Chapter 4

Coordinate Descent Methods

Some of the earliest approaches proposed for multivariable optimization took one variable at a time, and did some sort of approximate minimization with respect to that variable. These approaches have a certain intuitive appeal — they replace the difficult problem of minimizing with respect to many variables with a sequence of simpler problems, each of which involves minimizing with respect to a single variable. There are many variants of this basic approach, that have gone in and out of style over the years. Nowadays there is considerable interest, driven largely by the usefulness of these methods in data analysis problems.

We again develop most of the ideas with reference to the familiar smooth convex minimization problem defined by

$$\min_{x \in \mathbb{R}^n} f(x), \tag{4.1}$$

where f is smooth and convex with modulus of convexity μ and a bound L on the Lipschitz constant of the gradient for all points x in some region of interest, that is,

$$\frac{\mu}{2}\|y - x\|^2 \leq f(y) - f(x) - \nabla f(x)^T(y - x),$$

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|,$$

for all x, y in the region of interest. We showed in Lemmas 1.12 and 1.13 that, in the case of f twice continuously differentiable, these conditions are a consequence of uniform bounds on the eigenvalues of the Hessian, namely,

$$\mu I \preceq \nabla^2 f(x) \preceq LI. \tag{4.2}$$

In Section 4.1, we outline some of the variants on the coordinate descent theme. Section 4.2 focuses on stochastic coordinate descent, which admits a simple analysis of convergence in expectation, closely related to that obtained from stochastic gradient methods (Chapter 3).

4.1 Variants

The basic "calculus" of these schemes. The coordinate-wise subproblems need to be easy to solve. In particular, it must be cheap to evaluate individial components of the gradient. If finite diff is used to do gradient evalaution, this is true.

Block coordinate descent.

Various deterministic schemes. Mention Hooke and Jeeves - interpolating after a cycle to get a coordinate direction. Like taking an occasional gradient step — but the gradient is built up piecewise rather than being fully evaluated at a particular step.

Extensions to box constraints, ℓ_1 regularizer.

4.2 Stochastic Coordinate Descent

Here we analyze the basic stochastic coordinate descent (SCD) approach for minimization of a convex function. At each iteration, we select the coordinate to update from $\{1, 2, \ldots, n\}$ uniformly at random, and take a short step along the negative partial gradient for that coordinate. We show that for strongly convex functions ($\mu > 0$), the sequence of function values $\{f(x_k)\}$ approaches the optimal value $f(x^*)$ at a linear rate, and we dicuss how this rate relates to the rates obtained in Section 2.1.2 for steepest descent methods on strongly convex functions.

We assume that the problem has form (4.1), and in fact that f is twice continuously differentiable, with μ and L satisfying (4.2), with $\mu > 0$. We introduce another constant $L_{\max}$ defined to be the bound on all diagonal elements of $\nabla^2 f(x)$, for all x in the domain of interest. [1] That is,

$$L_{\max} \geq \max_{i=1,2,\ldots,n} [\nabla^2 f(x)]_{ii}, \quad \text{for all } x.$$

Iteration k of the basic SCD approach proceeds as follows: We choose index i_k from $\{1, 2, \ldots, n\}$ independently at random, then set

$$x_{k+1} = x_k - \alpha [\nabla f(x_k)]_{i_k} e_{i_k}, \tag{4.3}$$

for some steplength $\alpha > 0$ to be discussed below. From Taylor's theorem (specifically, (1.12)), we have

$$\begin{aligned} f(x_{k+1}) &= f(x_k) - \alpha [\nabla f(x_k)]_{i_k} \nabla f(x_k)^T e_{i_k} + \frac{1}{2}\alpha^2 [\nabla f(x_k)]_{i_k}^2 e_{i_k}^T \nabla^2 f(x_k - \gamma_{i_k} [\nabla f(x_k)]_{i_k} e_{i_k}) e_{i_k} \\ &\leq f(x_k) - \alpha [\nabla f(x_k)]_{i_k}^2 + \frac{1}{2}\alpha^2 L_{\max} [\nabla f(x_k)]_{i_k}^2 \\ &\leq f(x_k) - \left(\alpha - \frac{1}{2}\alpha^2 L_{\max}\right) [\nabla f(x_k)]_{i_k}^2. \end{aligned} \tag{4.4}$$

If we take the expectation of both sides of this inequality with respect to i_k, and note that x_k is independent of i_k, we have

$$\begin{aligned} \mathbb{E}_{i_k}(f(x_{k+1})) &\leq f(x_k) - \left(\alpha - \frac{1}{2}\alpha^2 L_{\max}\right) \mathbb{E}_{i_k}\left([\nabla f(x_k)]_{i_k}^2\right) \\ &= f(x_k) - \left(\alpha - \frac{1}{2}\alpha^2 L_{\max}\right) \frac{1}{n} \sum_{i=1}^{n} [\nabla f(x_k)]_i^2 \\ &= f(x_k) - \frac{1}{n}\left(\alpha - \frac{1}{2}\alpha^2 L_{\max}\right) \|\nabla f(x_k)\|^2. \end{aligned} \tag{4.5}$$

We have from Lemma 1.13, by setting $x = x_k$ and $y = x^*$ in (1.20), that

$$\|\nabla f(x_k)\|^2 \geq 2\mu (f(x_k) - f(x^*)).$$

[1] **SJW:** *mention here and earlier the level set* $\{x \mid f(x) \leq f(x_0)\}$

By subtracting $f(x^*)$ from both sides of (4.5), and using this inequality, we obtain

$$\begin{aligned}\mathbb{E}_{i_k}(f(x_{k+1})) - f(x^*) &\leq f(x_k) - f(x^*) - \frac{2\mu}{n}\left(\alpha - \frac{1}{2}\alpha^2 L_{\max}\right)(f(x_k) - f(x^*)) \\ &= \left[1 - \frac{2\mu}{n}\left(\alpha - \frac{1}{2}\alpha^2 L_{\max}\right)\right](f(x_k) - f(x^*)). \end{aligned} \tag{4.6}$$

We now take expectations of both sides of this bound with respect to *all* random indices used during the algorithm, an operation that we denote simply by $\mathbb{E}$. [2] We obtain

$$\mathbb{E}[\mathbb{E}_{i_k}(f(x_{k+1}) - f(x^*)] = \mathbb{E}[f(x_{k+1}) - f(x^*)] \leq \left[1 - \frac{2\mu}{n}\left(\alpha - \frac{1}{2}\alpha^2 L_{\max}\right)\right]\mathbb{E}[f(x_k) - f(x^*)].$$

We conclude that the sequence of function values $\{f(x_k)\}$ converges linearly in expectation. More specifically, the sequence $\{\mathbb{E}(f(x_k) - f(x^*))\}$ converges at a Q-linear rate to zero, with

$$\frac{\mathbb{E}(f(x_{k+1}) - f(x^*))}{\mathbb{E}(f(x_k) - f(x^*))} \leq \left[1 - \frac{2\mu}{n}\left(\alpha - \frac{1}{2}\alpha^2 L_{\max}\right)\right]. \tag{4.7}$$

As in Chapter 2, we can make the linear rate bound as fast as possible by choosing α to maximize $\alpha - \alpha^2 L_{\max}/2$, that is,

$$\alpha = \frac{1}{L_{\max}}. \tag{4.8}$$

(In fact, any choice of α in the range $(0, 2/L_{\max})$ would yield a Q-linear rate in (4.7).) For the choice (4.8), we have from (4.7) that

$$\frac{\mathbb{E}(f(x_{k+1}) - f(x^*))}{\mathbb{E}(f(x_k) - f(x^*))} \leq \left(1 - \frac{\mu}{nL_{\max}}\right). \tag{4.9}$$

It in instructive to compare this rate with the one obtained for a short-step *full-gradient* steepest descent method applied to a strongly convex f. We showed in Section 2.1.2 that

$$\|x_{k+1} - x^*\| \leq \left(1 - \frac{2}{(L/\mu) + 1}\right)\|x_k - x^*\|. \tag{4.10}$$

Because of Lemma 1.12, the quantities $f(x_k) - f(x^*)$ and $\|x_k - x^*\|^2$ converge at similar rates, so we get a more apt cnparison with (4.9) by squaring both sides of (4.10). By using the approximation $(1-\epsilon)^m \approx 1 - m\epsilon$ for any constants m and ϵ with $m\epsilon \ll 1$, we estimate that the rate constant for convergence of $\{f(x_k)\}$ in short-step steepest descent would be about

$$1 - \frac{4\mu}{L + \mu} \approx 1 - \frac{4\mu}{L}, \tag{4.11}$$

where we can neglect μ for all but the most well conditioned problems.

The main differences between (4.11) and (4.9) are (a) an extra factor of 4 in the numerator of (4.11); (b) an extra factor n in the denominator in (4.9); (c) $L_{\max}$ in (4.9) is replaced by L

[2] Note that neither side of this expression depends on $i_{k+1}, i_{k+2}, \ldots$ since these are used only to generate later iterations (from x_{k+2} onwards). Additionally, neither side depends on i_k, since we have already taken the expectation on the left with respect to this random variable, and since i_k is not used for x_k or any prior iterates on the right. Hence, the full expectation operator in (4.6) corresponds to taking expectations of both sides with respect to $i_0, i_1, \ldots, i_{k-1}$.

in (4.11). The difference (b) can be accounted for by the fact that each iteration of stochastic coordinate descent is generally much cheaper than an iteration of steepest descent, because we need only one element of the gradient rather than the full gradient. When the cost to evaluate a single element is $1/n$ of the cost of the full gradient (the best case for coordnate descent, in some sense), we could say that one step of steepest descent costs about as much as n steps of coordinate descent, and that the total reduction in $\mathbb{E}(f(x) - f(x^*))$ over n steps can be estimated from (4.9) as follows:

$$\left(1 - \frac{\mu}{nL_{\max}}\right)^n \approx 1 - \frac{\mu}{L_{\max}}.$$

The difference (c) is more difficult to analyze, though we can get some bounds on the differences between L and $L_{\max}$. Consider for simplicity the convex quadratic function $f(x) = (1/2)x^T Ax$ where A is positive definite. We have as in Section 2.1.3 that $L = \|A\|_2 = \lambda_{\max}(A)$, while from the definition of $L_{\max}$ we have $L_{\max} = \max_{i=1,2,\ldots,n} A_{ii}$. It is clear from definition of matrix norm that

$$L \geq \|Ae_i\|/\|e_i\| = \sqrt{\sum_{j=1}^{n} A_{ji}^2} \geq A_{ii},$$

so by taking the max of both sides, we have $L \geq L_{\max}$. On the other hand, we have by the relationship between trace and sum of eigenvalues (A.1) that

$$L = \lambda_{\max}(A) \leq \sum_{i=1}^{n} \lambda_i(A) = \sum_{i=1}^{n} A_{ii} \leq nL_{\max}.$$

We therefore have $L_{\max} \leq L \leq nL_{\max}$.[3] This comparison is favorable to coordinate descent; when $L_{\max} < L$ we have a more favorable rate constant. The difference (a) (the factor of 4) should not be taken too seriously, as the rate constants depend strongly on the choice of α and thus on the availability of information about μ, L, $L_{\max}$. Moreover, the use of line searches (as opposed to constant α) can change the linear rate significantly in practice.

Stepping back: coordinate descent has some intuitive appeal over steepest descent in that the full gradient for the latter is evaluated at a single point, whereas in coordinate descent, we incorporate new information step-by-step into the gradient calculation as the coordinate steps are taken.

4.3 Block Coordinate Descent

Is it worth sketching some examples of how BCD is used in algorithms e.g. in alternating least squares for nonneg matrix factorization, inside ADMM, etc?

Exercises

1. Consider the convex quadratic $f(x) = (1/2)x^T Ax$, with A symmetric positive definite, for which $\nabla^2 f(x) \equiv A$....

[3] Probably can do better for an upper bound e.g $L \leq \sqrt{n}L_{\max}$.