldsymbol{x} - \boldsymbol{x}_k),$$

for some $\alpha_k > 0$. By minimizing the RHS over $\boldsymbol{x}$ using first order condition, we have

$$\nabla f(\boldsymbol{x}_k) + \frac{1}{\alpha_k}(\boldsymbol{x} - \boldsymbol{x}_k) = 0 \quad \iff \quad \boldsymbol{x}^\star = \boldsymbol{x}_k - \alpha_k \cdot \nabla f(\boldsymbol{x}_k)$$

The solution of the "approximate f" is the solution at next iteration using gradient descent method!

Rough idea: Different local approximations $\Rightarrow$ different algorithms!

5.2 Gradient Descent with Fixed Stepsize

Assumption 1. *The function $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable with $\boldsymbol{dom}(f) = \mathbb{R}^n$, and it has have an L-Lipschitz continuous gradient, i.e.*

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{dom}(f)$$

Note: The definition of L-Lipschitz continuous gradient applies also for non-convex function.

Lemma 4. *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable, and it has have an L-Lipschitz continuous gradient, then*

$$(\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}))^\top (\boldsymbol{x} - \boldsymbol{y}) \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2^2, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{dom}(f).$$

If $\boldsymbol{dom}(f)$ is convex, then the above holds if and only if

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{dom}(f)$$

Theorem 5. *Consider gradient descent method with fixed stepsize $\alpha \leq 1/L$, then*

$$f(\boldsymbol{x}_k) - f^\star \leq \frac{\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2}{2\alpha k},$$

where $f^\star = \min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$ and $\boldsymbol{x}^\star \in \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$.

5.3 Gradient Descent with Backtracking

Theorem 6. *Consider gradient descent method with backtracking line search (Algorithm 3) with* $\alpha_0 = 1$ *and* $c = 1/2$*, then*

$$f(\boldsymbol{x}_k) - f^\star \leq \frac{\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2}{2\hat{\alpha}k},$$

where $f^\star = \min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$*,* $\boldsymbol{x}^\star \in \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$*, and* $\hat{\alpha} = \min\{1, \rho/L\}$*.*

6 Beyond Gradient Descent

- **Newton Direction:** Choosing $\boldsymbol{S}_k = [\nabla^2 f(\boldsymbol{x}_k)]^{-1}$
- **Scaled Gradient Direction:** Choosing $\boldsymbol{S}_k = \mathrm{diag}(\boldsymbol{\beta}_k)$, where $\boldsymbol{\beta}_k \in \mathbb{R}^n_{++}$ (*e.g.* inverse of the diagonal of the Hessian)
- **Quasi-Newton Direction:** Choosing $\boldsymbol{S}_k$ as an approximate inverse of the Hessian at $\boldsymbol{x}_k$
- **Regularized Newton Direction:** Choosing $\boldsymbol{S}_k = [\nabla^2 f(\boldsymbol{x}_k) + \mu_k \boldsymbol{I}]^{-1}$ where $\mu_k > 0$