Beyond Gradient Descent: Quadratic Objective Function

1 Quadratic Optimization Problem

In this lecture, we consider the following quadratic optimization problem

$$\min_x Q(x), \qquad Q(x) = \frac{1}{2}x^\top Ax - b^\top x, \tag{1}$$

In this problem, we assume that A is a $d \times d$ symmetric positive definite matrix. b and x are d dimensional vectors.

This is a strongly-convex optimization problem. Let $\lambda > 0$ be the smallest eigenvalue of A, and L be the largest eigenvalue of A, then λ is the strong-convexity parameter of $Q(x)$, and L is the smoothness parameter of $Q(x)$.

In machine learning, the ridge regression problem below can be regarded as quadratic optimization:

$$\min_w \left[\frac{1}{n}\sum_{i=1}^n (x_i^\top w - y_i)^2 + \lambda w^\top w\right],$$

where it can be written as

$$\min_w \left[\frac{1}{2}w^\top Aw - b^\top w\right],$$

with

$$A = \frac{1}{n}\sum_{i=1}^n x_i x_i^\top + \lambda I, \quad b = \frac{1}{n}\sum_{i=1}^n x_i y_i.$$

2 Gradient Descent

The solution of (1) is given by the linear equation

$$x_* = A^{-1}b.$$

The system can be solved by gradient descent, with gradient

$$\nabla_x Q(x) = Ax - b.$$

Therefore we can apply the gradient descent method, which leads to the following iterative algorithm:

$$x_t = x_{t-1} - \eta(Ax_{t-1} - b).$$

It can be checked that for this problem, we have

$$x_t - x_* = (I - \eta A)(x_{t-1} - x_*).$$

Therefore

$$x_t - x_* = (I - \eta A)^t (x_0 - x_*).$$

The solution converges only when the eigenvalue of $(I - \eta A)$ belongs to $(-1, 1)$, and let ρ be the largest eigenvalue of $(I - \eta A)$ in absolute value, then we have a convergence rate of ρ:

$$\|x_t - x_*\|_2 \leq \rho^t \|x_0 - x_*\|_2.$$

If we let $\eta = 1/L$, then $\rho = 1 - \lambda/L = 1 - 1/\kappa$, where $\kappa = L/\lambda$ is the condition number. The optimal choice is at $\eta = 2/(\lambda + L)$, and the corresponding optimal convergence rate is

$$\rho = \frac{L - \lambda}{L + \lambda} = \frac{\kappa - 1}{\kappa + 1},$$

which is a more refined result than what's given in the last lecture. However, the convergence behavior is similar when $\kappa \gg 1$.

3 Conjugate Gradient Method

In numerical linear algebra, the classical iterative method of choice for solving (1) is not gradient descent, but conjugate gradient (CG). The method can be written in the following form:

$$\begin{aligned} x_t =& x_{t-1} + \alpha_{t-1} p_{t-1} \\ p_t =& -\nabla Q(x_t) + \beta_{t-1} p_{t-1}, \end{aligned} \tag{2}$$

where α_{t-1} is chosen to minimizes the objective value

$$\alpha_{t-1} = \arg\min_{\alpha} Q(x_{t-1} + \alpha p_{t-1}),$$

and β_{t-1} is chosen so that

$$(r_t + \beta_{t-1} p_{t-1})^\top A p_{t-1} = p_t^\top A p_{t-1} = 0.$$

A standard implementation can be found in Algorithm 1, where α_t and β_t are computed using algebraically equivalent formulas. Details can be found in [1].

Algorithm 1: Conjugate Gradient

Input: A, b, and x_0
Output: x_T
Let $r_0 = b - Ax_0$
Let $p_0 = r_0$
for $t = 1, 2, \ldots, T$ **do**
  Let $\alpha_{t-1} = r_{t-1}^\top r_{t-1} / p_{t-1}^\top A p_{t-1}$
  Let $x_t = x_{t-1} + \alpha_{t-1} p_{t-1}$
  Let $r_t = r_{t-1} - \alpha_{t-1} A p_{t-1}$
  Let $\beta_{t-1} = r_t^\top r_t / r_{t-1}^\top r_{t-1}$
  Let $p_t = r_t + \beta_{t-1} p_{t-1}$
Return: x_T

We want to discuss some properties of this algorithm. It can be shown that

$$r_t = b - Ax_t = -\nabla Q(x_t).$$

The update direction of x_t is not along the gradient direction r_t but the aggregated gradient direction:

$$p_t = -\nabla Q(x_t) + \beta_{t-1} p_{t-1}.$$

By expanding p_{t-1} recursively, we obtain

$$p_t = -\nabla Q(x_t) - \beta_{t-1}\nabla Q(x_{t-1}) - \beta_{t-1}\beta_{t-2}\nabla Q(x_{t-2}) - \cdots$$

This means that the search direction p_t is a weighted average of historical gradients as well as the current gradient.

The coefficient α_t is like learning rate long the search direction p_t, and CG chooses α_t to achieve the steepest descent along that direction. The parameter β_t is a shrinkage factor for aggregating the historical gradient $\nabla Q(x)$, and CG chooses it to achieve the A-orthogonality among the search directions p_t. These A-orthogonal directions are called conjugate directions, which leads to the naming of conjugate gradient. Both α_t and β_t are automatically computed in CG.

Theorem 1 *We have the following convergence result for CG:*

$$\|x_t - x_*\|_A \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^t \|x_0 - x_*\|_A,$$

where κ is the condition number of (1).

We note that

$$\frac{1}{2}\|x_t - x_*\|_A^2 = Q(x_t) - Q(x_*).$$

Therefore in order to achieve an ϵ primal sub-optimality, we need only

$$O(\sqrt{\kappa}\log(1/\epsilon))$$

number of steps using CG, as opposed to

$$O(\kappa\log(1/\epsilon))$$

when we use gradient descent. This can be significantly better when κ is large. Moreover, it can be shown that CG converges to the optimal solution in at most d steps.

Conjugate gradient is an "optimal" first order method that employs first order gradient information, and converges faster than regular gradient descent. It also automatically sets the parameters α_t and β_t, which is an advantage for general linear systems. However, for machine learning, which has a finite sum structure over data, such as ridge regression, we often need to use stochastic versions of first order methods. In this case, it is difficult to generalize the automatic computation of α_t and β_t as in CG. Therefore we have to rely on a more general method that allows fixed ways to set α_t and β_t. Another disadvantage of CG is that it is difficult to generalize to nonsmooth regularization problems such as Lasso.

4 Heavy Ball Method

The *Heavy Ball Method*, by Polyak [2], is very similar to CG, and can be stated in the same recursive equation (2), used by the conjugate gradient algorithm. However, instead of the CG method, which sets α_t and β_t automatically, in the heavy-ball method, we choose constant $\alpha_t = \alpha \geq 0$ and $\beta_t = \beta \in [0, 1)$.

It can be verified that by eliminating p_t, the heavy-ball method with constant α and β in (2) can be rewritten as

$$x_t = x_{t-1} - \alpha \nabla Q(x_{t-1}) + \beta(x_{t-1} - x_{t-2}).$$

This formulation can be regarded as a discretization of the following differential equation

$$\frac{d^2x(t)}{dt^2} = -\alpha \nabla f(x(t)) - (1-\beta)\frac{dx(t)}{dt},$$

which describes the motion of a body ("the heavy ball") in a potential field under the force of friction. Because of energy loss caused by friction, the body eventually reaches a minimum point of the potential $f(x)$. Therefore this heavy ball motion solves the potential energy minimization problem.

In the following, we show that improved convergence over the standard gradient descent can be obtained by choosing appropriate β.

Theorem 2 *We have the following convergence result for the heavy-ball method. Consider $\alpha > 0$ and $\beta \in [0, 1]$ such that*

$$\beta \geq \max((1 - \sqrt{\alpha L})^2, (1 - \sqrt{\alpha \lambda})^2),$$

then there exists a constant $C(\alpha, \beta)$ that depends on α and β such that

$$\overline{\lim_{t \to \infty}} \|x_t - x_*\|_2^{2/t} \leq \beta.$$

Proof We have the following recursive equation:

$$x_t = x_{t-1} - \alpha A(x_{t-1} - x_*) + \beta(x_{t-1} - x_{t-2}).$$

That is,

$$x_t - x_* = [(1+\beta)I - \alpha A](x_{t-1} - x_*) - \beta I(x_{t-2} - x_*).$$

This means

$$\begin{bmatrix} x_t - x_* \\ x_{t-1} - x_* \end{bmatrix} = M \begin{bmatrix} x_{t-1} - x_* \\ x_{t-2} - x_* \end{bmatrix},$$

where

$$M = \begin{bmatrix} (1+\beta)I - \alpha A & -\beta I \\ I & 0 \end{bmatrix}.$$

Therefore if we let

$$z_t = \begin{bmatrix} x_{t+1} - x_* \\ x_t - x_* \end{bmatrix},$$

then we have

$$z_t = Mz_{t-1} = \cdots M^t z_0.$$

The theorem follows if we can show that the spectral radius (large absolute eigenvalue) of M is $\sqrt{\beta}$.

Next we will show that all eigenvalues of M have magnitude no more than $\sqrt{\beta}$.

First, let U be the orthogonal matrix such that $U^\top AU$ is a diagonal matrix Λ, and let

$$V = \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix},$$

then

$$V^\top MV = \begin{bmatrix} (1+\beta)I - \alpha\Lambda & -\beta I \\ I & 0 \end{bmatrix}.$$

This matrix can be rearranged with diagonal 2×2 matrix blocks of the form

$$M_2(\gamma) = \begin{bmatrix} 1+\beta-\alpha\gamma & -\beta \\ 1 & 0 \end{bmatrix},$$

with γ being an eigenvalue of A. Therefore the eigenvalues of M are the eigenvalues of $M_2(\gamma)$ with γ being an eigenvalue of A. It is easy to check that eigenvalues of $M(\gamma)$ are

$$\begin{aligned} \lambda_1 =& \frac{1}{2}\left(-\sqrt{(\alpha\gamma-\beta-1)^2-4\beta} - \alpha\gamma + \beta + 1\right), \\ \lambda_2 =& \frac{1}{2}\left(\sqrt{(\alpha\gamma-\beta-1)^2-4\beta} - \alpha\gamma + \beta + 1\right). \end{aligned}$$

The condition of the theorem implies that for $\gamma \in [\lambda, L]$, we have

$$\beta \geq (1-\sqrt{\alpha\gamma})^2.$$

This means that $\sqrt{(\alpha\gamma-\beta-1)^2-4\beta}$ is an imaginary number, and $|\lambda_1| = \sqrt{\beta}$ and $|\lambda_2| = \sqrt{\beta}$. It follows that the spectral radius of M is $\sqrt{\beta}$. ■

In Theorem 2, if we take $\sqrt{\alpha} = 1/(\sqrt{\lambda}+\sqrt{L})$, then we can take

$$\beta = \frac{\sqrt{L}-\sqrt{\lambda}}{\sqrt{L}+\sqrt{\lambda}} = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}.$$

Therefore the convergence rate of the heavy-ball method with the optimal settings of α and β matches that of the CG algorithm in Theorem 1.

Although the heavy ball method was stated for general nonlinear optimization by Polyak, only asymptotic convergence can be proved. Global convergence can be obtained only for quadratic functions, as in Theorem 2. The method is closely related to the Chebyshev iteration method for solving linear systems, which employs the same recursive relationship (2), but using Chebyshev polynomial to set weights α_t and β_t [1]. With optimal settings, the convergence result for the Chebyshev iteration method is the same as Theorem 1. In fact, the result of CG is obtained using Chebyshev iterations, with the fact that CG achieves the optimal primal suboptimality among all methods of the form (2).

This lecture shows that methods based on (2) are first order methods that can significantly accelerate the convergence of vanilla gradient descent. These methods are extended by Nesterov for general convex functions, with global convergence proof. In the next lecture, we will study Nesterov's acceleration method.

References

[1] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[2] Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Comp. Math. Math. Phys.*, 4(5):1–17, 1964.