EE 381V: Large Scale Optimization **Fall 2012**

Lecture 20 — November 8

Lecturer: Caramanis & Sanghavi *Scribe: Jason Jo*

20.1 Last time

In the previous lecture, we established convergence results for subgradient descent methods. Furthermore we proved the existence of a sequence of step sizes $\{\alpha_k\}$ such that we achieve the optimal convergence rate of $O\left(\frac{1}{\sqrt{k}}\right)$ where k denotes the iterate number.

20.2 Stochastic Subgradient Methods

We have previously covered various descent methods: gradient, coordinate, Newton's Method, etc. These methods were deterministic. Meaning that if we knew the previous iterate $x^{(k)}$, we could determine the next iterate $x^{(k+1)}$ with absolute certainty. For various reasons we want to introduce randomness into our descent methods. In general we have the following two cases in which it may be advantageous to use random methods:

1. We may have a deterministic problem, but it may be extremely large scale and using randomness makes it easier or tractable.

2. Or the problem inherently involves randomness in its formulation.

For the remainder of this lecture, we will focus on the first case and we will cover the second case in the next lecture, Lecture 21. We note that for the first case, it will be crucial to derive some sort of probabilistic control of the algorithm and obtain some sort of "with high probability" convergence results.

20.2.1 Examples

1. **Support Vector Machines**. Recall we used Support Vector Machines to classify points $x_i \in \mathbb{R}^n$ with labels $y_i \in \{+1, -1\}$ for $i \in [m]$ using a linear classifier.

   In the above situation in which the data is linearly separable, we may pose the following optimization problem:

$$\begin{aligned} \min_w \quad & \frac{1}{2}||w||_2^2 \\ \text{s.t.} \quad & y_i(\langle w, x_i\rangle - 1) \geq 1 \text{ for all } i \end{aligned} \tag{20.1}$$

   which we referred to as *maximal margin classification*.

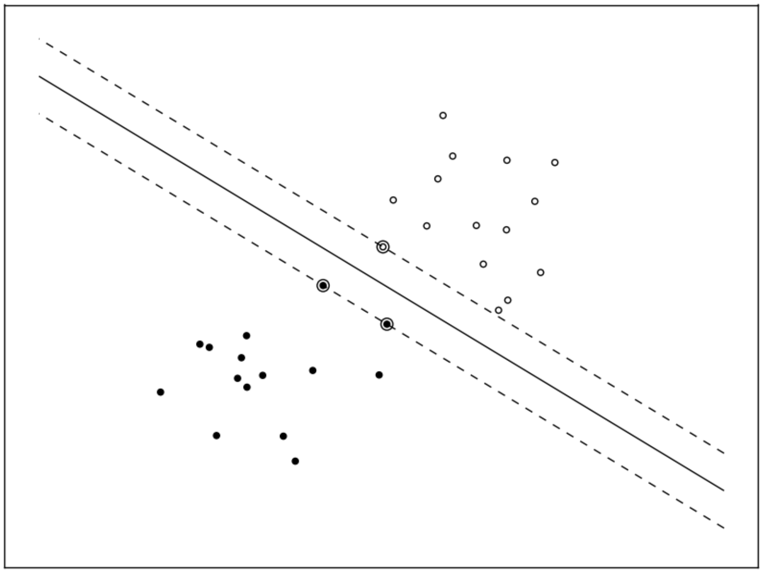

Figure 20.1. A linearly separable dataset and an example of a linear classifier.

But if the data is not linearly separable, then our convex optimization problem is not feasible. The natural extension is to replace our constraint with a loss function $\ell(y, \langle w, x\rangle)$. Now we may pose the unconstrained optimization problem:

$$\min_w \quad \frac{\lambda}{2}||w||_2^2 + \frac{1}{m}\sum_{i=1}^{m} \ell(y_i, \langle w, x_i\rangle) \tag{20.2}$$

One common loss function used is the *hinged loss function* $\ell(y, \langle w, x\rangle) = \max(0, 1 - y\langle w, x\rangle)$. This is not a differentiable function and if we wish to use a descent method, we can use the subgradient descent methods from the previous lecture. The λ term can be interpreted as a tradeoff term between penalizing misclassifications and maximizing the margin.

Depending on the type of loss function, descent methods may be prohibitively expensive, i.e. computationally slow. For example, if each data point $x_i \in \mathbb{R}^n$ lives in extremely high dimensional space, i.e. $n \gg 1$ or if we have too many samples $m \gg 1$ to process, the time to solve the convex optimization problem is very slow.

To make our problem tractable, instead of processing all the points, we will subsample our data points in a uniformly random fashion. In particular, at each step k, we will uniformly choose a subset of indices $\mathcal{S}_k \subseteq [m]$ and solve the smaller optimization problem:

$$\min_w \frac{\lambda}{2}||w||_2^2 + \frac{1}{|\mathcal{S}_k|}\sum_{i\in\mathcal{S}_k} \ell(y_i, \langle w, x_i\rangle). \tag{20.3}$$

We then have the descent direction $\tilde{g}^{(k)} = \lambda w^{(k)} + \frac{1}{|\mathcal{S}_k|}\sum_{i\in\mathcal{S}_k} \nabla_w \ell(y_i, \langle w^{(k)}, x_i\rangle)$ where $\nabla_w \ell(y_i, \langle w^{(k)}, x_i\rangle)$ can be the subgradient if ℓ is not differentiable. And we have the following Stochastic Subgradient Descent Algorithm for SVM:

$$w^{(k+1)} = w^{(k)} - \eta_k \tilde{g}^{(k)},$$

for η_k step size chosen in some manner (maybe exact line search or backtracking line search) to guarantee descent. For simple gradient descent, we would have had the descent direction at iterate k:

$$f^{(k)} := \nabla_w f = \lambda w^{(k)} + \frac{1}{m}\sum_{i=1}^{m} \nabla_w \ell(y_i, \langle w^{(k)}, x_i\rangle). \tag{20.4}$$

To see how $f^{(k)}$ and $\tilde{g}^{(k)}$ are related, observe that:

$$\mathbb{E}[\tilde{g}^{(k)} | w^{(k)}] = f^{(k)}. \tag{20.5}$$

Exercise 1. *Prove Equation (20.5).*

The descent direction $\tilde{g}^{(k)}$ is referred to as a *Noisy Unbiased Estimator* as it agrees on average with the deterministic object we are estimating. We note that while our stochastic descent direction $\tilde{g}^{(k)}$ agrees with the deterministic descent direction $f^{(k)}$ in expectation, there will be a variance in our estimates. The more samples we take in each step, the lower the variance gets. However the more samples we take the longer it takes per each iteration. Such is the tradeoff.

2. **Random Coordinate Descent**. We wish to minimize a function $f(x), x \in \mathbb{R}^n$. Random coordinate descent operates at each step k by choosing a coordinate index $i \in [n]$ uniformly random, and then making the following updates:

$$x_i^{(k+1)} = x_i^{(k)} - \eta_k \nabla_i f(x^{(k)}), \tag{20.6}$$
$$x_j^{(k+1)} = x_j^{(k)} \text{ for } j \neq i. \tag{20.7}$$

   Since we are choosing the descent coordinate in a uniformly random fashion, we will again obtain a noisy unbiased estimator.

3. $\min_{c\in\mathcal{C}} f_c(x_c)$, where we wish to minimize a sum of convex functions $f_c, c \in \mathcal{C}$ which each depends on a possible overlapping subset of variables x_c. For example in network capacity problems, we can define node capacity functions and there is no central function which knows everything. We can handle this optimization randomly by uniformly choosing one of the convex functions to minimize and perform a descent method on that, and iterate through.

20.3 Convergence of Noisy Unbiased Subgradient Methods

In this section we will focus solely on unconstrained optimization problems.

Definition 1. *$\tilde{g}(x)$ is a Noisy Unbiased Subgradient (NUS) if:*

$$\mathbb{E}[\tilde{g}(x)|x] \in \partial f(x) \text{ a.s.} \tag{20.8}$$

Note that x itself can be random. Recall from our previous examples, we have other sources of randomness as well. For example, for the SVM we randomly choose which data points to include in our empirical loss function.

Definition 2. *The Stochastic Subgradient Method is defined as:*

$$x^{(k+1)} = x^{(k)} - \alpha_k \tilde{g}(x^{(k)}), \tag{20.9}$$

for some appropriately chosen step size α_k.

We have the following convergence result:

Theorem 20.1. *For f a convex function with minimal value $f^* > -\infty$, if we have the following variance bound:*

$$\mathbb{E}[||\tilde{g}(x)||^2 \big| x] \leq G^2, \tag{20.10}$$

and initial error bound $||x^{(0)} - x^*||^2 \leq R^2$ and $f_{\text{best},k} = \min_{i \leq k} f(x^{(i)})$, then we have that:

$$\mathbb{E}[f_{best,k} - f^*] \leq \frac{R^2 + G^2 \sum_{i=0}^{k} \alpha_i^2}{2 \sum_{i=0}^{k} \alpha_i}. \tag{20.11}$$

Proof: Conditioning on $x^{(k)}$ we have the following:

$$\begin{aligned}
\mathbb{E}[||x^{(k+1)} - x^*||^2 \big| x^{(k)}] &= \mathbb{E}[||x^{(k)} - \alpha_k \tilde{g}^{(k)} - x^*||^2 \big| x^{(k)}] \\
&= ||x^{(k)} - x^*||^2 - 2\alpha_k \mathbb{E}\left[\tilde{g}^{(k)} \big| x^{(k)}\right]^T (x^{(k)} - x^*) + \alpha_k^2 \mathbb{E}\left[||\tilde{g}^{(k)}||^2 \big| x^{(k)}\right] \\
&\leq ||x^{(k)} - x^*||^2 - 2\alpha_k (f(x^{(k)} - f^*) + \alpha_k^2 G^2 \text{ by our hypotheses}
\end{aligned}$$

Now we may take the expectations of both sides with respect to $x^{(k)}$ (which is a random variable) to obtain:

$$\mathbb{E}[||x^{(k+1)} - x^*||^2] \leq \mathbb{E}[||x^{(k)} - x^*||^2] - 2\alpha_k \left(\mathbb{E}[f(x^{(k)})] - f^*\right) + \alpha_k^2 G^2. \tag{20.12}$$

Iteratively applying this bound yields:

$$\begin{aligned}
\mathbb{E}\left[||x^{(k+1)} - x^*||^2\right] &\leq \mathbb{E}\left[||x^{(0)} - x^*||^2\right] - 2\sum_{i=0}^{k} \alpha_i \left(\mathbb{E}[f(x^{(i)})] - f^*\right) + G^2 \sum_{i=0}^{k} \alpha_i^2 \\
&\leq R^2 - 2\sum_{i=0}^{k} \alpha_i \left(\mathbb{E}[f(x^{(i)})] - f^*\right) + G^2 \sum_{i=0}^{k} \alpha_i^2 \\
\Rightarrow 2\sum_{i=0}^{k} \alpha_i \left(\mathbb{E}[f(x^{(i)})] - f^*\right) &\leq R^2 + G^2 \sum_{i=0}^{k} \alpha_i^2 - \mathbb{E}\left[||x^{(k+1)} - x^*||^2\right] \\
&\leq R^2 + G^2 \sum_{i=0}^{k} \alpha_i^2
\end{aligned}$$

Next observe that $f_{\text{best},k} := \min_{i \leq k} f(x^{(i)}) \leq f(x^{(i)})$ for all i, therefore we have that:

$$\mathbb{E}[f_{\text{best},k}] \leq \mathbb{E}[f(x^{(i)})]. \tag{20.13}$$

Applying this inequality to our previous string of inequalities yields:

$$\begin{aligned}
\left(\mathbb{E}[f_{\text{best},k}] - f^*\right) 2\sum_{i=0}^{k} \alpha_i &\leq 2\sum_{i=0}^{k} \alpha_i \left(\mathbb{E}[f(x^{(i)})] - f^*\right) \\
&\leq R^2 + G^2 \sum_{i=0}^{k} \alpha_i^2,
\end{aligned}$$

implying the desired inequality:

$$\mathbb{E}[f_{\text{best},k}] - f^* \leq \frac{R^2 + G^2 \sum_{i=0}^{k} \alpha_i^2}{2\sum_{i=0}^{k} \alpha_i}. \tag{20.14}$$

And the conclusion holds. □

Recall from the previous lecture, we had the exact same bound for the deterministic subgradient method, i.e. without the expectations. For a particular choice of the stepsizes $\{\alpha_i\}$ we can obtain the optimal convergence rate of $O(1/\sqrt{k})$. However all we have shown is that we have proved this convergence rate in expectation, thus we may only expect it on average.

Can we expect anything better? Of course we could run the algorithm multiple times and do some sort of averaging process, but this sounds less than ideal. Observe that structurally we have obtained a bound on the error in expectation. We may use the *Markov Inequality* to obtain a bound in probability. A bound in probability is a stronger result than a bound in error:

Theorem 20.2. *Markov Inequality. For any random variable X and any $a > 0$:*

$$\mathbb{P}[|X| \geq a] \leq \frac{\mathbb{E}[|X|]}{a}. \tag{20.15}$$

Proof: Consider the scaled indicator function $aI_{|X|\geq a}$. One property of this function is that $aI_{|X|\geq a} \leq X$. Taking the expectation yields:

$$\begin{aligned} \mathbb{E}[aI_{|X|\geq a}] &\leq \mathbb{E}[X] \\ \Rightarrow a\mathbb{E}[I_{|X|\geq a}] &\leq \mathbb{E}[X] \\ \Rightarrow \mathbb{E}[I_{|X|\geq a}] &\leq \frac{\mathbb{E}[|X|]}{a} \\ \Leftrightarrow \mathbb{P}[|X| \geq a] &\leq \frac{\mathbb{E}[|X|]}{a}. \end{aligned}$$

□

Using the Markov Inequality, we immediately have the following corollary:

Corollary 20.3.

$$\mathbb{P}[f_{best,k} - f^* \geq \epsilon] \leq \frac{R^2 + G^2 \sum_{i=0}^k \alpha_i^2}{2\epsilon \sum_{i=0}^k \alpha_i} \tag{20.16}$$

Therefore we may conclude that with high probability, we will obtain the solution in one iteration of subgradient descent; we do not have to run subgradient descent numerous times and do some sort of averaging process. Actually, the stronger a.e.-convergence can be shown [Source?].

While we can expect with high probability that our randomized method will have the same error estimates as the corresponding deterministic methods, it is important to note that we can only hope to obtain the $O(1/\sqrt{k})$ convergence rate when we choose the descent step sizes $\{\alpha_i\}$ appropriately. Thus the actual convergence rate in practice may be far slower; this will vary depending on which problem we are attempting to solve.

20.4 Convergence Results for Random Coordinate Descent Method

In this section, we will quote results from Nesterov's paper on Random Coordinate Descent [Nesterov].

The Random Coordinate Descent algorithm is defined as:

```
Algorithm: Random Coordinate Descent
Data: $n$ is the dimensionality of our parameter $x$
Initialize $x^{(0)}$;
for $k = 0, 1, \dots$ do
    Uniformly choose a coordinate $i_k \in [n]$;
    Set $x_{i_k}^{(k+1)} = \text{argmin}_{x_{i_k}} f(x_{\setminus i_k}^{(k)}, x_{i_k})$ ;
    Set $x_{\setminus i_k}^{(k+1)} = x_{\setminus i_k}^{(k)}$ ;
end
```

For notational purposes we will define $\phi_k := \mathbb{E}[f(x^{(k)})]$. We then have the following theorem from [Nesterov]:

Theorem 20.4. *If $f(x)$ is a smooth function, $x \in \mathbb{R}^n$ with a uniform bound on the second order partial derivatives:*

$$\frac{\partial^2 f}{\partial x_i^2} \leq L \;\forall i,$$

and we have the initial error bound $||x^{(0)} - x^*||^2 \leq R^2$. Then we have the following convergence result in expectation:

$$\phi_k - f^* \leq \frac{2n}{k+4} L R^2. \tag{20.17}$$

Contrast this convergence rate to the deterministic gradient descent convergence rate when we assume an upper bound $\nabla^2 f \preceq \hat{L} I$, we have the convergence rate:

$$f(x^{(k)}) - f^* \leq \frac{2\hat{L}}{k+4} R^2. \tag{20.18}$$

How do these methods compare? How different can L and $n\hat{L}$ be? Each diagonal entry of $\nabla^2 f$ is bounded above by L. Because the trace of a symmetric matrix is equal to the sum of the singular values, and we have furthermore since $\nabla^2 f$ is a symmetric semipositive definite we have that:

$$\begin{aligned}
\hat{L} &= \lambda_{\max} \\
&\leq \sum_{i=1}^{n} \lambda_i \text{ because } \nabla^2 f \succeq 0 \Rightarrow \lambda_i \geq 0 \;\forall i, \\
&= \sum_{i=1}^{n} \nabla^2 f(i,i) \text{ by our trace comment above} \\
&\leq nL.
\end{aligned}$$

Therefore, in general $\hat{L} \leq nL$ and we conclude that the random coordinate descent method will have a slower convergence rate than the deterministic coordinate descent method. Of course the tradeoff is that each iterate is much simpler for the random coordinate descent method.

Suppose f has better regularity/convexity properties. In particular if f is strongly convex, we have the following convergence result:

Theorem 20.5. *If f is strongly convex with parameters $0 < m \leq M$ such that $mI \preceq \nabla^2 f \preceq MI$, then we have the following convergence rate for the random coordinate descent method:*

$$\phi_k - f^* \leq \left(1 - \frac{m}{nM}\right)^k (f^{(0)} - f^*). \tag{20.19}$$

Let's again compare the randomized method to its deterministic version. Recall for the deterministic coordinate descent method under strong convexity assumptions has the following convergence rate:

$$f(x^{(k)}) - f^* \leq \left(1 - \frac{m}{M}\right)^k (f_0 - f^*)$$

By comparing their corresponding convergence rates, it becomes clear that the random coordinate descent method becomes an attractive option once the ratio cost per iterate of the random method to the cost per iterate of the deterministic method becomes $O(\frac{1}{n})$. Then this speed up in per iterate time will counter balance the slower linear convergence.

Similar to Stochastic Subgradient Methods, we may apply the Markov Inequality to pass from convergence in expectation to convergence with high probability.

20.5 Next time

For the next lecture, we will consider the case of stochastic optimization methods when our problem formation inherently involves randomness. Here the problem itself will be random and may take the form:

$$\begin{array}{ll} \min_w & \mathbb{E}_w[f_0(x, w)] \\ \text{s.t.} & \mathbb{E}[f_i(x, w)] \leq 0 \text{ for } i = 1, \dots, m. \end{array} \tag{20.20}$$

If each f_i is a convex function in x for each w, the overall problem will be convex.

Bibliography

[Nesterov] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. Technical report, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, 2010. Discussion paper.