Coordinate Descent

Yilan Chen
Department of Computer Science and Engineering
University of California, San Diego
yilan@ucsd.edu

1 Introduction

The problem

$$\min_x f(x)$$

where $f(x) = g(x) + \sum_{i=1}^{n} h_i(x)$, with g convex and differentiable and each h_i convex, can be sloved by coordinate descent: let $x^{(0)} \in \mathbb{R}^n$, and repeat

$$x_i^{(k)} = \arg\min_{x_i} f(x_1^{(k)}, ..., x_{i-1}^{(k)}, x_i, x_{i+1}^{(k-1)}, ..., x_n^{(k-1)}), i = 1, ..., n \text{ and } k = 1, 2, 3, ...$$

In other words, we minimize $f(x)$ with respect to one element x_i, plug it back in f, and move to the next index. We always use the most recent information possible.

Tseng [2001] showed that for such f (provided f is continuous on a compact set $\{x : f(x) \leq f(x^{(0)})\}$ and f attains its minimum), any limit point of $x^{(k)}, k = 1, 2, 3, ...$ is a minimizer of f.

2 Coordinate Descent Method

We followed the basic ideas of classical coordinate descent [Wright, 2015], while every cycle we choose the parameter with the largest deravative to update by gradient descent. Algorithm 1 shows the process.

(a) Which coordinate to choose?
We calculated the deravative of the loss function with respect to the parameters and choose the parameter with largest deravative to update.

(b) How to set the new value of w_i?
We calculated the deravative of the loss function with respect to this parameter w_i and update it with gradient descent until this parameter converge.

This method needs to the loss function $L(\cdot)$ to be differentiable because we need to calculate the deravative to update the parameters. But in general, we may not necessarily need $L(\cdot)$ to be fully differentiable. As long as $L(\cdot)$ can be separated into two parts: one part is convex and differentiable and the other part is convex, the algorithm will converge. And we can explore this property as long as part of $L(\cdot)$ is differentiable. For the gradient, we can use subgradient instead.

3 Convergence

As proved in [Tseng, 2001, Wright, 2015], this kind of coordinate descent methods are guaranteed to converge as long as the loss function can be separated into two parts: one part is convex and differentiable and the other part is convex (provided f is continuous on a compact set $\{x : f(x) \leq f(x^{(0)})\}$ and f attains its minimum). The order of cycle through coordinates wouldn't affect the convergence. We choose the coordinate with largest gradient to update in order to converge faster.

Algorithm 1 Coordinate Descent

set $w = 0$
repeat
  $i = \text{argmax } \nabla L(w)$
  repeat
    $w_i = w_i - \eta[\nabla L(w)]_i$
  until w_i converge
until w converge

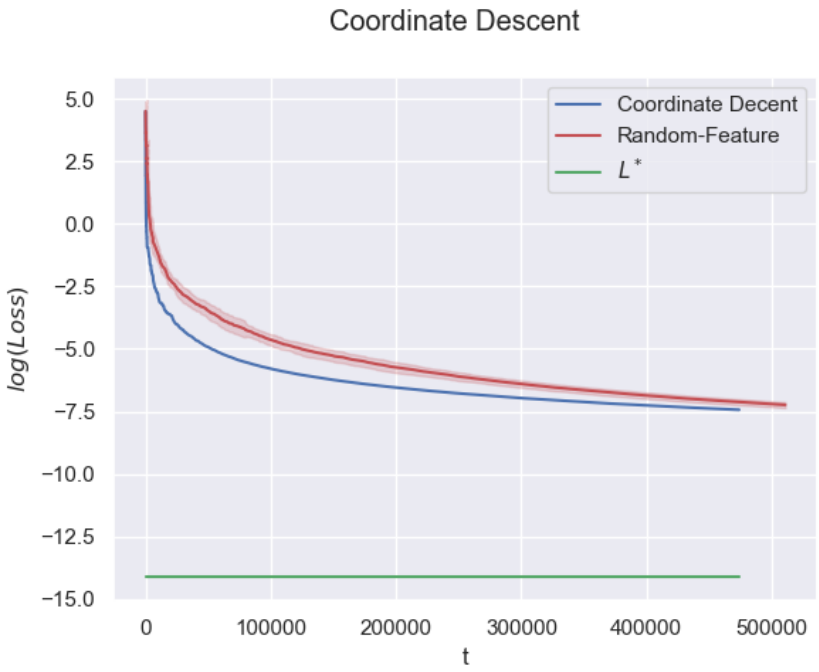

Figure 1: Coordinate descent. Ploted in logarithmic scale for better comparation.

Since in this problem, the loss function is logistic loss, which is convex and differentiable, the algorithm will converge.

4 Experimental Results

Figure 1 shows the loss curves of the logistic regression with wine dataset of UCI dataset. Random-feature coordinate descent chooses coordinates i uniformly at random to update every time. For random-feature coordinate descent, we did 10 experiments and ploted its mean and standard deviation as shaded area. The coordinate descent converge faster then random-feature coordinate descent. Both the coordinate descent and random-feature coordinate descent converge asymptotically to L^*. Note for clear comparation, we plot the curves in logarithmic scale and all losses converge to very small values.

5 Critical Evaluation

This coordinate descent scheme only use the first-order information. If the loss function has continuous second-order derivatives, we may explore more information to decide which coordinate to update. For faster convergence, we may extent to block-CD algorithms in a straightforward way, by updating a block of coordinates rather than a single coordinate.

In the case that the loss function is not differentiable, we may adopt other methods to update the parameters. For example, using subgradient instead of gradient.

From the view of computation, the accelerated coordinate descent algorithms and parallel variants can be explored for certain problem structures [Wright, 2015].

References

P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

S. J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.