CS 49/149: 21st Century Algorithms (Fall 2018): Lecture 8

Date: 11th October, 2018
Topic: Stochastic Gradient Descent
Scribe: Zephyr Lucas

Disclaimer: These notes have not gone through scrutiny and in all probability contain errors. Please email errors to deeparnab@dartmouth.edu.

---

1 Motivation

Consider the following typical convex optimization problem in Machine Learning:

$$\min_{x \in S} (f(x)) = \min_{x \in S} \left(\sum_{t=1}^{T} (f_i(x)) \right) \tag{1}$$

1.1 Least Squares

Finding the line which best represents a set of input points in $\mathbb{R}^n$. Given some set of T "data points" $\mathbf{a}_t$ (each of which is an element of $\mathbb{R}^n$), we want to find the $\mathbf{x}$ and $\mathbf{b}$ which minimizes:

$$\sum_{t=1}^{T} \left(\left|\left| \mathbf{a}_t^T \mathbf{x} - \mathbf{b} \right|\right|_2 \right) \tag{2}$$

Note that each of the $\left(\left|\left| \mathbf{a}_t^T \mathbf{x} - \mathbf{b} \right|\right|_2^2 \right)$ in the sum are convex functions (because they are linear).

1.2 Support Vector Machine

Classification problem to find the hyper plane that best divides the data.

- Input: data in the form: $\{(\mathbf{a}_t, b_t) | \mathbf{a}_t \in \mathbb{R}^n,\ b_t \in \{-1, 1, \}\}$
- Goal: Find a hyper plane that separates the points with b_t positive and the points with b_t negative.

This problem we have already solved using MWU. But what happens if no plane exists? Well, then we would like to minimize the number of mistakes. That is we find a hyper plane given by $\mathbf{x}$ such that the number of ordered pairs $(\mathbf{a}_t, b_t)$ such that $b_t \neq \text{sign}(\mathbf{a}_t \cdot \mathbf{x})$ is minimized. This is an NP-problem, so most likely we can't find an efficient solution. So instead we introduce a "Proxy" which captures the properties we would like:

1.2.1 Proxy Problem

In a perfect world we want try to find a hyper plane $\mathbf{x}$ which minimizes:

$$\sum_{t=1}^{T} (\Pi(t)) \quad , \qquad \Pi(t) = \begin{cases} 1 & \text{If } b_t \neq \text{sign}\,(\mathbf{a}_t \cdot \mathbf{x}) \\ 0 & \text{Otherwise} \end{cases}$$

So instead we find a hyper plane $\mathbf{x} \in \mathbb{R}^n$ which minimizes:

$$\sum_{t=1}^{T} (\max(0, 1 - b_t(\mathbf{a}_t \cdot \mathbf{x})))$$

This new function is convex (It is the *convexification* of the perfect goal) which is a nice property to have, but it is still not perfect because as we scale $\mathbf{x}$ up, then this becomes uniformly zero around the origin, and minimizing a constant is fairly uninteresting. So instead we minimize:

$$\sum_{t=1}^{T} (\max(0, 1 - b_t(\mathbf{a}_t \cdot \mathbf{x}))) + \frac{\lambda}{2} ||\mathbf{x}||_2^2$$

Which is still a convex function, so we can run gradient decent and our problem is solved? Not quite.

1.2.2 Gradient Decent on the Support Vector Machine Problem

We have our function:

$$f(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^{m} (f_i(\mathbf{x})) \qquad \nabla f(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^{m} (\nabla f_i(\mathbf{x}))$$

But calculating this gradient is very computationally expensive. Is there any way we can approximate this, and run gradient descent on the approximation? It turns out that gradient decent is very robust, and therefore an approximate gradient is good enough.

2 Stochastic Gradient Descent

Like in regular (unbounded) gradient descent, we are trying to minimize some convex function. There we approached the minimum by iterating with the rule:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t) \tag{3}$$

The high level idea of stochastic gradient descent is, instead of following the gradient we follow an (randomized) estimate of the gradient.

STOCHASTIC GRADIENT DESCENT

- Assume there exists and estimator $\mathbf{E} : \mathbb{R}^n \to \mathbb{R}^n$ ($\mathbf{E}$ is a randomized algorithm) with:
  - $\mathbf{E} : \mathbf{z} \mapsto \mathbf{g}(\mathbf{z})$ where $\mathbf{g}(\mathbf{z})$ is a random vector.
  - The expectation: $\mathbb{E}(\mathbf{g}(\mathbf{z})|\mathbf{z})$ is a sub-gradient of f. In particular, if f is differentiable, then this should give ∇f).
- Suppose $f(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^{m} (f_i(\mathbf{x}))$. Then we can sample $i \in \{1, 2, ...m\}$ uniformly at random and set $g(\mathbf{x}) = \nabla f_i(\mathbf{x})$ as out estimator for this iteration.

- Do regular gradient descent but update with $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t g(\mathbf{x}_t)$.

We need to see if this is a good estimator:

$$\mathbb{E}\left[g(\mathbf{x})|\mathbf{x}\right] = \sum_{i=1}^{m}\left(\frac{1}{m}\nabla f_i(\mathbf{x})\right)$$

Notice this is the gradient!! and g is much easier to calculate as you don't need to evaluate the whole sum each iteration.

2.1 Analysis (in the unconstrained case)

In normal gradient decent we have the following steps:

Proof.

$$\begin{aligned}
\textit{err}(t) &\leq (\mathbf{x}_t - \mathbf{x}_*)^T \nabla f(\mathbf{x}_t) \\
&\leq \frac{1}{\eta}(\mathbf{x}_t - \mathbf{x}_*)^T(\mathbf{x}_t - \mathbf{x}_{t+1}) \\
&\leq \frac{1}{2\eta}\left(||\mathbf{x}_t - \mathbf{x}_*||_2^2 - ||\mathbf{x}_{t+1} - \mathbf{x}_*||_2^2 + ||\mathbf{x}_{t+1} - \mathbf{x}_t||_2^2\right) \\
&\leq \frac{1}{2\eta}\left(D_t^2 - D_{t+1}^2\right) + \frac{\eta}{2}||\nabla f(\mathbf{x}_t)||_2^2
\end{aligned}$$

$$\therefore \sum_{t=1}^{T}(\textit{err}(t)) \leq \frac{1}{2\eta}D_1^2 + \frac{\eta}{2}\sum_{t=1}^{T}\left(||\nabla f(\mathbf{x}_t)||_2^2\right)$$

□

But if we are no longer using the gradient as our function that takes us from $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$, then the first inequality is no longer true. Instead we find (regardless of our choice of $\mathbf{g}$)

$$\sum_{t=1}^{T}\left(\left(\mathbf{x}_t - \mathbf{x}_*\right)^T \mathbf{g}_t\right) \leq \frac{1}{2\eta}||\mathbf{x}_1 - \mathbf{x}_*||_2^2 + \frac{\eta}{2}\sum_{t=1}^{T}||\mathbf{g}_t||_2^2$$

Because we know the expected value of $\mathbf{g}_t$ is the gradient we can take expectations:

$$\mathbb{E}\left(\text{LHS}\right) = \sum_{t=1}^{T}\left(\left(\mathbf{x}_t - \mathbf{x}_*\right)^T \nabla f(\mathbf{x}_t)\right)$$

$$\mathbb{E}\left(\text{RHS}\right) = \frac{1}{2\eta}||\mathbf{x}_1 - \mathbf{x}_*||_2^2 + \frac{\eta}{2}\sum_{t=1}^{T}\left(\mathbb{E}\left(||\mathbf{g}_t||_2^2\right)\right)$$

Stochastic Gradient Descent

For *any* $\{\mathbf{g}_1, \mathbf{g}_2, \ldots \mathbf{g}_T\}$, running gradient descent with $x_{t+1} = x_t - \eta g_t$, then we know:

$$\sum_{t=1}^{T} (\mathbf{x}_t - \mathbf{x}_*)^T \mathbf{g}_t \leq \frac{1}{2\eta} ||\mathbf{x}_1 - \mathbf{x}_*||_2^2 + \frac{\eta}{2} \cdot \sum_{t=1}^{T} \left(||\mathbf{g}_t||_2^2 \right)$$

When we are doing Stochastic gradient descent, our $\mathbf{g}_t$ is a random vector with an additional property, namely that $\mathbf{g} = g(\mathbf{x}_t)$ with $\mathbb{E}(\mathbf{g}_t|\mathbf{x}_t) = \nabla f(\mathbf{x}_t)$. Further we find that at the end of the algorithm,

$$\mathbb{E}_{\text{final}}(\mathbf{g}_t) = \mathbb{E}_{t+1}(\mathbf{g}_t) = \mathbb{E}_t(\mathbb{E}(\mathbf{g}_t|\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_t)$$

Therefore we find:

$$\mathbb{E}_t \left[\nabla f(\mathbf{x}_t) \right] = \mathbb{E} \left[\nabla f(\mathbf{x}_t) \right]$$

Which gives us:

$$\begin{aligned} \mathbb{E}\left[\sum_{t=1}^{T} \left((\mathbf{x}_t - \mathbf{x}_*)^T \mathbf{g}_t \right) \right] &= \sum_{t=1}^{T} \left(\mathbb{E}\left[(\mathbf{x}_t - \mathbf{x}_*)^T \mathbf{g}_t \right] \right) \\ &= \sum_{t=1}^{T} \left(\mathbb{E}_{\mathbf{x}_1, \ldots \mathbf{x}_t} \left[\mathbb{E}_{t+1} \left[(\mathbf{x}_t - \mathbf{x}_*)^T \mathbf{g}_t \right] \right] \right) \end{aligned}$$

Now, $\mathbb{E}_{t+1}\left[(\mathbf{x}_t - \mathbf{x}_*)^T \mathbf{g}_t \right]$ is a random variable, but when we calculate it we get to fix the choices for $\mathbf{x}_1, \ldots \mathbf{x}_t$. So this becomes:

$$\sum_{t=1}^{T} \left(\mathbb{E}_{\mathbf{x}_1, \ldots \mathbf{x}_t} \left[\mathbb{E}\left[(\mathbf{x}_t - \mathbf{x}_*)^T \mathbf{g}_t | \mathbf{x}_t \right] \right] \right)$$

Now we observe that fixing $\mathbf{x}_t$ causes this to become deterministic.

$$\begin{aligned}
\mathbb{E}_{\text{final}}\left[\sum_{t=1}^{T}\left(\left(\mathbf{x}_t-\mathbf{x}_*\right)^T\mathbf{g}_t\right)\right] &= \sum_{t=1}^{T}\left(\mathbb{E}_{\mathbf{x}_1,\mathbf{x}_2,\ldots\mathbf{x}_t}\left[\left(\mathbf{x}_t-\mathbf{x}_*\right)^T\mathbb{E}\left(\mathbf{g}_t|\mathbf{x}_t\right)\right]\right)\\
&= \sum_{t=1}^{T}\left(\mathbb{E}_{\mathbf{x}_1,\mathbf{x}_2,\ldots\mathbf{x}_t}\left[\left(\mathbf{x}_t-\mathbf{x}_*\right)^T\nabla f(\mathbf{x}_t)\right]\right)\\
&\geq \sum_{t=1}^{T}\left(\mathbb{E}_{\mathbf{x}_1,\mathbf{x}_2,\ldots\mathbf{x}_t}\left[f(\mathbf{x}_t)-f(\mathbf{x}_*)\right]\right)\\
&\geq \sum_{t=1}^{T}\left(\mathbb{E}\left[f(\mathbf{x}_t)\right]\right)-Tf(\mathbf{x}_*)\\
\mathbb{E}_{\text{final}}\left[\sum_{t=1}^{T}\left(\frac{1}{T}\left(\mathbf{x}_t-\mathbf{x}_*\right)^T\mathbf{g}_t\right)\right] &\geq \frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}\left[f(\mathbf{x}_t)\right]\right)-f(\mathbf{x}_*)\\
&\geq \mathbb{E}\left[\sum_{t=1}^{T}\left(\frac{f(\mathbf{x}_t)}{T}\right)\right]-f(\mathbf{x}_*)
\end{aligned}$$

Because f is convex, we know:

$$\frac{1}{T}\sum_{t=1}^{T}\left(f(\mathbf{x}_t)\geq f\left(\frac{1}{T}\sum_{t=1}^{T}\left(\mathbf{x}_t\right)\right)\right.$$

Therefore we have:

$$\begin{aligned}
\mathbb{E}_{\text{final}}\left[\sum_{t=1}^{T}\left(\frac{1}{T}\left(\mathbf{x}_t-\mathbf{x}_*\right)^T\mathbf{g}_t\right)\right] &\leq \frac{1}{T}\mathbb{E}\left[\frac{1}{2\eta}\left|\left|\mathbf{x}_1-\mathbf{x}_*\right|\right|_2^2+\frac{\eta}{2}\frac{\eta}{2}\sum_{t=1}^{T}\left(\left|\left|\mathbf{g}_t\right|\right|_2^2\right)\right]\\
&\leq \frac{1}{2\eta T}\left|\left|\mathbf{x}_1-\mathbf{x}_*\right|\right|_2^2+\frac{\eta}{2T}\sum_{t=1}^{T}\left(\mathbb{E}_{\text{final}}\left[\left|\left|\nabla f(\mathbf{x}_t)\right|\right|_2^2\right]\right)
\end{aligned}$$

Quality of an Estimator

The *quality of an estimator* for stochastic gradient descent is given by ρ, where

$$\mathbb{E}\left[\left|\left|\mathbf{g}(\mathbf{z})\right|\right|_2^2\right]\leq\rho^2$$

Example

If we let:

$$f(\mathbf{x})=\frac{1}{2m}\left|\left|A\mathbf{x}-\mathbf{b}\right|\right|_2^2=\frac{1}{2m}\sum_{i=1}^{m}\left(\left(\mathbf{a}_i^T\mathbf{x}-b_i\right)^2\right)$$

and we let $\mathbf{g}$ be given by sampling i from $\{1, 2, \ldots m\}$ and return $\mathbf{a}_i^T\mathbf{x} - b_i$. This means:

$$\mathbb{E}\left[||\mathbf{g}||_2^2\right] = \frac{1}{m} \cdot \sum_{i=1}^{m} \left(||a_i||_2^2 \cdot (\mathbf{a}_i^T\mathbf{x} - b_i)^2\right) = \frac{1}{m} ||A||_F$$

Where $||A||_F$ denotes the Frobemius norm of A, which is simply the sum of the squares of the matrix entries. The Frobemius norm is always larger than the largest eigenvector, but they are generally fairly similar. So even if you have to run slightly longer, you get huge savings. Here the quality of the estimator is actually given by $\rho = ||A||_F$!

2.2 Stochastic Gradient Descent Theorem with Minimal Assumptions

If we have an estimator with a high enough quality then Stochastic Gradient Descent can guarantee:

$$\mathbb{E}\left[f(\mathbf{x}_{(T)})\right] - f(\mathbf{x}_*) \leq \frac{D^2}{2\eta T} + \frac{\eta\rho^2}{2}$$

In particular this means if we want the right hand side to be less than ϵ, then we must set η and run for iterations given by:

$$\eta = \frac{\epsilon}{2\rho} \ , \qquad T = \frac{D^2\rho^2}{\epsilon^2}$$

Instead of sampling just one i, we can sample a bunch of different is to construct our approximation of the gradient. This procedure is called Batch Stochastic Gradient Descent and reduces the variance without changing the expectation. In theory, batching doesn't change anything, but in practice it tends to perform much better, particularly in distributed computing environment where we get to "reuse" previous gradients.

2.2.1 Other forms of Stochastic Gradient Descent

What happens when our function is some other really complicated (non-sum based) function, that was really expensive to evaluate. We can instead pick one dimension and walk only in that direction based only off that coordinate's gradient. This means we only have to consider one dimension of our gradient each iteration and not the whole thing. In particular this means we have:

$$[\mathbf{x}_{t+1}]_i = \mathbf{x}_t - \eta \cdot n \cdot [\nabla f(\mathbf{x})]_i$$

Which still has:

$$\mathbb{E}\left[\mathbf{x}_{t+1} - \mathbf{x}_t\right] = -\eta\nabla f(\mathbf{x}_t)$$

In practice, we don't chose our dimension (i) uniformly at random, but instead proportion the probabilities based off the smoothness of the function.