IE 598: Big Data Optimization **Fall 2016**

Lecture 12: Coordinate Descent Algorithms

Lecturer: Niao He *Scriber: Lucas Buccafusca*

Overview: In this lecture we are going to discuss a new family of algorithm, called Coordinate Descent Methods. We will dicuss the general frameowrk and three diffirent variants of block coordinate gradient descent (Gauss Southwell, Randomized, and Cyclic) by emloying diffirent updating rules.

12.1 Introduction: Coordinate Descent Algorithms

Once again, the goal is to solve

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

where f is convex and smooth (f is continuously differentiable and gradient is Lipschitz continuous).

Motivation. As discussed in previous lectures, when n is large, it becomes computationally expensive to calculate full gradients, which means gradient descent is not necessarily always efficient. Observe that for unconstrained problems, x_* is an optimal solution if and only if $\nabla f(x_*) = 0$, namely, $\nabla_i f(x_*) = 0, \forall i = 1, \ldots, n$. To find the optimal solution, it makes sense to search along each coordinate direction; if at some point, the objective is not decreasing at every coordinate direction, then we have reached the optimum. This motivates the so called *Coordinate Minimization algorithms*, or also called *Coordinate Descent algorithms*.

Coordinate descent algorithms are derivative-free optimization methods.

Coordinate Minimization The general idea for a coordinate descent algorithm is shown below

Initialize $x^{(0)}$
  for $t = 1, 2, \ldots$
    Pick coordinate i from $1, 2, \ldots n$
    $x_i^{(t+1)} = \text{argmin}_{x_i\in\mathbb{R}} f(x_i, \omega_{-i}^t)$
  end

where ω_{-i}^t represent all other coordinates and are specified below.

The above framework is rather general, and there are many ways to choose ω_{-i}^t and to choose the coordinates to update at each iteration.

Rules for updating. We mention two styles here:

- **Gauss – Seidel style**

  $$\omega_{-i}^t = (x_1^{(t+1)}, \ldots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \ldots, x_n^{(t)})$$

  When updating each coordinate, the Gauss-Seidel style fixes the rest coordinates to be the most up-to-date solution, with a potential advantage of converging faster.

- **Jacobi style**

$$\omega_{-i}^{t} = (x_1^{(t)}, \ldots, x_{i-1}^{(t)}, x_{i+1}^{(t)}, \ldots, x_n^{(t)})$$

When updating each coordinate, the Jacobi style fixes the rest coordinates to be the solution obtained from previous cycle, i.e. it doesn't care about intermediary iterates until we complete all coordinates. The Jacobi update has a unique advantage; we can run the update for each coordinate in parallel.

Rules for selecting coordinates There are several ways and orders to decide which coordinates to update at each iteration.

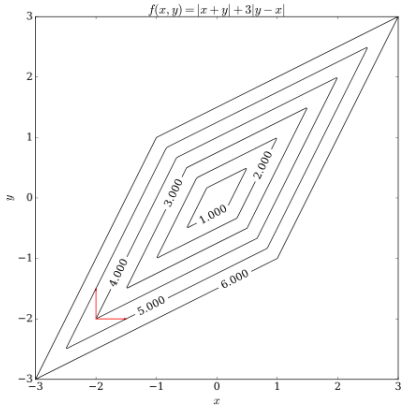
- **Cyclic Order** : run all coordinates in cyclic order, i.e. $1 \to 2 \to 3 \to \ldots \to n$ at each iteration
- **Random Sampling** : Randomly select some coordinate to update (sample with replacement)
- **Gauss−Southwell** : At each iteration, pick coordinate i so that $i = \underset{1 \le j \le n}{\operatorname{argmax}} |\nabla_j f(x^{(t)})|$
- **Random Permutation** : run cyclic order on a permuted index (sample without replacement)

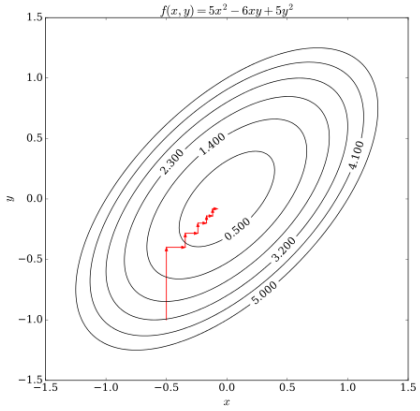
For example if $n = 3$ we could have the following:
Cyclic: $(1 \to 2 \to 3) \to (1 \to 2 \to 3) \to (1 \to 2 \to 3) \to \ldots$
Random permutation : $(2 \to 1 \to 3) \to (3 \to 1 \to 2) \to (1 \to 2 \to 3) \to \ldots$
Random sampling: $1 \to 1 \to 2 \to 3 \to 2 \to 1 \to \ldots$

Note that except for the Gauss-Southwell case, the general Coordinate Descemt algorithm can be seen as a derivative-free algorithm. Here are some immediate observations.

Remarks on Coordinate Descent

1. The objective function values are non-decreasing: $f(x^{(0)}) \ge f(x^{(1)}) \ge \ldots$
2. If f is strictly convex and smooth, the algorithm converges to a global minimum (optimal solution)
3. If f is non-convex or not even smooth, the algorithm might not converge at all.
   Example: $f(x, y) = |x + y| + 3|x - y|$.
   If started with $(x^0, y^0) = (-1, -1)$, the algorithm will not move. [1]





[1] Example and image from https://en.wikipedia.org/wiki/Coordinate descent

Suppose that the algorithm is on a corner point in the non-smooth case; then there are two directions it can try, indicated by the red arrows. However, every step along these two directions will increase the objective function's value, so the algorithm will not take a step, even though the sum of both steps would bring the algorithm closer to the optimum. We see strict convergence for the smooth case on the right.

The framework can be generalized for block updates e.g. you split your decision variable into blocks and you can cyclically update each block. This is usually known as *block coordinate descent*. In the case when we have tow blocks, block coordinate descent simply reduces to the *alternating minimization*.

Despite of its popularity, the global convergence rates of coordinate descent algorithms have only been recently established and most analysis focus on block coordinate gradient descent type of algorithms.

- CD with random sampling [Nesterov, 2010] and [Richtarik and Takac, 2011]
- CD with cyclic order and random permutation [Beck and Tetruashvili, 2013] and [Hong et.al., 2014]
- CD with Gauss-Southwell update [Nesterov, 2010] and [Nutini et.al., 2015]

12.2 Several Coordinate Descent Algorithms and Convergences

Consider the unconstrained optimization problem with block structure:

$$\min_{x=[x_1;\ldots;x_b]\in\mathbb{R}^{n_1}\times\ldots\mathbb{R}^{n_b}} f(x)$$

where $n_1+\ldots+n_b=n$. Let us decompose the identity matrix $I_{n\times n}=[U_1|\ldots|U_b]$, where U_i is a matrix of size n by n_i with diagonals being 1 and 0 elsewhere. Hence, for any vector x, we have $x=\sum_{i=1}^{b}U_ix_i$, and $x_i=U_i^Tx$.

We assume that

1. f is L-smooth, $||\nabla f(x)-\nabla f(y)||_2\leq L||x-y||_2$ for some Lipschitz constant $L\geq 0$.

2. $f(\cdot,x_{-i})$ is L_i-smooth, i.e. $||\nabla f(x+U_ih_i)-\nabla f(x)||_2\leq L_i||h_i||_2,\ \forall h_i\in\mathbb{R}^{n_i}$ and $x\in\mathbb{R}^n$.

Denote $L_{max}=\max_i L_i$ and $L_{min}=\min_i L_i$. We have $L_i\leq L\leq\sum_i L_i\leq b\cdot L_{max},\forall i=1,\ldots,b$.

12.2.1 Gauss-Southwell Block Coordinate Gradient Descent

GS-BCGD At iteration t, pick index $i_t=\underset{1\leq j\leq n}{\operatorname{argmax}}|\nabla_j f(x^{(t)})|$

$$x^{(t+1)}=x^{(t)}-\frac{1}{L}U_{i_t}\nabla_{i_t}f(x^{(t)})$$

Here, we use a fixed step-size of $\frac{1}{L}$ at each iteration.

Theorem 12.1 *If f is convex and L-smooth, we have*

$$f(x^{(t)}) - f^* \leq \frac{2Lb||x^0 - x^*||^2}{t}$$

If f is μ-strongly convex and L-smooth, then

$$f(x^{(t)}) - f^* \leq (1 - \frac{\mu}{bL})^t (f(x^0) - f^*)$$

Proof: We have

$$f(x^{(t)}) - f(x^{(t+1)}) \geq \frac{1}{2L}||\nabla f_{i_t}(x^{(t)})||_2^2 \geq \frac{1}{2Lb}||\nabla f(x^{(t)})||_2^2.$$

From the previous analysis of gradient descent, we see that the only difference is an additional b term that propagates through. Hence, following same arguments leads to the convergence results. ■

12.2.2 Randomized Block Coordinate Gradient Descent

Randomized-BCGD At iteration t, pick index i_t uniformly randomly from $\{1, \ldots, b\}$
$$x^{(t+1)} = x^{(t)} - \frac{1}{L} U_{i_t} \nabla_{i_t} f(x^{(t)})$$

Note that the index i_t is a discrete random variable, and $P(i_t = i) = \frac{1}{b}, i = 1, \ldots, b$

Theorem 12.2 *If f is convex and L-smooth, we have*

$$\mathbb{E}[f(x^{(t)}) - f^*] \leq \frac{2Lb \cdot R(x_0)^2}{t}$$

where $R(x_0) = \max\{||x - x^*|| : f(x) \leq f(x_0)\}$. If f is μ-strongly convex and L-smooth, then

$$\mathbb{E}[f(x^{(t)}) - f^*] \leq (1 - \frac{\mu}{bL})^t (f(x^0) - f^*)$$

Proof: In this case, we have

$$f(x^{(t)}) - f(x^{(t+1)}) \geq \frac{1}{2L}||\nabla f_{i_t}(x^{(t)})||_2^2$$

Taking expectation on both side over i_t: we have

$$f(x^{(t)}) - \mathbb{E}_{i_t} f(x^{(t+1)}) \geq \frac{1}{2L}\mathbb{E}_{i_t}||\nabla f_{i_t}(x^{(t)})||_2^2$$

Since $P(i_t = i) = \frac{1}{b}$, thus

$$f(x^{(t)}) - \mathbb{E}_{i_t} f(x^{(t+1)}) \geq \frac{1}{2L} \sum ||\nabla f_i(x^{(t)})||_2^2 P(i_t = i) = \frac{1}{2Lb}||\nabla f(x^{(t)})||_2^2$$

Similarly to above, we see that the differences are the expectation and the factor b. Another difference is that we will no longer have

$$f(x^{(t)}) - f^* \leq \|x^0 - x^*\| \cdot \|\nabla f(x^{(t)})\|_2$$

since we cnanot ensure that $\|x^0 - x^*\| \geq \|x^t - x^*\| \geq \|x^{t+1} - x^*\|$. But instead we have

$$f(x^{(t)}) - f^* \leq R(x_0)\|\nabla f(x^{(t)})\|_2.$$

Combining these two facts lead to the convergence results. ■

Remarks. There are several ways to improve the rate:

- Use larger step-sizes (use $\gamma_t = \frac{1}{L_{i_t}}$ instead of $\gamma_t = \frac{1}{L}$).
- Use non-uniform sampling distribution (use $P_i = \frac{L_i}{\sum L_i}$ instead of $P_i = \frac{1}{b}$) i.e. choose blocks with larger Lipschitz constants more frequently.

This gives rise to an improved convergence bound (see details in [Nesterov, 2010] and [Richtarik and Takac, 2011])

$$\mathbb{E}[f(x^{(t)}) - f^*] \leq \frac{2\sum_{i=1}^{b} L_i \cdot R(x_0)^2}{t}$$

For instance, in the case when $L_i = \frac{L}{b}, i = 1, \ldots, b$, the randomized block coordinate descent algorithm (Randomized-BCGD) achieves the same complexity result as the full gradient descent (GD) algorithm, but the iteration cost is $O(b)$ times cheaper.

12.2.3 Cyclic Block Coordinate Gradient Descent

Cyclic-BCGD	At iteration t, for $i = 1 \ldots b$, do
	$x_i^{(t)} = x_{i-1}^{(t)} - \frac{1}{L} U_i \nabla_i f(x_{i-1}^{(t)})$
	Set $x^{(t+1)} = x_b^{(t+1)}$

Note that the Cyclic-BCGD is a deterministic algorithm, and the iteration cost for Cyclic-BCGD is $O(b)$ times larger than the previous Randomized-BCGD since it needs to go through entire blocks.

Theorem 12.3 *If f is convex and L-smooth, we have*

$$f(x^{(t)}) - f^* \leq \frac{4L(b+1)R(x_0)^2}{t}.$$

If f is μ-strongly convex and L-smooth, it converges linearly:

$$f(x^{(t)}) - f^* \leq (1 - \frac{\mu}{2(b+1)L})^t (f(x^0) - f^*)$$

Proof: In this case, we have

$$f(x^{(t)}) - f(x^{(t+1)}) \geq \sum_{i=1}^{b} (f(x_{i-1}^{(t)}) - f(x_i^{(t)})) \geq \sum_{i=1}^{b} \frac{1}{2L} ||\nabla_i f(x_{i-1}^{(t)})||_2^2)$$

We now claim that:

$$||\nabla f(x^{(t)})||_2^2 \leq (2b+2) \sum_{i=1}^{b} ||\nabla_i f(x_{i-1}^{(t)})||_2^2$$

To prove the claim, we first observe that

$$||\nabla_i f(x^{(t)})||_2^2 \leq 2||\nabla_i f(x_{i-1}^{(t)}) - \nabla_i f(x^{(t)})||_2^2 + 2||\nabla_i f(x_{i-1}^{(t)})||_2^2$$

since $||a+b||^2 \leq 2||a||^2 + 2||b||^2$.

We now use the Lipschitz conditions to simplify to:

$$||\nabla_i f(x^{(t)})||_2^2 \leq 2L^2||x_{i-1}^{(t)} - x^{(t)}||^2 + 2||\nabla_i f(x_{i-1}^{(t)})||^2 = 2L^2||\sum_{j=1}^{i-1} \tfrac{1}{L} U_j \nabla_j f(x_{j-1}^{(t)})||^2 + 2||\nabla_i f(x_{i-1}^{(t)})||^2$$

Cancelling terms:

$$||\nabla_i f(x^{(t)})||_2^2 \leq 2||\sum_{j=1}^{b} \nabla_j f(x_{j-1}^{(t)})||^2 + 2||\nabla_i f(x_{i-1}^{(t)})||^2$$

Taking summation over $i = 1, \ldots, b$ and invoking the fact that

$$||\nabla f(x^{(t)})||_2^2 = \sum_{i=1}^{b} ||\nabla_i f(x^{(t)})||_2^2$$

we prove the claim.

This now gives us:

$$f(x^{(t)}) - f(x^{(t+1)}) \geq \tfrac{1}{4L(b+1)} ||\nabla f(x^{(t)})||_2^2$$

From the previous analysis of gradient descent, we arrive at the stated results. ■

Remark. If a larger stepsize $\gamma_i = \frac{1}{L_i}$ is adopted, the rate can be improved to

$$f(x^{(t)}) - f^* \leq \frac{4L_{max}(bL^2/L_{min}^2 + 1)R(x_0)^2}{t}.$$

where $L_{max} = \max_i L_i$ and $L_{min} = \min_i L_i$.

Summary. Here is brief summary of the converge rates for the three variants of block coordinate gradient descent algorithms:

Method	$\gamma_t = \frac{1}{L}$	$\gamma_t = \frac{1}{L_i}$
Gauss-Southwell	$\mathcal{O}(\frac{Lb}{t})$	$\mathcal{O}(\frac{L_{max}b}{t})$
Cyclic	$\mathcal{O}(\frac{Lb}{t})$	$\mathcal{O}(\frac{L_{max}(bL^2/L_{min}^2+1)}{t})$
Randomized	$\mathcal{O}(\frac{Lb}{t})$	$\mathcal{O}(\frac{\sum L_i}{t})$

Some open questions still remain about these complexity bounds. For example, are these bounds optimal, or under which situation can we drop the dependence on b, under which regime is cyclic RBGD better than randomized RBGD? In practice, their behaviors could vary a lot depending on the choice of step-sizes and the underlying parameters of problem instances.

References

[1] Nesterov, Yu , "Efficiency of coordinate descent methods on huge-scale optimization problems." SIAM Journal on Optimization 22.2 (2012): 341-362.

[2] Richtarik, Peter, and Martin Takac. "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function." Mathematical Programming 144.1-2 (2014): 1-38.

[3] Beck, Amir, and Luba Tetruashvili. "On the convergence of block coordinate descent type methods." SIAM Journal on Optimization 23.4 (2013): 2037-2060.

[4] Hong, Mingyi, Xiangfeng Wang, Meisam Razaviyayn, and Zhi-Quan Luo. "Iteration complexity analysis of block coordinate descent methods." arXiv preprint arXiv:1310.6957 (2013).

[5] Nutini, Julie, Mark Schmidt, Issam H. Laradji, Michael Friedlander, and Hoyt Koepke. "Coordinate descent converges faster with the Gauss-Southwell rule than random selection." Proceedings of the 32nd International Conference on Machine Learning (ICML-15). 2015.