Block Successive Upper Bound Minimization (BSUM) Project

December 15, 2017

1 Project Description

Before starting this project, let us introduce **Gradient Descent (GD)**, which is a first-order iterative optimization algorithm for finding local minimum of a function f:

$$\min_{x\in\Re^n} f(x), \tag{1.1}$$

where f is considered as convex and smooth function, i.e., f is continuously differentiable.

In GD algorithm, to find a local minimum of a function f from current point, it takes step proportional to the negative of the gradient. However, when n is large, it becomes computationally expensive to compute full gradients, i.e., gradient descent calculation is not certainly always efficient. To find the optimal solution, it better to search along each coordinate direction until the objective function is not decreasing at certain coordinate direction, then it has reached the optimum points. This coordinate based algorithm belongs to family of algorithms called Coordinate Minimization algorithms, or also called **Coordinate Descent algorithms** [1]. The general idea for a coordinate descent algorithm is shown below:

Algorithm 1 : Coordinate descent algorithm

1: Initialize $x^{(0)}$, $t = 0, \epsilon > 0$;
2: **repeat**
3: Pick coordinate i from $1, 2, ...n$;
4: Set $x_i^{(t+1)} = \underset{x_i\in\Re}{\operatorname{argmin}} f(x_i, w_{-i}^{(t)})$;
5: $t = t + 1$
6: **until** $\| \frac{f(x^{(t)}) - f(x^{(t+1)})}{f(x^{(t)})} \| \leq \epsilon$

where $w_{-i}^{(t)} = (x_1^{(t+1)}, \ldots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \ldots, x_n^{(t)})$ in case of Gauss−Seidel style or $w_{-i}^{(t)} = (x_1^{(t)}, \ldots, x_{i-1}^{(t)}, x_{i+1}^{(t)}, \ldots, x_n^{(t)})$ in case of Jacobi style.

A well-know method for minimizing real-valued continuously differentiable function f of n real variables, subject to bound constraints, is so called **Block Coordinate Descent (BCD)** method. In BCD, at each iteration, the coordinates are partitioned into N blocks, where f is minimized with respect to one of the coordinate blocks while keeping the other coordinates fixed [2]. However, if f is a non-convex function, and BCD doesn't always guarantee convergence.

To overcome the above issue, **Block Successive Upper Bound Minimization (BSUM)** algorithm [3] [4] can be utilized, where BSUM is considered as a generalized form of BCD that optimizes block by block upper-bound function of the original objective function. BSUM can be used for solving separable smooth or non-smooth convex optimization problems that have linear coupling constraints. To solve the family of these

problems, BSUM updates each block of variables iteratively through minimizing upper-bound function until it converges to stationary points (solutions).

Problem to solve: *Consider the following optimization problem:*

$$\begin{aligned} &\min \ f(x) \\ &\text{subject to } \ x \in \mathscr{X}, \end{aligned} \tag{1.2}$$

where $f : \Re^n \to \Re$ is a given convex function f (you are free to choose any function [see, e.g., [5]]), and $\mathscr{X}$ is nonempty closed convex set.

We introduce an additional vector $y \in \Re^n$, and consider the following equivalent optimization problem:

$$\begin{aligned} &\min \ f(x) + \frac{1}{2\alpha}\|(x-y)\|_2^2. \\ &\text{subject to } \ x \in \mathscr{X}, y \in \Re^n \end{aligned} \tag{1.3}$$

where α is a positive scalar parameter, $\|\|_2$ is the standard Euclidean distance ([see [6], Sec. 3.4.3, page 53]).

1. Generate some data and try out BSUM technique through minimizing the cost function f (1.3) over x while keeping y fixed, then minimize the cost function f over y while keeping x fixed.
2. Use Cyclic, Gauss-Southwell (G-So), and randomized coordinate selection rules ((see, e.g., [[7], Sec. 13.3]) and compare the results (plot them).
3. Illustrate how the BSUM converge, and how the parameter α affects the convergence rate.
4. Upload Python/Julia code and report to the piazza system.

REFERENCES

[1] ise.illinois.edu, "Coordinate descent algorithms," http://niaohe.ise.illinois.edu/IE598/pdf/IE598-lecture12-coordinate%20descent%20algorithms.pdf, [Online; accessed December 11, 2017].

[2] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of optimization theory and applications*, vol. 109, no. 3, pp. 475–494, 2001.

[3] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[4] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, "A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing," *IEEE Signal Processing Magazine*, vol. 33, no. 1, pp. 57–77, 2016.

[5] N. A. Lorenzo Stella, "Proximal operators," https://kul-forbes.github.io/ProximalOperators.jl/latest/functions.html#Norms-and-regularization-functions-1, [Online; accessed December 11, 2017].

[6] T. J. N. Bertsekas, Dimitri P., "Parallel and distributed computation:numerical methods," https://dspace.mit.edu/bitstream/handle/1721.1/3719/part3.pdf?sequence=6, [Online; accessed December 11, 2017].

[7] ise.illinois.edu, "Case study on logistic regression," http://niaohe.ise.illinois.edu/IE598/pdf/IE598-lecture13-case%20study%20on%20logistic%20regression.pdf, [Online; accessed December 11, 2017].