Block Coordinate Minimization

Kun Yuan

November 19, 2024

In this lecture, we will discuss

- Block coordinate descent
- Block coordinate gradient descent
- Coordinate friendly stuctures

Most of the materials are from

H.-J. Shi, S. Tu, Y. Xu, and W. Yin, *A Primer on Coordinate Descent Algorithms*, arXiv:1610.00040, 2016.

Block coordinate descent

- Consider the following optimization problem

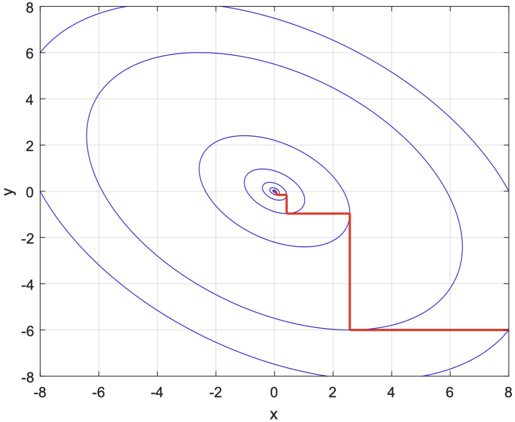
$$\min_{\mathbf{x}} \quad f(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_s)$$

where $\mathbf{x} \in \mathbb{R}^d$ is decomposed into s block variables $\mathbf{x}_1, \cdots, \mathbf{x}_s$.

Algorithm 1 Block Coordinate Descent

1: Set $k = 0$ and choose $\mathbf{x}^0 \in \mathbb{R}^n$;
2: **repeat**
3: Choose index $i_k \in \{1, 2, ..., s\}$;
4: Update $\mathbf{x}_{i_k}$ to $\mathbf{x}_{i_k}^k$ by a certain scheme depending on $\mathbf{x}^{k-1}$ and f ;
5: Keep $\mathbf{x}_j^k = \mathbf{x}_j^{k-1}$ for $j \neq i_k$;
6: Let $k = k + 1$
7: **until** termination condition is satisfied;

Block coordinate descent



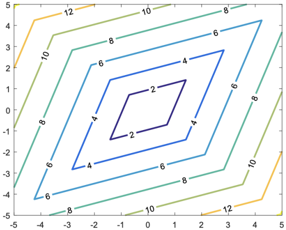
Update schemes

- Block coordinate minimization

$$\mathbf{x}_{i_k}^k = \arg\min_{\mathbf{x}_{i_k}} \; f(\mathbf{x}_{i_k}, \mathbf{x}_{\neq i_k}^{k-1})$$

- Most intuitive classic scheme. Guaranteed to converge for convex and differentiable problems.

- May not converge for non-convex or non-smooth problems:



Update schemes

- Block proximal point update

$$\mathbf{x}_{i_k}^k = \arg\min_{\mathbf{x}_{i_k}} \; f(\mathbf{x}_{i_k}, \mathbf{x}_{\neq i_k}^{k-1}) + \frac{1}{2\alpha_{i_k}^{k-1}} \|\mathbf{x}_{i_k} - \mathbf{x}_{i_k}^{k-1}\|_2^2$$

- Adds a quadratic proximal term to guarantee convergence

Update schemes

- Block proximal point update

$$\mathbf{x}_{i_k}^k = \arg\min_{\mathbf{x}_{i_k}} \; f(\mathbf{x}_{i_k}, \mathbf{x}_{\neq i_k}^{k-1}) + \frac{1}{2\alpha_{i_k}^{k-1}} \|\mathbf{x}_{i_k} - \mathbf{x}_{i_k}^{k-1}\|_2^2$$

- Block proximal linear update

$$\mathbf{x}_{i_k}^k = \arg\min_{\mathbf{x}_{i_k}} f(\mathbf{x}_{i_k}^{k-1}) + \langle \nabla_{i_k} f(\mathbf{x}_{i_k}^{k-1}, \mathbf{x}_{\neq i_k}^{k-1}), \mathbf{x}_{i_k} - \mathbf{x}_{i_k}^{k-1} \rangle + \frac{1}{2\alpha_{i_k}^{k-1}} \|\mathbf{x}_{i_k} - \mathbf{x}_{i_k}^{k-1}\|_2^2$$

- Or equivalently, **block gradient descent** algorithm

$$\mathbf{x}_{i_k}^k = \mathbf{x}_{i_k}^{k-1} - \alpha_{i_k}^{k-1} \nabla_{i_k} f(\mathbf{x}_{i_k}^{k-1}, \mathbf{x}_{\neq i_k}^{k-1})$$

Block stochastic gradient descent

- Consider the stochastic optimization problem:

$$\min_{x \in \mathbb{R}^d} \quad f(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\xi} \sim D}[F(\mathbf{x}_1, \cdots, \mathbf{x}_s; \boldsymbol{\xi})]$$

  - $\boldsymbol{\xi}$ is a random variable indicating data samples
  - D is the data distribution; unknown in advance
  - $\mathbf{x} \in \mathbb{R}^d$ is decomposed into s block variables $\mathbf{x}_1, \cdots, \mathbf{x}_s$

- Block stochastic gradient descent:

$$\mathbf{x}_{i_k}^k = \mathbf{x}_{i_k}^{k-1} - \alpha_{i_k}^{k-1} \mathbf{g}_{i_k}^{k-1} \quad \text{where} \quad \mathbf{g}_{i_k}^{k-1} = \nabla_{\mathbf{x}_{i_k}} \nabla F(\mathbf{x}^{k-1}; \boldsymbol{\xi}^{k-1})$$

where $\mathbf{g}_{i_k}^{k-1}$ is the block stochastic gradient descent at iteration $k-1$.

Choosing update index i_k

- Cyclic sampling

$$i_{k+1} = (k \mod s) + 1, \quad k \in \mathbb{N}.$$

Sometimes, we use a permutation of $\{1, \cdots, s\}$, which is called *shuffling*.

- Uniform sampling

$$P(i_k = j) = \frac{1}{s}, \quad j = 1, \ldots, s.$$

- Importance sampling

$$P(i_k = j) = p_\alpha(j) = \frac{L_j^\alpha}{\sum_{i=1}^{s} L_i^\alpha}, \quad j = 1, \ldots, s.$$

- Arbitrary sampling

$$P(i_k = j) = p_j, \quad j = 1, \ldots, s,$$

Choosing update index i_k

- Gauss-Southwell selection rule

$$i_k = \arg\max_{1 \le j \le s} \|\nabla_j f(\mathbf{x}^{k-1})\|$$

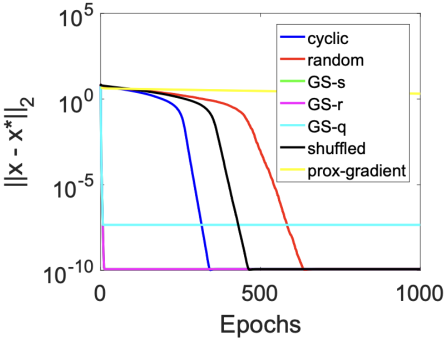
- Maximum block improvement rule

$$i_k = \arg\max_{1 \le j \le s} f(\mathbf{x}_j, \mathbf{x}_{\neq j}^{k-1})$$

These two rules are called the greedy rules; expensive to implement but can usually achieve much faster convergence, especially in sparse problems

Choosing update index i_k

The LASSO problem: $\min_{\mathbf{x}} \|\mathbf{x}\|_1 + \frac{\lambda}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2$



Convergence guarantees

Now we establish convergence guarantees for block coordinate gradient descent method with uniformly sampling.

Assumption (Component smoothness)

We assume for any x_i *and* y_i *that*

$$\begin{aligned}&\|\nabla_i f(\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_s) - \nabla_i f(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{y}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_s)\|_2 \\ &\le L_i \|\mathbf{x}_i - \mathbf{y}_i\|_2, \quad \forall i \in \{1, \cdots, s\}\end{aligned}$$

We define the coordinate Lipschitz constant $L_{\max}$ to be such that

$$L_{\max} = \max_{i=1,2,\cdots,s} L_i$$

Convergence guarantees

Theorem

Suppose $f(\mathbf{x})$ is component Lipschitz smooth, it holds that block coordinate gradient descent method with uniform sampling converges at the following rate:

$$\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}\|\nabla f(\mathbf{x}^k)\|^2 \leq \frac{2sL_{\max}D_0}{K+1}$$

where $D_0 = \mathbb{E}[f(\mathbf{x}^0)] - f(\mathbf{x}^\star)$.

Convergence guarantees

Proof: Since $f(\mathbf{x})$ is component Lipschitz smooth, we have

$$\begin{aligned}
f(\mathbf{x}^{k+1}) &= f(\mathbf{x}^k - \alpha_k [\nabla_{i_k} f(\mathbf{x}^k)]\mathbf{e}_{i_k}) \\
&\leq f(\mathbf{x}^k) - \alpha_k \|\nabla_{i_k} f(\mathbf{x}^k)\|^2 + \frac{1}{2}\alpha_k^2 L_{i_k} \|\nabla_{i_k} f(\mathbf{x}^k)\|^2 \\
&\leq f(\mathbf{x}^k) - \alpha_k \left(1 - \frac{L_{\max}}{2}\alpha_k\right) \|\nabla_{i_k} f(\mathbf{x}^k)\|^2 \\
&= f(\mathbf{x}^k) - \frac{1}{2L_{\max}} \|\nabla_{i_k} f(\mathbf{x}^k)\|^2. \qquad (\alpha_k = 1/L_{\max})
\end{aligned}$$

Taking expectations over the random block index i_k, we have

$$\begin{aligned}
\mathbb{E}_{i_k}[f(\mathbf{x}^{k+1})] &\leq f(\mathbf{x}^k) - \frac{1}{2sL_{\max}} \sum_{i=1}^{s} \|\nabla_i f(\mathbf{x}^k)\|^2 \\
&= f(\mathbf{x}^k) - \frac{1}{2sL_{\max}} \|\nabla f(\mathbf{x}^k)\|^2
\end{aligned}$$

Convergence guarantees

Taking expectations over the all random variables $\mathbf{x}^k$, we have

$$\mathbb{E}\|\nabla f(\mathbf{x}^k)\|^2 \leq 2sL_{\max}(\mathbb{E}[f(\mathbf{x}^k)] - \mathbb{E}[f(\mathbf{x}^{k+1})])$$

Summing up the above inequality over $k = 0, 1, \cdots, K$, we have

$$\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}\|\nabla f(\mathbf{x}^k)\|^2 \leq \frac{2sL_{\max}D_0}{K+1}$$

where $D_0 = \mathbb{E}[f(\mathbf{x}^0)] - f(\mathbf{x}^\star)$.

Coordinate friendly structures

- Let $T : \mathbb{R}^n \to \mathbb{R}^n$ represent an update mapping

$$\mathbf{x}^k = T(\mathbf{x}^{k-1}).$$

- let T_i denote the coordinate update mapping of T for block $\mathbf{x}_i$, i.e.,

$$T_i(\mathbf{x}) = (T(\mathbf{x}))_i, \quad i = 1, \ldots, s.$$

- Let $\mathcal{N}[a \to b]$ denote the number of basic operations necessary to compute quantity b from a.

- Coordinate friendly structure

$$\mathcal{N}[\mathbf{x} \mapsto T_i(\mathbf{x})] = O\left(\frac{1}{s}\mathcal{N}[\mathbf{x} \mapsto T(\mathbf{x})]\right), \forall i$$

Coordinate friendly structures

- Consider the LASSO problem:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \ \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

- The full gradient descent mapping is:

$$\mathbf{x}^k = T_{\mathrm{GD}}(\mathbf{x}^{k-1}) = \mathbf{x}^{k-1} - \alpha(\mathbf{A}^T\mathbf{A}\mathbf{x}^{k-1} - \mathbf{A}^T\mathbf{b})$$

- The coordinate gradient descent mapping is:

$$T_{\mathrm{GD},i}(\mathbf{x}^{k-1}) = x_i^{k-1} - \alpha(\mathbf{A}^T\mathbf{A}\mathbf{x}^{k-1} - \mathbf{A}^T\mathbf{b})_i, \quad i = 1, \ldots, s.$$

- Coordinate friendly structure

$$T_{\mathrm{GD},i}(\mathbf{x}^{k-1}) = \mathbf{x}_i^{k-1} - \alpha[((\mathbf{A}^T\mathbf{A})_{i,:}\mathbf{x}^{k-1} - (\mathbf{A}^T\mathbf{b})_i)]$$

which takes $O(n^2/s)$ operations after precomputing $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}^T\mathbf{b}$.

Coordinate friendly structures

- Consider logistic regression:

$$\text{minimize}_{\mathbf{w}} F(\mathbf{w}) = \sum_{j=1}^{m} \log(1 + \exp[-y_j \mathbf{w}^T \mathbf{x}_j])$$

- The full gradient descent mapping is:

$$\nabla F(\mathbf{w}) = \sum_{j=1}^{m} \frac{-y_j \exp(-y_j \mathbf{w}^T \mathbf{x}_j)}{1 + \exp(-y_j \mathbf{w}^T \mathbf{x}_j)} \mathbf{x}_j,$$

- Define $\mathcal{M}(\mathbf{w}^{k-1}) = \{\exp[-y_j (\mathbf{w}^{k-1})^T \mathbf{x}_j], j = 1, \ldots, m\}$. The coordinate gradient descent mapping is:

$$T_{\mathrm{GD},i}(\mathbf{w}^{k-1}) = \mathbf{w}_i^{k-1} - \alpha \sum_{j=1}^{m} \frac{-y_j \mathcal{M}(\mathbf{w}^{k-1})(\mathbf{x}_j)_i}{1 + \mathcal{M}(\mathbf{w}^{k-1})(\mathbf{x}_j)_i},$$

which takes $O(\frac{mn}{s})$ because computing $\exp[-y_j (\mathbf{w}^{k-1})^T \mathbf{x}_j]$ is avoided.

Coordinate friendly structures

- Consider logistic regression:

$$\min_{\mathbf{w}}\ F(\mathbf{w}) = \sum_{j=1}^{m} \log(1 + \exp[-y_j \mathbf{w}^T \mathbf{x}_j])$$

- The full gradient descent mapping is:

$$\nabla F(\mathbf{w}) = \sum_{j=1}^{m} \frac{-y_j \exp(-y_j \mathbf{w}^T \mathbf{x}_j)}{1 + \exp(-y_j \mathbf{w}^T \mathbf{x}_j)} \mathbf{x}_j,$$

- Define $\mathcal{M}(\mathbf{w}^{k-1}) = \{\exp[-y_j (\mathbf{w}^{k-1})^T \mathbf{x}_j], j = 1, \ldots, m\}$. The coordinate gradient descent mapping is:

$$T_{\mathrm{GD},i}(\mathbf{w}^{k-1}) = \mathbf{w}_i^{k-1} - \alpha \sum_{j=1}^{m} \frac{-y_j \mathcal{M}(\mathbf{w}^{k-1})(\mathbf{x}_j)_i}{1 + \mathcal{M}(\mathbf{w}^{k-1})(\mathbf{x}_j)_i},$$

which takes $O(\frac{mn}{s})$ because computing $\exp[-y_j (\mathbf{w}^{k-1})^T \mathbf{x}_j]$ is avoided.

Coordinate friendly structures

- Obtain $\mathcal{M}(\mathbf{w}^k)$ from $\mathcal{M}(\mathbf{w}^{k-1})$ takes $O(\frac{mn}{s})$ operations

$$\begin{aligned} & \exp\left[-y_j(\mathbf{w}^k)^T\mathbf{x}_j\right] \\ = \; & \exp\left[-y_j(\mathbf{w}^{k-1})^T\mathbf{x}_j\right] \cdot \exp\left[-y_j(\mathbf{w}^k_{i_k} - \mathbf{w}^{k-1}_{i_k})^T(\mathbf{x}_j)_{i_k}\right], \forall j. \end{aligned}$$