CS-GY 9223 I: Lecture 8
Coordinate decent and non-convex models.

NYU Tandon School of Engineering, Prof. Christopher Musco

Main idea: Trade slower convergence (more iterations) for cheaper iterations.

Stochastic Coordinate Descent: Only compute a single random entry of $\nabla f(\mathbf{x})$ on each iteration:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}) \\ \frac{\partial f}{\partial x_2}(\mathbf{x}) \\ \vdots \\ \frac{\partial f}{\partial x_d}(\mathbf{x}) \end{bmatrix} \qquad \nabla_i f(\mathbf{x}) = \begin{bmatrix} 0 \\ \frac{\partial f}{\partial x_i}(\mathbf{x}) \\ \vdots \\ 0 \end{bmatrix}$$

Update: $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} + \eta \nabla_i f(\mathbf{x}^{(t)})$.

COORDINATE DESCENT

When $\mathbf{x}$ has d parameters, computing $\nabla_i f(\mathbf{x})$ often costs just a $1/d$ fraction of what it costs to compute $\nabla f(\mathbf{x})$

Example: $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ for $\mathbf{A} \in \mathbb{R}^{n \times d}, \mathbf{x} \in \mathbb{R}^d, \mathbf{b} \in \mathbb{R}^n$.

- $\nabla f(\mathbf{x}) = 2\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{A}^T\mathbf{b}$.
- $\nabla_i f(\mathbf{x}) = 2\left[\mathbf{A}^T\mathbf{A}\mathbf{x}\right]_i - 2\left[\mathbf{A}^T\mathbf{b}\right]_i$.

- $\mathbf{A}\mathbf{x}^{(t+1)} = \mathbf{A}\left(\mathbf{x}^{(t)} + c \cdot \mathbf{e}_i\right)$ — $O(n)$ time
- $2\left[\mathbf{A}^T\left(\mathbf{A}\mathbf{x}^{(t+1)} - \mathbf{b}\right)\right]_i$ — $O(n)$ time

Stochastic Coordinate Descent:

- Choose number of steps T and step size η.
- For $t = 1, \ldots, T$:
  - Pick random $j \in 1, \ldots, d$.
  - $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \eta \nabla_j f(\mathbf{x}^{(i)})$
- Return $\hat{\mathbf{x}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{x}^{(t)}$.

Theorem (Stochastic Coordinate Descent convergence)

Given a G-Lipschitz function f with minimizer $\mathbf{x}^*$ and initial point $\mathbf{x}^{(1)}$ with $\|\mathbf{x}^{(1)} - \mathbf{x}^*\|_2 \leq R$, SCD with step size $\eta = \frac{1}{Rd}$ satisfies the guarantee:

$$\mathbb{E}[f(\hat{\mathbf{x}}) - f(\mathbf{x}^*)] \leq \frac{2GR}{\sqrt{T/d}}$$

How can we improve on this?

Often it doesn't make sense to sample i uniformly at random:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -.5 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 10 \\ 42 \\ -11 \\ -51 \\ 34 \\ -22 \end{bmatrix}$$

Select indices i proportional to $\|\mathbf{a}_i\|_2^2$:

$$\Pr[\text{select index } i \text{ to update}] = \frac{\|\mathbf{a}_i\|_2^2}{\sum_{j=1}^{d} \|\mathbf{a}_j\|_2^2} = \frac{\|\mathbf{a}_i\|_2^2}{\|\mathbf{A}\|_F^2}$$

Let's analyze this approach.

Specialization of SCD to $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$:

Randomized Coordinate Descent (Strohmer, Vershynin 2007 / Leventhal, Lewis 2018)

- For iterate $\mathbf{x}^{(t)}$, let $\mathbf{r}^{(t)}$ be the residual:

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

- $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - c\mathbf{e}_j$. Here c is a scalar and $\mathbf{e}_j$ is a standard basis vector.
- $\mathbf{r}^{(t+1)} = \mathbf{r}^{(t)} - c\mathbf{a}_j$. Here $\mathbf{a}_j$ is the i^{th} column of $\mathbf{A}$.

What choice for c minimizes $\|\mathbf{r}^{(t+1)}\|_2^2$?

Specialization of SCD to $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$:

Randomized Coordinate Descent

- Choose number of steps T.
- Let $\mathbf{x}^{(1)} = \mathbf{0}$ and $\mathbf{r}^{(1)} = \mathbf{b}$.
- For $t = 1, \ldots, T$:
  - Pick random $j \in 1, \ldots, d$. Index j is selected with probability proportional to $\|\mathbf{a}_j\|_2^2 / \|A\|_F^2$.
  - Set $c = \mathbf{a}_j^T \mathbf{r}^{(t)} / \|\mathbf{a}_j\|_2^2$
  - $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - c\mathbf{e}_j$
  - $\mathbf{r}^{(t+1)} = \mathbf{r}^{(t)} - c\mathbf{a}_j$
- Return $\mathbf{x}^{(T)}$.

Claim

$$\mathbb{E}\|\mathbf{r}^{(t+1)}\|_2^2 = \|\mathbf{r}^{(t)}\|_2^2 - \frac{1}{\|\mathbf{A}\|_F^2}\|\mathbf{A}^T\mathbf{r}^{(t)}\|_2^2$$

Any residual $\mathbf{r}$ can be written as $\mathbf{r} = \mathbf{r}^* + \bar{\mathbf{r}}$ where $\mathbf{r}^* = \mathbf{A}\mathbf{x}^* - \mathbf{b}$ and $\bar{\mathbf{r}} = \mathbf{A}(\mathbf{x}^t - \mathbf{x}^*)$. Note that $\mathbf{A}^T\mathbf{r}^* = 0$ and $\bar{\mathbf{r}} \perp \mathbf{r}^*$.

Claim

$$\begin{aligned}\mathbb{E}\|\bar{\mathbf{r}}^{(t+1)}\|_2^2 &= \|\bar{\mathbf{r}}^{(t)}\|_2^2 - \frac{1}{\|\mathbf{A}\|_F^2}\|\mathbf{A}^T\bar{\mathbf{r}}^{(t)}\|_2^2 \\ &\geq \|\bar{\mathbf{r}}^{(t)}\|_2^2 - \frac{\lambda_{\min}(\mathbf{A}^T\mathbf{A})}{\|\mathbf{A}\|_F^2}\|\bar{\mathbf{r}}^{(t)}\|_2^2\end{aligned}$$

Theorem (Randomized Coordinate Descent convergence)

After T steps of RCD with importance sampling run on $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$, we have:

$$\mathbb{E}[f(\mathbf{x}^{(t)}) - f(\mathbf{x}^*)] \leq \left(1 - \frac{\lambda_{\min}(\mathbf{A}^T\mathbf{A})}{\|\mathbf{A}\|_F^2}\right)[f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*)]$$

Corollary: After $T = O(\frac{\|\mathbf{A}\|_F^2}{\lambda_{\min}(\mathbf{A}^T\mathbf{A})} \log \frac{1}{\epsilon})$ we obtain error $\epsilon\|\mathbf{b}\|_2^2$.

Is this more or less iterations than the $T = O(\frac{\lambda_{\max}(\mathbf{A}^T\mathbf{A})}{\lambda_{\min}(\mathbf{A}^T\mathbf{A})} \log \frac{1}{\epsilon})$ required for gradient descent to converge?

$$\|\mathbf{A}\|_F^2 = \mathrm{tr}(\mathbf{A}^T\mathbf{A}) = \sum_{i=1}^{d} \lambda_i(\mathbf{A}^T\mathbf{A})$$

$$\lambda_{\max}(\mathbf{A}^T\mathbf{A}) \leq \|\mathbf{A}\|_F^2 \leq d \cdot \lambda_{\max}(\mathbf{A}^T\mathbf{A})$$

For solving $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$,

$$(\#\text{ GD Iterations}) \leq (\#\text{ RCD Iterations}) \leq d \cdot (\#\text{ GD Iterations})$$

But RCD iterations are cheaper by a factor of d.

When does $\|A\|_F^2 = \text{tr}(A^T A) = d \cdot \lambda_{\max}(A^T A)$?

When does $\|A\|_F^2 = \text{tr}(A^T A) = 1 \cdot \lambda_{\max}(A^T A)$?

Roughly:

Stochastic Gradient Descent performs well when data points (rows) are repetitive.

Stochastic Coordinate Descent performs well when data features (columns) are repetitive.

NON-CONVEX OPTIMIZATION

VISUALIZATION

Given $f(\mathbf{x})$ which is potentially non-convex, find $\hat{\mathbf{x}}$ such that $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon$.

We understand very little about optimizing non-convex functions in comparison to convex functions, but not nothing. In many cases, we're still figuring out the right questions to ask.

Definition (Stationary point)

For a differentiable function f, a stationary point is any $\mathbf{x}$ with:

$$\nabla f(\mathbf{x}) = \mathbf{0}$$

local/global minima - local/global maxima - saddle points

Reasonable goal: Find an approximate stationary point $\hat{x}$ with

$$\|\nabla f(\hat{x})\|_2 \leq \epsilon.$$

SMOOTHNESS FOR NON-CONVEX FUNTIONS

Definition

A differentiable (potentially non-convex) function f is β smooth if for all $\mathbf{x}, \mathbf{y}$,

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq \beta \|\mathbf{x} - \mathbf{y}\|_2$$

Corollary: For all $\mathbf{x}, \mathbf{y}$

$$\left|\nabla f(\mathbf{x})^T(\mathbf{x} - \mathbf{y}) - [f(\mathbf{x}) - f(\mathbf{y})]\right| \leq \frac{\beta}{2}\|\mathbf{x} - \mathbf{y}\|_2^2.$$

GRADIENT DESCENT FINDS APPROXIMATE STATIONARY POINTS

Theorem

If Gradient Descent is run with step size $\eta = \frac{1}{\beta}$ *on a differentiable function* f *with global minimum* $\mathbf{x}^*$ *then after* $T = O(\frac{\beta[f(\mathbf{x}^{(1)}) - f(\mathbf{x}^*)]}{\epsilon})$ *we will find an* ϵ*-approximate stationary point* $\hat{\mathbf{x}}$.

GRADIENT DESCENT FINDS APPROXIMATE STATIONARY POINTS

Theorem

If Gradient Descent is run with step size $\eta = \frac{1}{\beta}$ *on a differentiable function* f *with global minimum* $\mathbf{x}^*$ *then after* $T = O(\frac{\beta[f(\mathbf{x}^{(1)}) - f(\mathbf{x}^*)]}{\epsilon})$ *we will find an* ϵ*-approximate stationary point* $\hat{\mathbf{x}}$.

Questions in non-convex optimization

If GD can find a stationary point and that seems to work for your problem, are there algorithms which find a stationary point faster using preconditioning, acceleration, stocastic methods, etc.?

Questions in non-convex optimization

What if my function only has global minima and stationary points? Randomized methods (SGD, perturbed gradient methods, etc.) can "escape" stationary points under some minor assumptions.

Example: $\min_x \frac{-x^T A^T A x}{x^T x}$

- **Global minimum**: Top eigenvector of $A^T A$ (i.e., top principal component of A).
- **Stationary points:** All other eigenvectors of A.

Useful for lots of other matrix factorization problems beyond vanilla PCA.

Questions in non-convex optimization

- Can random or careful initialization lead to a good minima?
- Can we escape “shallow” local minima.
- Is a global minima even needed?