Indian Institute of Technology Kanpur
CS773 Online Learning and Optimization

Scribe: Md Enayat Ullah
Instructor: Purushottam Kar
Date: September 19, 2016

LECTURE
12

Conjugate Gradient Method

1 Introduction

In this lecture, we will look at the Conjugate Gradient method and particularly how it is useful in solving a quadratic optimization problem. We will also look at the Steepest Descent variant of gradient descent, and tabulate a brief comparison of Steepest Descent method, Conjugate Gradient method and Newtons method (which will be covered in the coming lectures).

2 Quadratic Optimization

As the name suggests, quadratic optimization problems have an objective which is quadratic in the function variable. Let $\mathbf{x} \in \mathbb{R}^n, A \in \mathbb{R}^{n\times n}$, we have

$$\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{b}^T\mathbf{x}$$
$$\text{where } A \succ 0$$

Note that this is a smooth convex unconstrained optimization problem. The optimality condition is that the gradient must vanish at the optima. Therefore,

$$\nabla f(\mathbf{x}^*) = \mathbf{0}$$
$$\implies \mathbf{x}^* = A^{-1}\mathbf{b}$$

A^{-1} is guaranteed to exist because A is p.s.d. We therefore have a closed form solution for this problem. However, the closed-form solution requires inverting an $(n \times n)$ matrix, which is computationally expensive $(O(n^3))$. We therefore, will look at efficient ways to address this. A simple example of quadratic optimization is the Linear Regression problem.

Example 12.1. Given $X \in \mathbb{R}^{n\times d}, Y \in \mathbb{R}^n$. Find $W \in \mathbb{R}^d$

$$\begin{aligned}
&\min_W \frac{1}{2}\left\|XW - Y\right\|_2^2 \\
&= \min_W \frac{1}{2}W^T X^T XW - Y^T XW + \frac{1}{2}Y^T Y \\
&= \min_W \frac{1}{2}W^T X^T XW - Y^T XW \qquad \left(\because \frac{1}{2}Y^T Y \text{ is a constant}\right)
\end{aligned}$$

Let $A = X^T X, \mathbf{b} = Y^T X$. We have

$$\min_W \frac{1}{2}W^T AW - \mathbf{b}^T W$$

which is same as the quadratic optimization problem discussed before. The above example can also be framed as a system of linear equations $XW = Y$.

3 Steepest Descent Method

Steepest Descent Method is a variant of Gradient Descent in which we make the maximum possible descent in a fixed direction such that it offers a maximum decrease in the objective function value. We therefore solve an optimization problem at each update step wherein we minimize the objective function along a given direction with respect to the step size. We now look at the method of Steepest Descent with respect to the quadratic optimization problem.

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{b}^T\mathbf{x} \\ f(\mathbf{x}) &= \frac{1}{2}\left\|\mathbf{x}\right\|_A^2 - \mathbf{b}^T\mathbf{x} \\ \nabla f(\mathbf{x}) &= A\mathbf{x} - \mathbf{b} \end{aligned}$$

We define the residual (or residual error) at time t, $\mathbf{r}^t$ as:

$$\mathbf{r}^t = \mathbf{b} - A\mathbf{x}^t$$

which is the error the iterate x^t incurs. Note that $\nabla f(\mathbf{x}^t) = -\mathbf{r}^t$.
Consider the following Gradient Descent update

$$\begin{aligned} &\mathbf{x}^{t+1} = \mathbf{x}^t + \eta(\mathbf{b} - A\mathbf{x}^t) \\ &\text{Let } \mathbf{x}(\eta) = \mathbf{x}^t + \eta(\mathbf{b} - A\mathbf{x}^t) \\ &\text{and } g(\mathbf{x}(\eta)) = f(\mathbf{x}^t + \eta(\mathbf{b} - A\mathbf{x}^t) \end{aligned}$$

Steepest Descent method essentially solves the following optimization problem:

$$\min_{\eta} \; g(\mathbf{x}\left(\eta\right))$$

Since this is a smooth convex unconstrained optimization problem, we have the optimality condition as:

$$\begin{aligned} &\nabla g(\mathbf{x}(\eta)) = 0 \\ \implies &g'(\mathbf{x}(\eta))^T\mathbf{x}'(\eta) = 0 \\ \implies &(A\mathbf{x}(\eta) - \mathbf{b})^T(\mathbf{b} - A\mathbf{x}^t) = 0 \\ \implies &\left\langle A(\mathbf{x}^t + \eta(\mathbf{b} - A\mathbf{x}^t) - \mathbf{b}, \mathbf{b} - A\mathbf{x}^t\right\rangle = 0 \\ \implies &\left\langle A(\mathbf{x}^t + \eta\mathbf{r}^t) - \mathbf{b}, \mathbf{r}^t\right\rangle = 0 && \left(\because\ \mathbf{r}^t = \mathbf{b} - A\mathbf{x}^t\right) \\ \implies &\left\langle A\mathbf{x}^t - \mathbf{b} + \eta A\mathbf{r}^t, \mathbf{r}^t\right\rangle = 0 \\ \implies &\left\langle -\mathbf{r}^t + \eta A\mathbf{r}^t, \mathbf{r}^t\right\rangle = 0 \\ \implies &-\left\|\mathbf{r}^t\right\|^2 + \eta\mathbf{r}^{t^T}A\mathbf{r}^t = 0 \\ &\therefore\ \eta = \frac{\left\|\mathbf{r}^t\right\|^2}{\mathbf{r}^{t^T}A\mathbf{r}^t} \end{aligned}$$

This is the optimal step size which guarantees maximum decrease along the direction of the gradient at every step. Also, note that

$$\begin{aligned} &g'(\mathbf{x}(\eta))^T\mathbf{x}'(\eta) = 0 \\ \implies &\underbrace{(A\mathbf{x}(\eta) - \mathbf{b})^T}_{\nabla f(\mathbf{x}^{t+1})}\underbrace{(\mathbf{b} - A\mathbf{x}^t)}_{-\nabla f(\mathbf{x}^t)} = 0 \end{aligned}$$

The gradient at the next iterate ($\nabla f(\mathbf{x}^{t+1})$) and the current direction of movement ($-\nabla f(\mathbf{x}^t)$) are orthogonal to each other (in the standard inner product). Therefore, in the Steepest Descent method, the two consecutive update directions are always orthogonal to each other.

3.1 Convergence Rate

Definition 12.1 (Condition Number)**.** The condition number (κ) of a quadratic optimization problem is defined as:

$$\kappa = \kappa(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$$

where $\lambda_{max}(A)$ and $\lambda_{min}(A)$ are the maximum and minimum eigenvalues of the matrix A respectively.

Steepest Descent method offers a linear rate of convergence. The objective function value decreases as per the following geometric progression:

$$f(\mathbf{x}^t) \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^t f(\mathbf{x}^0)$$

Note that this is very similar to the linear rate of convergence we covered while discussing Gradient Descent for strongly-convex and strongly-smooth objectives. We had:

$$f(\mathbf{x}^t) \leq \left(1 - \frac{\alpha}{L}\right)^t f(\mathbf{x}^0)$$

where α and L are the strong-convexity and strong-smoothness parameters of the objective respectively, i.e. $\alpha \leq \left\|\nabla^2 f(\mathbf{x})\right\|_2 \leq L$. In the quadratic optimization problem, $\nabla^2 f(\mathbf{x}) = A$. Therefore, α and L are the smallest and largest eigenvalues of A. We have, $\kappa = \dfrac{L}{\alpha}$ and

$$f(\mathbf{x}^t) \leq \left(1 - \frac{1}{\kappa}\right)^t f(\mathbf{x}^0)$$
$$\implies f(\mathbf{x}^t) \leq \left(\frac{\kappa - 1}{\kappa}\right)^t f(\mathbf{x}^0)$$

In general, this is the rate of convergence Gradient Descent offers in quadratic optimization. The improvement in the constant is particularly an advantage of the Steepest Descent method. Moreover, depending on the value of κ, optimization problems can be classified as follows:

$$\kappa = \begin{cases} \text{small :} & \text{Well-conditioned problem} \\ \text{large :} & \text{Ill-conditioned problem} \\ \infty : & \text{Singular} \end{cases}$$

Well-conditioned problems are easy to solve and they converge quickly, whereas ill-conditioned problems usually require a prior pre-conditioning to reduce it into a well-conditioned problem. Singular problems do not have a unique solution.

4 Conjugate Gradient Method

In the Steepest Descent method, we saw that the consecutive direction of updates are orthogonal to each other (in the standard inner product sense). Conjugate Gradient method hinges on the intuition that it more fitting to consider orthogonality with respect to the inner product induced by A, because it will exploit the geometry of A and weed out the redundant updates. We therefore, will see that the subsequent directions of updates in Conjugate Gradient method are orthogonal to each other with respect to the induced inner product. Consider the following problem with regard to finding pairwise conjugate directions with respect to the induced inner product:

Given $A \in \mathbb{R}^{n\times n}$ such that $A \succ 0$
Find $\left\{\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^n\right\}, \mathbf{p}^i \in \mathbb{R}^n$
such that span $\left\{\mathbf{p}^i\right\} = \mathbb{R}^n$ and $\left\langle \mathbf{p}^i, \mathbf{p}^j \right\rangle_A = \mathbf{p}^{i^T} A \mathbf{p}^j = 0 \; \forall \; i \neq j$
Such $\mathbf{p}^i$'s are known as pairwise conjugate directions.

One solution to the above problem are the eigenvectors $\{\mathbf{e}_i\}$'s of A. Since the eigenvectors are linearly independent, the n eigenvectors form a basis for $\mathbb{R}^n$ which spans the space $\mathbb{R}^n$. Also,

$$\begin{aligned}
A &= \sum_{i=1}^{n} \lambda_i \mathbf{e}_i \mathbf{e}_i^T \\
\therefore \; \mathbf{e}_i^T A \mathbf{e}_j &= e_i^T \sum_{i=1}^{n} \lambda_i \mathbf{e}_i \mathbf{e}_i^T \mathbf{e}_j \\
&= \mathbf{e}_i^T \lambda_j \mathbf{e}_j \mathbf{e}_j^T \mathbf{e}_j && \left(\because \; \mathbf{e}_i^T \mathbf{e}_j = 0 \; \forall \; i \neq j \right) \\
&= \lambda_j \mathbf{e}_i^T \mathbf{e}_j = 0
\end{aligned}$$

Note that we do not necessarily require eigenvectors in the Conjugate Gradient method. Any set of pairwise conjugate directions will work.
Since span $\left\{\mathbf{p}^i\right\} = \mathbb{R}^n$, the optimal solution $(x^* = A^{-1}b)$ can be represented as a linear combination of $\mathbf{p}^i$'s. We can therefore estimate $\alpha_i (\in \mathbb{R})$'s such that

$$\mathbf{x}^* = \sum_{i=1}^{n} \alpha_i \mathbf{p}^i \tag{1}$$

We use Gram-Schmidt process with a slight modification to estimate α_i's. The modification is that instead of considering orthogonality with respect to the standard inner-product, we use the inner-product induced by A. From Eq. 1, we have

$$\begin{aligned}
\left\langle \mathbf{x}^*, \mathbf{p}^i \right\rangle_A &= \left\langle \sum_{i=1}^{n} \alpha_i \mathbf{p}^i, \mathbf{p}^i \right\rangle_A \\
\implies \left\langle \mathbf{x}^*, A\mathbf{p}^i \right\rangle &= \left\langle \sum_{i=1}^{n} \alpha_i \mathbf{p}^i, A\mathbf{p}^i \right\rangle \\
\implies \mathbf{b}\mathbf{p}^i &= \alpha_i \mathbf{p}^{i^T} A \mathbf{p}^i && (\because \; A\mathbf{x}^* = \mathbf{b}) \\
\therefore \alpha_i &= \frac{\mathbf{p}^{i^T}\mathbf{b}}{\mathbf{p}^{i^T}\mathbf{A}\mathbf{p}^i}
\end{aligned}$$

Algorithm 1 presents the generic pseudo-code of Conjugate Gradient method. Step-length η is calculated in the same way as in Steepest Descent Method.

```
Algorithm 1: Conjugate Gradient Method

Input: $A \succ 0, \mathbf{b}$
$\mathbf{x}^0 = \mathbf{0}$
$\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0$
$\mathbf{p}^0 = \mathbf{r}^0$
for $t = 1, 2, \ldots, T-1$ do
    $\mathbf{x}^t = \arg\min_{\eta} f(\mathbf{x}^{t-1} - \eta \mathbf{p}^{t-1})$
    $\mathbf{r}^t = \mathbf{b} - A\mathbf{x}^t$
    $\mathbf{p}^t = \mathbf{r}^t - \sum_{j<t} \frac{\mathbf{r}^{t^T}\mathbf{p}^j}{\langle \mathbf{p}^j, \mathbf{p}^j \rangle}\mathbf{p}^j$
end for
```

The above implementation is simple, but is wasteful as it makes the same computations repeatedly. Algorithm 2 below is a more efficient implementation of Conjugate Gradient method.

```
Algorithm 2: Conjugate Gradient Method

Input: $A \succ 0, \mathbf{b}$
$\mathbf{x}^0 = \mathbf{0}$
$\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0$
$\mathbf{p}^0 = \mathbf{r}^0$
for $t = 1, 2, \ldots, T-1$ do
    $\alpha^t = \frac{\|\mathbf{r}^t\|_2^2}{\langle \mathbf{p}^t, \mathbf{p}^t \rangle_A}$
    $\mathbf{x}^{t+1} = \mathbf{x}^t + \alpha^t \mathbf{p}^t$
    $\mathbf{r}^{t+1} = \mathbf{r}^t - \alpha^t A\mathbf{p}^t$
    $\beta^t = \frac{\|\mathbf{r}^{t+1}\|_2^2}{\|\mathbf{r}^t\|_2^2}$
end for
```

In the coming lecture, we will look at more properties of Conjugate Gradient method, and in particular we will prove that Conjugate Gradient method converges in less than n steps, or if the matrix A has m distinct eigenvalues, it takes m steps to get to the true solution.

5 Experiments

Consider the quadratic optimization problem with $A = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} 10 \\ 10 \end{bmatrix}$.

Initial point $\mathbf{x}^0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, Optima $\mathbf{x}^* = \begin{bmatrix} 10 \\ 1 \end{bmatrix}$.

Contour plots for Steepest descent and and Conjugate Gradient method are shown in Figure 1 and Figure 2 respectively. Steepest Descent takes multiple iterations to reach the optima, whereas Conjugate Gradient converges in just one step.

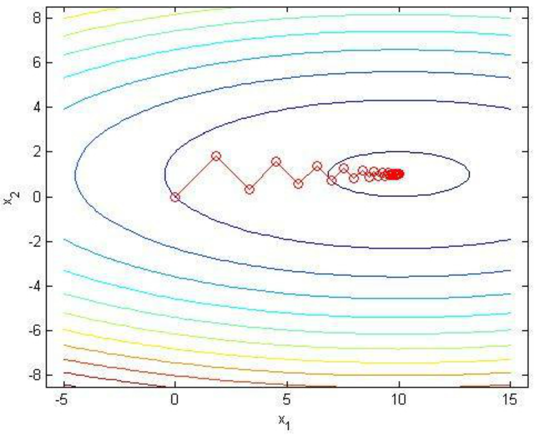

Figure 1: Steepest Descent Method

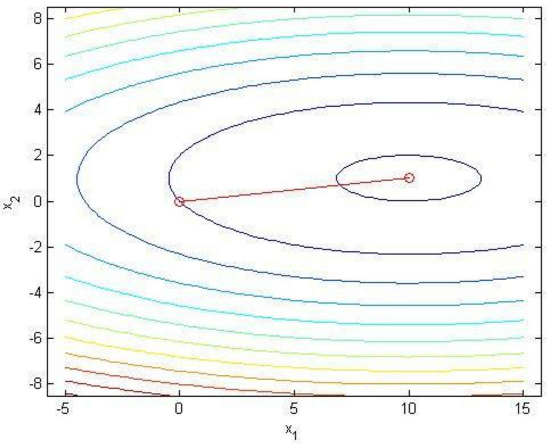

Figure 2: Conjugate Gradient Method

5.1 Extensions

- What if A is non-symmetric?
  Bi-conjugate Gradient method is a generalized version of Conjugate Gradient which works even when A is not symmetric. It essentially solves the problem $A^T\mathbf{x} = \mathbf{b}$ along with $A\mathbf{x} = \mathbf{b}$. (Wikipedia (2016))

- What if A is ill-conditioned?
  A pre-conditioning is performed, wherein we multiply a pre-conditioner M^{-1} to $(A\mathbf{x} - \mathbf{b})$, such that $\kappa\left(M^{-1}A\right)$ is small. The problem then reduces to

$$\underbrace{\left(M^{-1}A\right)}_{A'}\mathbf{x} - \underbrace{\left(M^{-1}\mathbf{b}\right)}_{\mathbf{b}'}$$

  A^{-1} is an ideal pre-conditioner, as it gives an identity matrix, and Conjugate Gradient method only takes 1 step to converge, but again if we can compute A^{-1}, we get the solution directly. A practical choice of pre-conditioner is $M = \hat{A}^{-1}$, where $\hat{A} = LL^T$ is an approximation of A using the Cholesky decomposition.

- Non-linear Optimization:
  In general, Conjugate Gradient method can be used to solve non-linear optimization.

$$\min_{\mathbf{x}} f(\mathbf{x})$$

  where the objective f may be non-linear in $\mathbf{x}$. Conjugate Gradient is essentially a family of methods, depending how we calculate α^t 's and β^t 's. Algorithm 2, wherein $\beta^t = \dfrac{\|\mathbf{r}^{t+1}\|_2^2}{\|\mathbf{r}^t\|_2^2}$ is known as Fletcher-Reeves method, which is used in non-linear optimization as well (Shewchuk (1994)).

6 Conclusion

Table 1 summarizes a brief comparison between Steepest Descent method, Conjugate Gradient method and Newtons method. In the coming lectures, we will see how the entries of the table, in particular to Newtons method, are obtained. Note that the entries are with respect to the quadratic optimization formulation of Linear Regression wherein we have n samples of d dimensional vectors. A trade-off between computational complexity and convergence rate is very apparent. For example: Newtons method converges in 1 step, but requires $O(d^3)$ computations, whereas Steepest Descent method is computationally cheap, but gives a relatively worse convergence guarantee.

Method	Run-time (per iteration)	Convergence (Quadratic Opt)	Convergence (General)
Steepest Descent (GD)	$O(nd)$	$\left(\frac{\kappa-1}{\kappa+1}\right)^t$	$\left(\frac{\kappa-1}{\kappa}\right)^t$
Conjugate Gradient (CG)	$O(nd)$ or $O(n^2)$	$\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^t$ or d	-
Newtons Method	$O(d^3)$	1	$\left(\frac{1}{2}\right)^{2^t}$

Table 1: Steepest Descent vs. Conjugate Gradient vs Newtons Method

References

Jonathan Richard Shewchuk. An introduction to the conjugate gradient method without the agonizing pain, 1994.

Wikipedia. Biconjugate gradient method — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Biconjugate%20gradient%20method&oldid=710819936`, 2016.