ECE 901: Large-scale Machine Learning and Optimization **Spring 2018**

Lecture 5 — February 6

Lecturer: Dimitris Papailiopoulos *Scribe: Michael Fernandes, Gia Hong Geoffrey Lau*

Note: These lecture notes are still rough, and have only have been mildly proofread.

In the last lecture, we discussed how structure of a function helps in reducing algorithmic complexity, and we also explored in detail the Gradient Descent method.

In this lecture, we are interested in studying the convergence rate of Gradient Descent applied on strongly convex problems and its complexity. We also compare convergence rates of Gradient Descent applied to strongly convex functions, smooth functions, and non-convex smooth functions. Towards the end of the lecture, we analyze the order of complexity of Gradient Descent on the Logistic Regression function with regularization, as an example.

5.1 Analysis of Gradient Descent on Smooth and Strongly Convex Problems

Let's recall, a continuously differentiable function f is β-smooth if the gradient ∇f is β-Lipschitz, that is:

$$||\nabla f(x) - \nabla f(y)|| \leq \beta ||x - y|| \tag{5.1}$$

Also if the function f is λ strongly convex, we can prove that the Co-coercivity of gradient is given by

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq \frac{1}{\lambda} ||\nabla f(x) - \nabla f(y)||^2 \tag{5.2}$$

We will now consider functions having both strong convexity and smoothness. This should allow for drastic improvements in convergence rates. Let us consider a function f that is β-smooth and λ strongly convex on $\mathbb{R}^n$. Let us denote $\kappa = \frac{\beta}{\lambda}$ as the condition number of f. Then the Co-coercivity of the gradient is given by

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq \frac{\lambda\beta}{\beta + \lambda} ||x - y||^2 + \frac{1}{\beta + \lambda} ||\nabla f(x) - \nabla f(y)||^2 \tag{5.3}$$

Corollary 5.1. *We have that $y = x^*$ is the optimum vector. Then:*

$$\nabla f(x)^T (x - x^*) \geq c_1 ||x - x^*||^2 + c_2 ||\nabla f(x)||^2 \tag{5.4}$$

where $\nabla f(x)^T$ denotes the gradient of the function, and $(x - x^*)$ denotes how far the current point x is from the optimal x^*.

This inequality tells us that a strong correlation exists between the gradient of a function and the optimum.

Theorem 5.2. *Let f be β-smooth and λ strongly convex on $\mathbb{R}^n$. Then gradient descent with step size $\gamma{=}\frac{2}{\lambda+\beta}$ satisfies,*

$$||x_t - x^*||^2 \leq exp\Big(-\frac{2t}{\kappa}\Big)||x_0 - x^*||^2 \tag{5.5}$$

where $\kappa = \frac{\beta}{\lambda}$ is the condition number of function f. A low condition number is preferred as it implies that the function is well-conditioned.

Proof: For the purpose of the proof let us define $||x_{k+1} - x^*||{=}\Delta_{t+1}$

$$\begin{aligned}
||x_{k+1} - x^*|| &= ||x_k - \gamma\nabla f(x_k) - x*||^2 \\
\Delta_{t+1} &= \Delta_t - 2\gamma\langle\nabla f(x_k), x_k - x^*\rangle + \gamma^2||\nabla f(x_k)||^2 \\
&\leq \Delta_t - 2\gamma\Big(\frac{\lambda\beta}{\lambda+\beta}||x_t - x^*||^2 + \gamma^2||\nabla f(x_k)||^2\Big) \\
&= ||x_t - x^*|| - \frac{4\lambda\beta}{(\lambda+\beta)^2}\Delta_t \\
&= \Big(1 - \frac{4\lambda\beta}{(\lambda+\beta)^2}\Big)\Delta_t \\
&= \Big(\frac{\lambda^2 + 2\lambda\beta + \beta^2 - 4\lambda\beta}{(\lambda+\beta)^2}\Big)\Delta_t \\
&\leq \Big(\frac{\lambda-\beta}{\lambda+\beta}\Big)^2\Delta_t \qquad (5.6) \\
&= \Big(\frac{1-\beta/\lambda}{1+\beta/\lambda}\Big)^2\Delta_t \\
&= \Big(\frac{1-\kappa}{1+\kappa}\Big)^2\Delta_t \\
&\vdots \\
&\leq \Big(\frac{1-\kappa}{1+\kappa}\Big)^2 \cdot \Delta_0 \\
&= e^{2tlog(1-\frac{2}{\kappa})} \cdot \Delta_0 \\
&\leq e^{-\frac{2t}{\kappa}} \cdot \Delta_0
\end{aligned}$$

which concludes the proof and also tells us gradient descent is exponentially faster on β-smooth and λ-strongly convex functions. □

5.2 Comparison of Convergence Rates of Various Function Classes

An overview of the best-case convergence rates of some commonly-encountered functions are listed in the table below (note that $R = ||x_0 - x^*||$, and T refers to the number of iterations):

Function Class	Convergence Rate
L-Lipschitz	$\frac{RL}{\sqrt{T}}$
β-smooth and λ-strongly convex	$R^2 \exp\left(-2\frac{\lambda}{\beta}T\right)$
L-Lipschitz and λ-strongly convex	$\frac{L^2}{\lambda T}$
β-smooth	$\frac{\beta R^2}{T}$

Based on this table and the proof presented in the previous section, we conclude that the structure of a function can help in improving computational complexity. However, we should be cautious that the bounds of complexity are not always tight.

The reader is encouraged to refer to the readings associated with this lecture ([Bubeck] Theorems 3.9, 3.10, 3.12, 3.13, 3.14, 3.15) for details of the proofs for upper and lower bounds of complexity for different function structures.

5.3 Computational Complexity of the Gradient Descent Method

Our interest in how the complexity of the Gradient Descent method scales with size (n) and dimension (d) of a given dataset leads us to this analysis.

The gradient of the function (∇f) is our unit of cost, and our goal is to measure the total number of gradient evaluations in order to reach ϵ-accuracy. The total cost is then:

$$\text{Total Cost} = O([\text{\# of gradient evaluations}][\text{\# of iterations (for } \epsilon\text{-accuracy)}])$$

5.3.1 An Example: Logistic Regression with Regularization

To illustrate an example, we would like to characterize the structure of the Logistic Regression function that has a Regularization term. This would help us determine the complexity of Gradient Descent when applied to this function:

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} log(1 + e^{-y_i\langle w,x_i\rangle}) + \lambda||w||^2 \tag{5.7}$$

Let's list down the characteristics of various entities in this function:

- $log(1+e^x)$ is 1-Lipschitz and $\frac{1}{4}$-smooth
- $g_1(g_2(x))$ is L_1L_2-Lipschitz
- $\langle x, w\rangle$ is $||x||$-Lipschitz
- $g(\langle x, w\rangle)$ is $\beta||x||^2$-smooth

Therefore, the first summation term in $f(w)$ is only $\frac{1}{n}\sum_i ||x_i||$-Lipschitz and $\frac{1}{4}(\frac{1}{n}\sum_i ||x_i||^2)$-smooth. It is the regularization term that has the property of being λ-strongly convex. As a result, $f(w)$ is a β-smooth and λ-strongly convex function.

To analyze the complexity of Gradient Descent on $f(w)$, we assume the following (where d is the dimension of the dataset):

- $||x_i|| = O(\sqrt{d})$
- $||w^*|| = O(\sqrt{d})$
- $R = ||w_o - w^*|| \leq O(\sqrt{d})$ (R is the initial distance between w_0 and w^*)
- $\beta = O(d)$
- $\lambda = O(1)$ (λ is a constant that does not scale with dimension d)

In order to **achieve ϵ-error**, we require the total number of iterations T of Gradient Descent to satisfy the following condition:

$$T = (d)log(\frac{d}{\epsilon}) \tag{5.8}$$

where d refers to the number of dimensions.

The **cost per iteration** is then given by:

$$\nabla l_i(w) = l_i'(\langle x_i, w\rangle)x_i \tag{5.9}$$

where l_i' is the pointwise derivative of the loss function for datapoint i. The complexity of evaluating $(\langle x_i, w\rangle)x_i$ for a single datapoint i is $O(d)$, and since the dataset is of size n, we end up with a complexity per iteration of $O(nd)$ for one pass over our data.

Recalling the previous expression for total cost involved in achieving ϵ-accuracy, we then arrive at a complexity of $O(nd^2\log(\frac{d}{\epsilon}))$ for Gradient Descent on the regularized Logistic Regression function. We observe that while complexity is linear in the number of datapoints n, however, it is quadratic in the dimensions d!

This is undesirable as we would like complexity of our algorithms to be linear in both n and d. Therefore, this motivates us to explore other methods of performing Gradient Descent, such as Stochastic Gradient Descent, which we will see in the next lecture.