Large Scale Optimization for Machine Learning: Lecture 4

Lecturer: Meisam Razaviyayn Scribe: Sina Baharlouei

Sep 5, 2017

1 Recap and Agenda

In the previous lectures, we defined concepts of local and global optimum, convex sets and convex functions. Also, we discussed the necessary conditions for a point to be optimum, the relationship between local and global optimum for a convex function and some important properties of convex sets and convex functions.

In this lecture, we mainly focus on the "Iterative Descent" methods which are a family of Algorithms for finding local optimum of a convex function.

2 Overview

As it is mentioned in the previous lecture, if $\nabla f(x) = 0$, then x could be considered as a candidate local minimum point. Now suppose we have a point in which $\nabla f(x) \neq 0$. It cannot be a local optimum point, but how can we find another point like $x^{'}$ so that $f(x^{'}) < f(x)$ using gradient of function on the current point? Intuitively, as we move in the opposite direction of gradient, we get closer to the local minimum point (Gradient vector of a function at a certain point is perpendicular to the tangent line at that point). The following proposition gives the mathematical equivalence of this intuition:

Proposition 1. *If* $\nabla f(x)^T d < 0$ *then* $\exists \delta$ *such that* $f(\boldsymbol{x} + \alpha \boldsymbol{d}) < f(\boldsymbol{x})$ $\forall \alpha \in (0, \delta)$.

Proof. By the mean value theorem, we can write $f(\boldsymbol{x} + \alpha \boldsymbol{d}) = f(\boldsymbol{x}) + \alpha \boldsymbol{d}^T \nabla f(\boldsymbol{\xi})$ where $\boldsymbol{\xi} = \gamma \boldsymbol{x} + (1 - \gamma)(\boldsymbol{x} + \alpha \boldsymbol{d})$ for some $\gamma \in (0, 1)$. We can

choose α small enough that $\boldsymbol{\xi}$ is near $\boldsymbol{x}$ and $\boldsymbol{d}^T \nabla f(\boldsymbol{\xi}) < 0$. This gives us $f(\boldsymbol{x} + \alpha \boldsymbol{d}) < f(\boldsymbol{x})$. □

3 Iterative Descent Algorithms

Generally speaking, Iterative Descent Algorithms consist of two main components: Direction sequence, Step-size sequence. By choosing these two sequences, an iterative descent algorithm would be obtained. Also these algorithms have an update rule as follow:

$$x^{r+1} \leftarrow x^r + \alpha^r d^r \quad \forall r \in \{0, 1, ...\} \tag{1}$$

So for each Iterative Algorithm, $\{a^r\}$ which is the step-size sequence and $\{d^r\}$ which is the direction sequence, should be proposed.

3.1 Choices of direction

Each direction sequence with elements each of which has an angle more than $\frac{\pi}{2}$ with the gradient vector on that point, could be considered as a candidate sequence for direction. In this section we introduce some popular directions, widely used in iterative descent Algorithms:

3.1.1 Gradient Descent

A natural way of selecting direction sequence is to choose direction exactly in the opposite direction of gradient vector in each iteration. This choice gives us Gradient/Steepest Descent Algorithm:

$$d^r = -\nabla f(x^r) \tag{2}$$

3.1.2 Newton Direction

Based on the second order Taylor expansion, we can find a more accurate direction for converging, rather than gradient descent's direction. At $x = x'$, $f(x)$ can be approximated by:

$$f(x) \approx g(x) = f(x') + \nabla f(x')^T (x - x') + \frac{1}{2}(x - x')^T \nabla^2 f(x')(x - x')$$

It is minimized by solving $\nabla g(x) = 0$:

$$\nabla g(x) = \nabla f(x') + \nabla^2 f(x')(x - x') = 0$$

which yields to:

$$x = x' - (\nabla^2 f(x'))^{-1} \nabla f(x')$$

So for estimating x^{r+1} we can use the following formula for direction:

$$d^r = (\nabla^2 f(x^r))^{-1} \nabla f(x^r) \tag{3}$$

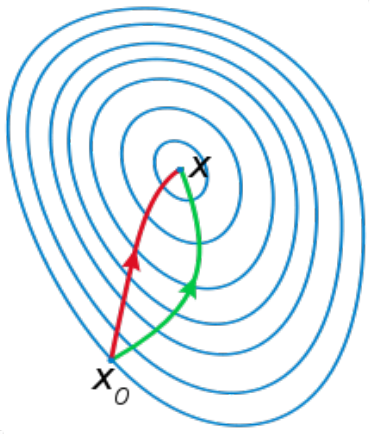

Figure 1: Newton Optimization(Red) versus Gradient Descent(Green)

3.1.3 Diagonally scaled gradient descent

Though using Newton method rather than gradient descent, reduces the number of iterations to converge, the cost of computing Newton direction is a drawback; since computation of the inverse of Hessian matrix is a time-consuming procedure. Instead, an approximation of Newton method could be used to reduce the computational complexity. In Diagonally scaled gradient descent, we use the diagonal matrices as the direction sequence.

$$d^r = -D^r \nabla f(x^r) \quad \text{for some } D^r > 0 \tag{4}$$

For instance, we can use the diagonal elements of Newton direction as the direction sequence of Algorithm. As we said, this choice reduces the

computation complexity of Newton method.

$$D^r = diag(\nabla^2 f(x^r))^{-1} \tag{5}$$

3.2 Choices of step-size

In the last section, some well-known directions for iterative descent algorithms has been proposed. In this part, we introduce some popular choices of step-size in iterative descent methods.

3.2.1 Constant step-size

A simple way for selecting step-size is to choose a constant value for step-size.

$$\alpha^r = \alpha \quad \forall r \in \{1, 2, ...\} \tag{6}$$

This method is widely used, because of its simplicity. Picking a large α might prevent Algorithm to converge; however, choosing a small α reduces the rate of convergence.

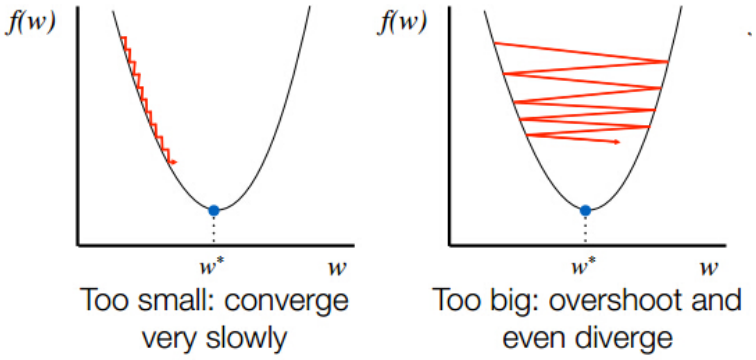

Figure 2: Constant step-size

3.2.2 Exact minimization of step-size

One intuitive approach for the choice of step-size is to find the best step-size in each iteration as follow:

$$\alpha^r \in \arg\min_{\alpha \geq 0} f(x^r + \alpha d^r) \tag{7}$$

Remark. *This method chooses the best step-size in each iteration locally. So it does not necessarily guarantee to achieve best sequence choice globally.*

3.2.3 Limited minimization of step-size

This approach is similar to the exact minimization of step-size. The only difference is that α can be chosen from a bounded set as follow:

$$\alpha^r \in \arg\min_{\alpha \in (0,\bar{\alpha}]} f(x^r + \alpha d^r) \tag{8}$$

This extra condition prevents over-optimization of the algorithm.

3.2.4 Diminishing method

Another way for choosing step-size sequence is to make step-size smaller in each step-size. This method guarantees convergence of the algorithm if is satisfies following conditions:

- $\lim_{r\to\infty} a^r = 0$
- $\sum_r \alpha^r = \infty$

For example $\{a_r\} = \frac{1}{r}$ is an acceptable choice. But $\{a_r\} = \frac{1}{r^2}$ or $\{a_r\} = \frac{1}{2^r}$ are not acceptable. Since:

$$\sum_r \frac{1}{r^2} = \frac{\pi^2}{6}$$

$$\sum_r \frac{1}{2^r} = 1$$

One important question might be asked is why the second condition is necessary? This is because without having the second condition the set of points that algorithm can be reached would be bounded. And so it might not converge to optimal point.

3.2.5 Back tracking (Armijo)

According to the first order Taylor expansion: $f(\boldsymbol{x}+\alpha\boldsymbol{d}) = f(\boldsymbol{x})+\alpha\boldsymbol{d}^T\nabla f(\boldsymbol{x})+ o(\alpha)$ which implies

$$\underbrace{-\alpha\boldsymbol{d}^T\nabla f(\boldsymbol{x})}_{\text{predicted decrease}} \approx \underbrace{f(\boldsymbol{x}) - f(\boldsymbol{x}+\alpha\boldsymbol{d})}_{\text{actual decrease}}.$$

The intuition behind Armijo method is to find maximum step-size among $\bar{\alpha},\ \bar{\alpha}\beta,\ \bar{\alpha}\beta^2\ldots$ such that the actual decrease of function will be more than a coefficient of predicted decrease. As we cannot guarantee that predicted decrease is less than actual decrease, the coefficient(σ) must be less than 1. Mathematically speaking:

Definition 1. *Let $\beta, \sigma \in (0,1)$ be constant numbers, and $\bar{\alpha}$ be initial step-size, Armijo back-tracking method sequence would be defined as follow:*

$$a^r = max\{\bar{\alpha}\beta^i | f(x^r) - f(x^r + \bar{\alpha}\beta^i d^r) \geq -\sigma\bar{\alpha}\beta^i\nabla f(x^r)^T d^r, i = 0, 1, ...\}$$

Proposition 2. *If $\nabla f(x^r)^T d^r < 0$ then a^r is well-defined.*

3.3 Convergence Analysis

In this section we introduce different aspects of analyzing the convergence of an iterative descent Algorithm. One important question about an optimization algorithm is that whether it converges to a stationary point at all? Also it is crucial to ask about the rate of convergence of an Algorithm.

3.3.1 Asymptotic rate of convergence

This measurement tends to analyze the behavior of Algorithm asymptotically. In other words, how fast the Algorithm converges to an optimum point in infinity? So first an error function would be defined. For example it could be $e(x) = ||x - x^*||$ or $e(x) = f(x) - f(x^*)$. The asymptotic behavior of function would be determined as follow:

$$\limsup_{r\to\infty} \frac{e(\boldsymbol{x}^{r+1})}{e(\boldsymbol{x}^r)} = \beta$$

There is three possibility for β:

- Sublinear($\beta = 1$): $e(x^r) = \frac{1}{r} \quad \Rightarrow \quad \limsup_{r\to\infty} \frac{e(x^{r+1})}{e(x^r)} = 1.$
- Linear($\beta \in (0,1)$): $e(x^r) = \left(\frac{1}{2}\right)^r \quad \Rightarrow \quad \limsup_{r\to\infty} \frac{e(x^{r+1})}{e(x^r)} = \frac{1}{2}$
- Superlinear($\beta = 0$): $e(x^r) = \left(\frac{1}{2}\right)^{r^2} \quad \Rightarrow \quad \limsup_{r\to\infty} \frac{e(x^{r+1})}{e(x^r)} = \left(\frac{1}{2}\right)^{2r+1} = 0$

3.3.2 Iteration complexity analysis

If an Algorithm converges faster than another one, It does not mean necessarily the number of iterations for first Algorithm is less than the second one (Because this measure only considers the rate of convergence asymptotically). Instead we can analyze the number of iterations required to achieve $\epsilon - \textit{optimal}$ solution which means $e(x^r)$ would be less than ϵ. Currently, most of the researchers analyze the worst case number of iterations for an Algorithm.

This measurement tends to analyze the behavior of Algorithm

3.3.3 Convergence to a stationary point

$f(x^{r+1}) < f(x^r)$ does not necessarily imply $\{x^r\}$ converging to a local optimal point even if your function is convex or your step size is small.
For example, let $f(x) = x^2$. If we let $x^r = (-1)^r\left(1 + \frac{1}{r}\right)$, then $f(x^r) = \left(1 + \frac{1}{r}\right)^2$. The objective value decreases as you iterate but $\{x^r\}$ does not converge to a stationary solution.

3.3.4 Gradient Related condition

The gradient related condition is satisfied if for any sub-sequence of $\{x^r\}_{r\in\kappa}$ converging to a non-stationary point, the corresponding sub-sequence of directions $\{d^r\}_{r\in\kappa}$ is bounded and $\limsup_{r\to\infty, r\in\kappa} \nabla f(x^r)^T d^r < 0$ For example:

$$d^r = -D^r \nabla f(x^r) \text{ with } \bar{\gamma} I \geq D^r \geq \gamma I > 0, \forall r$$

The second condition in the example, shows the elements of the diagonal matrix are between γ and $\bar{\gamma}$.

- **Descent Lemma**. Suppose we have a smooth function $f : \mathbb{R}^n \to \mathbb{R}$ such that $||\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})|| \leq L||\boldsymbol{x} - \boldsymbol{y}||$. Then for any $\boldsymbol{h}$, we have $f(\boldsymbol{x} + \boldsymbol{h}) \leq f(\boldsymbol{x}) + \boldsymbol{h}^T \nabla f(\boldsymbol{x}) + \frac{L}{2}||\boldsymbol{h}||^2$.

  Proof. Let $g(t) = f(\boldsymbol{x} + t\boldsymbol{h})$. Since $g(1) - g(0) = \int_0^1 g'(t)\ dt$, we have

$$\begin{aligned} f(\boldsymbol{x} + \boldsymbol{h}) - f(\boldsymbol{x}) &= \int_0^1 \boldsymbol{h}^T \nabla f(\boldsymbol{x} + t\boldsymbol{h})\ dt \\ &= \int_0^1 \boldsymbol{h}^T \nabla f(\boldsymbol{x}) dt + \int_0^1 \boldsymbol{h}^T \left(\nabla f(\boldsymbol{x} + t\boldsymbol{h}) - \nabla f(\boldsymbol{x})\right)\ dt \\ &\leq \boldsymbol{h}^T \nabla f(\boldsymbol{x}) + \int_0^1 ||\boldsymbol{h}|| \cdot ||\nabla f(\boldsymbol{x} + t\boldsymbol{h}) - \nabla f(\boldsymbol{x})||\ dt \\ &\leq \boldsymbol{h}^T \nabla f(\boldsymbol{x}) + \int_0^1 tL||\boldsymbol{h}||^2 dt \\ &= \boldsymbol{h}^T \nabla f(\boldsymbol{x}) + \frac{L}{2}||\boldsymbol{h}||^2. \end{aligned}$$

- **Theorem**
  Assume $\boldsymbol{x}^{r+1} \leftarrow \boldsymbol{x}^r + \alpha^r \boldsymbol{d}^r$, $\boldsymbol{d}^r$ is gradient related, function f has Lipschitz gradient and one of the following step size rules hold:
  (a) Diminishing: $\alpha^r \to 0$ and $\sum_r \alpha^r = \infty$
  (b) Armijo
  (c) Small enough: $0 < \epsilon \leq \alpha^r \leq \frac{(2-\epsilon)|\nabla f(\boldsymbol{x}^r)^T \boldsymbol{d}^r|}{L||\boldsymbol{d}^r||^2}$
  Then every limit point of the iterates is a stationary point, i.e.

$$\text{if } \{\boldsymbol{x}^r\}_{r \in \kappa}, \text{ then } \nabla f(\bar{\boldsymbol{x}}) = 0.$$

- Proof of part (a)
  Proof is by contradiction. Assume the contrary that $\{\boldsymbol{x}^r\}_{r \in \kappa} \to \bar{\boldsymbol{x}}$ but $\bar{\boldsymbol{x}}$ is not stationary. Therefore, by the gradient related condition, we should have $\nabla f(\boldsymbol{x}^r)^T \boldsymbol{d}^r \leq C < 0$ for large enough r. Furthermore, after

some point, the step-size α^r is small enough and by the descent lemma, we have $f(\boldsymbol{x}^{r+1}) - f(\boldsymbol{x}^r) \leq \frac{\epsilon}{2}\alpha^r \nabla f(\boldsymbol{x}^r)^T \boldsymbol{d}^r \leq \frac{\epsilon}{2}\alpha^r C \leq 0$. Summing up all inequalities, we have that

$$\sum_{r=0}^{t} f(\boldsymbol{x}^{r+1}) - f(\boldsymbol{x}^r) \leq \sum_{r=0}^{t} \frac{\epsilon}{2}\alpha^r \nabla f(\boldsymbol{x}^r)^T \boldsymbol{d}^r \leq \sum_{r=0}^{t} \frac{\epsilon}{2}\alpha^r C \leq 0.$$

$$\Rightarrow \; f(\boldsymbol{x}^{t+1}) - f(\boldsymbol{x}^0) \leq \frac{\epsilon}{2} C \sum_{r=0}^{t} \alpha^r \leq 0.$$

By letting $t \to \infty$ we have,

$$f(\bar{\boldsymbol{x}}) - f(\boldsymbol{x}^0) \leq \frac{\epsilon}{2} C \sum_{r=0}^{\infty} \alpha^r \leq 0$$

from which we conclude $\sum\limits_{r=0}^{\infty} \alpha^r$ is upper bounded by a constant which contradicts the assumption that $\sum_{r=0}^{\infty} \alpha^r = \infty$.

- Proof of part (b)
  Proof by contradiction: assume the contrary that $\{x^r\} \to \bar{x}$ and $\bar{x}$ is not a stationary point. Let us assume that $\alpha^r = \beta^{i_r}$ with β is as defined in the definition of Armijo rule. If $i_r = 0, \; \forall r \geq r_0$ then the proof is the same as the proof in part (c). Therefore, let us assume that $i_r \geq 1$ for a subsequence of iterates. By restricting to a subsequence if necessary, we can write

$$f(x^r) - f(x^r + \alpha^r \mathbf{d}^r) \geq -\sigma \alpha^r \nabla f(x^r)^T \mathbf{d}^r \geq 0, \tag{9}$$

and

$$f(x^r) - f(x^r + \frac{\alpha^r}{\beta}\mathbf{d}^r) < -\sigma \frac{\alpha^r}{\beta} \nabla f(x^r)^T \mathbf{d}^r. \tag{10}$$

Defining $p^r = \frac{d^r}{\|d^r\|}$ and $\bar{\alpha}^r = \alpha^r \frac{\|d^r\|}{\beta}$, (10) can be rewritten as

$$\frac{f(x^r) - f(x^r + \bar{\alpha}^r p^r)}{\bar{\alpha}^r} < -\sigma \nabla f(x^r)^T \mathbf{p}^r. \tag{11}$$

Since $\{p^r\}$ belongs to a compact set (the surface of unit ball), it has a limit point $\bar{p}$. On the other hand, letting $r \to \infty$ in (9) implies that $\alpha^r \mathbf{d}^r \to 0$. Taking the limit $r \to \infty$ in (11) leads to

$$-\nabla f(\bar{x})^T \bar{p} \leq -\sigma \nabla f(\bar{x})^T \bar{p}.$$

Since $\sigma < 1$, we conclude that $\nabla f(\bar{x})^T \bar{p} \geq 0$ which contradicts the gradient related condition.

- Proof of Part (c)
  Proof by contradiction: We have $f(\boldsymbol{x}^r + \alpha^r \boldsymbol{d}^r) - f(\boldsymbol{x}^r) \leq \alpha^r \nabla f(\boldsymbol{x}^r)^T \boldsymbol{d}^r + \frac{L}{2}(\alpha^r)^2 ||\boldsymbol{d}^r||^2$ from Descent Lemma. Suppose $\{\boldsymbol{x}^r\}_{r \in \kappa} \to \bar{\boldsymbol{x}}$ but $\bar{\boldsymbol{x}}$ is not stationary.

$$\begin{aligned} f(\boldsymbol{x}^r + \alpha^r \boldsymbol{d}^r) - f(\boldsymbol{x}^r) &\leq \alpha^r \nabla f(\boldsymbol{x}^r)^T \boldsymbol{d}^r + \frac{L}{2} \alpha^r ||\boldsymbol{d}^r||^2 \frac{(2-\epsilon)|\nabla f(\boldsymbol{x})^T \boldsymbol{d}^r|}{L||\boldsymbol{d}^r||^2} \\ &= \alpha^r \nabla f(\boldsymbol{x}^r)^T \boldsymbol{d}^r \left(1 - \frac{2-\epsilon}{2}\right) \\ &= \frac{\epsilon}{2} \alpha^r \nabla f(\boldsymbol{x}^r)^T \boldsymbol{d}^r \leq 0 \end{aligned} \tag{12}$$

  which implies that $f(\boldsymbol{x}^{r+1}) \leq f(\boldsymbol{x}^r)$. If $\boldsymbol{x}^r \to \bar{\boldsymbol{x}}$ then $f(\boldsymbol{x}^r) \to f(\bar{\boldsymbol{x}})$. Therefore by taking limit as $r \to \infty$ in (12), we obtain $\lim_{r \to \infty} \nabla f(\boldsymbol{x}^r)^T \boldsymbol{d}^r = 0$ which contradicts gradient related condition in slide 7.