Term paper: Incremental Majorization-Minimization for large scale Machine Learning (Mixed Mode)

Aarsh Prakash Agarwal (150004)
Electrical Engineering Department, IIT Kanpur
aarshp@iitk.ac.in

Shivam Utreja (150682)
Electrical Engineering Department, IIT Kanpur
sutreja@iitk.ac.in

Abstract

One of the crucial problems in machine learning is optimizing sum of functions. Majorization-minimization(MM) is a very popular algorithm and often used due to its convergence guarantees even in non-convex cases (with the some assumptions). However, its scalability is still an issue. With increasing data, several stochastic optimization algorithms such as [9] [7], SGD [2], etc have been developed to deal with the scale. But, most of algorithms suffer from slow convergence. We therefore, present a scalable version of MM, particularly focussing on **"MISO" algorithm [8] by J. Marial** and analyze its convergence. We also compare it with state of the art algorithms and present our insights. The text highlighted in blue are our insights and interpretations

1 Introduction

The majorization-minimization (MM) algorithms can be described as optimization algorithms that minimize the objective by creating a function which upper bounds the objective and subsequently minimizing the created upper bound. These two steps are performed in the alternating fashion until the convergence is achieved. The function which is used to upper bound the objective is called *surrogate function*. MM algorithm therefore, isn't actually a particular algorithm but a schema or a class of algorithms. Some of the popular algorithms that can be derived from this schema, by choosing the appropriate surrogate function are EM (expectation-maximization), multidimensional-scaling, image reconstruction, difference of convex programming, etc. The problem we are targeting can be summarized as minimizing a large sum of functions, i.e.,

$$\min_{\theta\in\Theta}\left[f(\theta) \triangleq \frac{1}{T}\sum_{t=1}^{T} f^t(\theta)\right] \tag{1}$$

where Θ is a convex subset of $\mathbb{R}^p$ and each $f^t : \mathbb{R}^p \to \mathbb{R}$ is continuous. Scale of problem is determined by T. T can also be thought as number of training points in a typical machine learning setting – θ being the model parameters and $f^t(\theta)$ being the t^{th} datapoint derived from the parameters. In order to tackle the very large scale problem, online algorithms have been developed which essentially deals with small portion of dataset/ single point in a dataset per iteration. This stochastic approach results in very cheap computation per iteration and a sublinear rate of convergence if objective is convex w.r.t θ.

The author proposes an alternative algorithm based on MM to deal with scale. The proposed "MISO" algorithm (Minimization by Incremental Surrogate Optimization) is shown to have cheap computations per iteration (similar to stochastic/online approach) along with the linear convergence rate if the surrogate function is convex/strongly convex. However, one drawback of MISO is it's high memory cost. The algorithm is discussed in detail in section 2. The theoretical convergence of the algorithm is discussed in section 3 in cases where the surrogate functions are approximated by first order function. The details of practical implementations are discussed in section 4.

2 Algorithm

In this section we will first look at the basic MM algorithm with first order surrogate functions and then at the MISO which takes the advantage of inherent structure of the problem in which f is a function summation.

2.1 MM algorithm

The basic MM algorithm for objective $\arg\min_{\theta\in\Theta} f(\theta)$ is described in Algorithm 1. The surrogate function g_n upper

Algorithm 1: MM algorithm

Input: Initialize $\theta_0 \in \Theta$, Number of iterations N

for $i = 1, 2, \ldots . N$ **do**

  Compute the surrogate function g_n at θ_{n-1} of f

  Minimize g_n and Update θ_n as:

$$\theta_n = \text{argmin}_{\theta\in\Theta} g_n(\theta)$$

end for

Output: Optimal θ_N at the end of iterations

bounds or majorizes f at θ_{n-1} which implies that the error in approximation $h_n = g_n - f \geq 0$ for all $\theta \in \Theta$ and $h_n(\theta_{n-1}) = 0$.

2.2 MISO (Minimization by Incremental Surrogate Optimization)

Next up, we present the algorithm proposed by [8], which utilizes the fact that f is a sum of T components. Here, at

each iteration we randomly choose a function $f^{\hat{t}_n}$ among T functions and update it's surrogate function $g_n^{\hat{t}_n}$. Rest of surrogate functions are kept the same and sum of surrogate functions is minimized. Intuitively, it feels similar to an online/stochastic process, where a random training point is chosen among all the points, and the updates are based on that point. Formally, algorithm is described in 2:

Algorithm 2: MISO algorithm

Input: Initialize $\theta_0 \in \Theta$, Number of iterations N

Initialize surrogate functions g_0^t of f^t at θ_0 $\forall t = 1, 2...T$

for $n = 1, 2, ...N$ **do**

  Randomly choose $\hat{t}_n$ from $\{t\}_{t=1}^N$

  Compute the surrogate function $g_n^{\hat{t}_n}$ at θ_{n-1} of $f^{\hat{t}_n}$ for the chosen $\hat{t}_n$

  Update the rest as $g_n^t = g_{n-1}^t \quad \forall t \neq \hat{t}_n$

  Update θ_n as:

$$\theta_n \leftarrow \arg\min_{\theta \in \Theta} \left[\bar{g}_n(\theta) = \frac{1}{T} \sum_{t=1}^{N} g_n^t(\theta) \right]$$

end for

Output: Optimal θ_N at the end of iterations

3 Theoretical convergence

Here, we will show theoretical convergence for both the convex and non-convex cases and few special cases for the proposed algorithm MISO [2].

For all the cases and proofs, we would assume the first order surrogate function [6]. In general, function g is the first order surrogate of function $f : \mathbb{R}^p \to \mathbb{R}$ at point κ if it satisfies the properties of surrogate function and the approximation error $h = g - f$ is L-smooth; i.e., $h > 0 \quad \forall \theta \in \Theta$ and, $h(\kappa) = 0$ as well as gradient of h is L-Lipschitz continuous which would imply $\nabla h(\kappa) = 0$. Set of all first order surrogate function are denoted by $\mathscr{S}_L(f, \kappa)$ and set of all first order surrogate function which are $\rho-$ strongly convex are represented by $\mathscr{S}_{L,\rho}(f, \kappa)$. Clearly, $\mathscr{S}_{L,\rho}(f, \kappa) \subset \mathscr{S}_L(f, \kappa)$.

First order surrogate function follow few properties which are mentioned below:

$$|h(\theta)| \leq \frac{L}{2}\|\theta - \kappa\|_2^2 \tag{2}$$

$$f(\theta') \leq f(\theta) + \frac{L}{2}\|\theta - \kappa\|_2^2 \tag{3}$$

$$f(\theta') + \frac{\rho}{2}\|\theta - \theta'\|_2^2 + \leq f(\theta) + \frac{L}{2}\|\theta - \kappa\|_2^2 \tag{4}$$

where, θ' minimizes the surrogate function g and the last property [4] only holds if the function $g \in \mathscr{S}_{L,\rho}(f, \kappa)$. With first order surrogates properly defined, the convergence analysis can now be done. Let's start with convergence analysis for non-convex problems

3.1 Convergence Analysis for Non-convex Problems

In general, non-convex problems cannot achieve a global minima. Therefore, we would aim for *asymptotic stationary point* (ASP) instead of a global minima. Also, for non-convex objective f, we would work under the assumptions that;

- f is bounded below,
- the directional derivative of $f^t \quad \forall t = 1, 2, ...T$, represented by $\nabla f^t(\theta, \theta - \theta')$ exists in the direction $(\theta - \theta')$ $\forall \theta, \theta' \in \Theta$.

The first assumption makes sense as, otherwise any algorithm can never achieve local minima even in infinite iterations. Also, if the f^t is differentiable, directional derivative $\nabla f^t(\theta, \theta - \theta') = \nabla f^t(\theta)^\top(\theta - \theta')$. With this mild assumption, we can define ASP in terms of sequence $\{\theta_n\}$ which follows:

$$\lim_{n \to \infty} \inf_{\theta in \Theta} \frac{\nabla f(\theta_n, \theta - \theta_n)}{\|\theta - \theta_n\|_2} > 0 \tag{5}$$

which indicates that if f is differentiable, it's derivative $\nabla f(\theta_n)$ converges to zero, which is what happens at local minima. With above assumptions and definitions, two propositions can be proposed as:

3.1.1 Proposition-1: If the surrogate function $g_n^{\hat{t}_n} \in \mathscr{S}_L(f^{\hat{t}_n}, \theta_{n-1})$ and majorizes $f^{\hat{t}_n}$ at θ_{n-1}, then sequence $\{f(\theta_n)\}$ decreases monotonically and $\{\theta_n\}$ achieves ASP.

For first part, define $\bar{g}_n = \frac{1}{T}\sum_{t=1}^T g_n^t$. Then, following recursive relation will hold,

$$\bar{g}_n = \bar{g}_{n-1} + \frac{g_n^{\hat{t}_n} - g_n^{\hat{t}_n}}{T}$$

We have $\bar{g}_n(\theta_n) \leq \bar{g}_n(\theta_{n-1})$ as θ_n is the minima for $\bar{g}_n$. Therefore,

$$\Rightarrow \bar{g}_n(\theta_n) \leq \bar{g}_n(\theta_{n-1}) = \bar{g}_{n-1}(\theta_{n-1}) + \frac{g_n^{\hat{t}_n}(\theta_{n-1}) - g_n^{\hat{t}_n}(\theta_{n-1})}{T}$$

Now, the expression $g_n^{\hat{t}_n}(\theta_{n-1}) - g_{n-1}^{\hat{t}_n}(\theta_{n-1}) = f^{\hat{t}_n}(\theta_{n-1}) - g_{n-1}^{\hat{t}_n}(\theta_{n-1}) \leq 0$, as $g_n^{\hat{t}_n}$ surrogates $f^{\hat{t}_n}$ at θ_{n-1}. Therefore we have $\bar{g}_n(\theta_n) \leq \bar{g}_{n-1}(\theta_{n-1})$. Therefore, we have $\{\bar{g}_n(\theta_n)\}$ as a decreasing sequence which will converge. Taking the expectation of summation of $g_n^{\hat{t}_n}(\theta_{n-1}) - f^{\hat{t}_n}(\theta_{n-1})$ we get,

$$\mathbb{E}\left[\sum_{n=1}^{\infty} g_n^{\hat{t}_n}(\theta_{n-1}) - f^{\hat{t}_n}(\theta_{n-1})\right] = \sum_{n=1}^{\infty} \mathbb{E}\left[g_n^{\hat{t}_n}(\theta_{n-1}) - f^{\hat{t}_n}(\theta_{n-1})\right]$$

$$= \sum_{n=1}^{\infty} \mathbb{E}\left[\mathbb{E}\left[g_n^{\hat{t}_n}(\theta_{n-1}) - f^{\hat{t}_n}(\theta_{n-1})|\mathscr{F}_n\right]\right]$$

$$= \sum_{n=0}^{\infty} \mathbb{E}\left[\bar{g}_n(\theta_n) - f(\theta_n)\right] = \mathbb{E}\left[\sum_{n=0}^{\infty} \bar{g}_n(\theta_n) - f(\theta_n)\right] \leq \infty$$

In above equations, $\mathscr{F}_n$ represents all the information summing upto θ_n. From above we have $\{\bar{g}_n(\theta_n) - f(\theta_n)\}_{n\geq 0}$ as a converging sequence, and we already have $\{\bar{g}_n(\theta_n)\}_{n\geq 0}$ monotonically decreasing and converging. Therefore, $\{f_n(\theta_n)\}_{n\geq 0}$ converges and decreases monotonically.

Now to show $\{\theta_n\}_{n\geq 0}$ achieves ASP; we would first show that for the approximation error $\bar{h}_n = \bar{g}_n - f$, the derivative of it's norm, $\nabla\|\bar{h}_n(\theta_n)\|_2 \to 0$. Note that,we have already shown $\{h_n(\theta_n)\}$ as a converging sequence. Take $\theta' = \theta_n - \frac{1}{L}\nabla\bar{h}_n(\theta_n)$, and since $\bar{h}_n(\theta') > 0$, the following inequality holds from definition of L-smooth:

$$0 \leq \bar{h}_n(\theta') \leq \bar{h}_n(\theta_n) - \frac{1}{2L}\nabla\|\bar{h}_n(\theta_n)\|_2^2$$

$$\implies \nabla\|\bar{h}_n(\theta_n)\|_2^2 \leq 2L(\bar{h}_n(\theta_n) - \bar{h}_n(\theta')) \leq 2L\bar{h}_n(\theta_n) \to 0$$

Since $\nabla\bar{h}_n(\theta_n)$ exists, the directional derivative of f can be written as

$$\nabla f(\theta_n, \theta - \theta_n) = \nabla\bar{g}_n(\theta_n, \theta - \theta_n) - \nabla\bar{h}_n(\theta_n)^\top(\theta - \theta_n)$$

Referring to the Step 6 of algorithm 2, we are minimizing $\bar{g}_n$ at each iteration. Therefore $\nabla\bar{g}_n(\theta_n, \theta - \theta_n) \geq 0$, and this would imply, $\nabla f(\theta_n, \theta - \theta_n) \geq -\nabla\bar{h}_n(\theta_n)^\top(\theta - \theta_n)$. From Cauchy-Schwarz, we have $\nabla f(\theta_n, \theta - \theta_n) \geq -\|\nabla\bar{h}_n(\theta_n)\|_2\|(\theta - \theta_n)\|_2$. Rearranging and taking infimum we would get,

$$\lim_{n\to\infty}\inf_{\theta in\Theta}\frac{\nabla f(\theta_n, \theta - \theta_n)}{\|\theta - \theta_n\|_2} \geq -\lim_{n\to\infty}\|\nabla\bar{h}_n(\theta_n)\|_2 = 0$$

This proposition guarantees a convergence at a local minima even for non convex functions with mild reasonable assumptions. Also, *the proposition holds even if f is non-smooth.* Although, the conditions of f seem mild, but the conditions on surrogate function are harsh as it is quite difficult of find a surrogate function which will produce L-smooth error especially for non-smooth f. Clearly, the choice of surrogate function affects the overall convergence and discussion on surrogate function has been limited to the case of unconstrained optimization. Also, as pointed out by [1], no examples of surrogate function for constrained optimization have been provided

Let's relax the conditions of above proposition by defining composite functions as $f = f' \circ e$, which would mean $f(\theta) = f'(e(\theta))$. Here, e and f' are defined as $e : \mathbb{R}^p \to \mathbb{R}^d$ and $f' : \mathbb{R}^d \to \mathbb{R}$. In the context of MISO Algorithm 2, the f^t would be $f^t = f'^t \circ e^t$. Similarly, surrogates $g^{\hat{t}_n}$ would also be $g^{\hat{t}_n} = g'^{\hat{t}_n} \circ e^{\hat{t}_n}$. Thus, a second proposition can be proposed as follows:

3.1.2 Proposition-2: If f^t are composed as $f^t = f'^t \circ e^t$ and surrogates $g^{\hat{t}_n}$ as $g^{\hat{t}_n} = g'^{\hat{t}_n} \circ e^{\hat{t}_n}$, where surrogate functions $g'^{\hat{t}_n} \in \mathscr{S}_L(f'^{\hat{t}_n}, e^{\hat{t}_n}(\theta_{n-1}))$ and majorizes $f'^{\hat{t}_n}$. Assuming that functions e^t are C-Lipschitz continuous, then sequence $\{f(\theta_n)\}$ decreases monotonically and $\{\theta_n\}$ achieves ASP.

Proof of the first will follow a similar line of arguments as in proposition 3.1.1. Thus we have converging and monotonically decreasing sequences $\{f(\theta_n)\}$ and $\{\bar{g}_n(\theta_n) - f(\theta_n)\}$, where $\bar{g_n}$ and $\bar{h}_n$ are defined as above. Additionally, we will have $\bar{h}_n^t = \bar{g}_n^t - f^t$. Also, corresponding to surrogates $g'^{\hat{t}_n}$ of $f'^{\hat{t}_n}$ we have $\bar{h'}_n^t$ and $\bar{g'}_n^t$ similarly defined. Composite relation also holds for approximation error as $\bar{h}_n^t = \bar{h}_n'^t \circ e^t$

Again since $g'^{\hat{t}_n}$ majorizes $f'^{\hat{t}_n}$, following the steps in previous proposition will show that $\|\nabla\bar{h}_n'^t(e^t(\theta_n))\|_2 \to 0$. For $\|\bar{h}_n^t\|_2$, assume some δ such that $\theta_n + \delta \in \Theta$. Now,

$$\begin{aligned}\bar{h}_n^t(\theta_n + \delta) &= \bar{h}_n'^t(e^t(\theta_n + \delta)) \\ &= \bar{h}_n^t(\theta_n) + \|\delta\|_2\nabla\bar{h}_n'^t(e^t(\theta_n))^\top\boldsymbol{z} + \mathscr{O}(k\|\delta\|_2)\end{aligned}$$

where $\boldsymbol{z} : \|\boldsymbol{z}\|_2 < C$ and second equality is because $\bar{h}_n'^t$ is L-smooth. Choosing $\delta = \alpha(\theta - \theta_n)$ where $0 < \alpha < 1$ and rearranging the above equation, would result the directional derivative to be

$$|\nabla\bar{h}_n^t(\theta_n, \theta - \theta_n)| \leq C\|\nabla\bar{h}_n'^t(e^t(\theta_n))\|_2\|\theta - \theta_n\|_2$$

The directional derivative of f can be written as:

$$\nabla f(\theta_n, \theta - \theta_n) = \nabla\bar{g}_n(\theta_n, \theta - \theta_n) - \frac{1}{T}\nabla\bar{h}_n^t(\theta_n, \theta - \theta_n)$$

And following the lines of proof in proposition 3.1.1, $\nabla\bar{g}_n(\theta_n, \theta - \theta_n) > 0$, therefore, resulting in

$$\lim_{n\to\infty}\inf_{\theta in\Theta}\frac{\nabla f(\theta_n, \theta - \theta_n)}{\|\theta - \theta_n\|_2} \geq -C\lim_{n\to\infty}\|\nabla\bar{h}_n'^t(e^t(\theta_n))\|_2 = 0$$

For the convergence of composite functions, e^t have to be smooth and that is non negotiable. Composite functions are generally seen when we transform domain to some Hilbert space. In many problems we simultaneously learn the space we are projecting, and therefore, for convergence it is difficult to keep a check on whether the projected space is C smooth or not. The next section would provide proofs for the convergence rates for Convex objectives.

3.2 Convergence analysis for Convex Objectives

Since, in convex objectives, the global minima exists, we will have much stronger results for convergence than the non-convex cases. Assume that f is lower bounded, and achieves minima f^* at θ^*, i.e., $f^* = \min_{\theta\in\Theta} f(\theta)$. The following proposition can be proposed:

3.2.1 Proposition 3: g_n^t majorizes f^t at iteration n and $g_n^t \in \mathscr{S}_{L,\rho}(f^t, \theta_{n-1})$ are $\rho-$ strongly convex with $\rho \geq L$ then following holds

$$\mathbb{E}\big[f(\bar{\theta}_n) - f^*\big] \leq \frac{LT\|\theta_0 - \theta^*\|_2^2}{2n} \tag{6}$$

where $\bar{\theta}_n = \sum_{i=1}^{n} \theta_i$ *is running average of iterates. If* f *is* $\mu-$*strongly convex, then condition becomes:*

$$\mathbb{E}\big[f(\theta_n) - f^*\big] \leq \left(1 - \frac{2\mu}{T(\mu+\rho)}\right)^{n-1} \frac{L\|\theta_0 - \theta^*\|_2^2}{2} \tag{7}$$

In order to prove this, define κ_{n-1}^t as $\{\kappa_{n-1}^t \mid g_n^t \in \mathscr{S}_{L,\rho}(f^t, \kappa_{n-1}^t)\}$ and $\kappa_{n-1}^t \in \Theta$. The points κ_{n-1}^t can be drawn from Bernoulli distribution with probabilities $p(\kappa_{n-1}^t = \theta_{n-1}|\mathscr{F}_{n-1}) = \delta$ and $p(\kappa_{n-1}^t = \kappa_{n-2}^t|\mathscr{F}_{n-1}) = 1-\delta$, where $\delta \triangleq \frac{1}{T}$ and $\mathscr{F}_n$ represents all the information summing upto θ_n. Basically, κ_{n-1}^t can be chosen between θ_{n-1} and κ_{n-2}^t. For all $n \geq 1$, expectation can be calculated as:

$$\mathbb{E}\big[\|\theta^* - \kappa_{n-1}^t\|_2^2\big] = \delta\mathbb{E}\big[\|\theta^* - \theta_{n-1}\|_2^2\big] + (1-\delta)\mathbb{E}\big[\|\theta^* - \kappa_{n-2}^t\|_2^2\big] \tag{8}$$

The equation [4] in MISO setting would become:

$$f(\theta_n) - f(\theta) \leq \frac{1}{T}\sum_{t=1}^{T}\left(\frac{L}{2}\|\theta - \kappa_{n-1}^t\|_2^2 - \frac{\rho}{2}\|\theta - \theta_n\|_2^2\right) \tag{9}$$

Define $A_{n-1} = \mathbb{E}[\frac{1}{2T}\sum_{t=1}^{T}\|\theta - \kappa_{n-1}^t\|_2^2]$ and $\xi_n = \mathbb{E}[\frac{1}{2}\|\theta - \theta_n\|_2^2]$. We have two cases:

Non- strongly Convex Case ($\rho = L$) From equation 9, choosing $\theta = \theta^*$ we get,

$$\mathbb{E}[f(\theta_n) - f^*] \leq LA_{n-1} - L\xi_n$$

From equation (8), it can easily seen that $A_n = \delta\xi_n + (1-\delta)A_{n-1}$ and therefore we get a recursive relation as:

$$\mathbb{E}[f(\theta_n) - f^*] \leq \frac{L}{\delta}(A_{n-1} - A_n)$$

If the above expression is summed for $i = 1, 2....n$ and Jensen's Inequality is applied we get:

$$\mathbb{E}[f(\bar{\theta}_n) - f^*] \leq \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[f(\theta_i) - f^*] \leq \frac{LA_0}{n\delta}$$

which is same as what promised in equation 6. As wee can see from 6, the convergence of non-strongly convex functions is influenced by initialization θ_0. And therefore, θ_0 must be chosen carefully, as it can lead to faster convergence

$\mu-$strongly convex case: Now, f^ts are $\mu-$ strongly convex therefore, equation 9 results in:

$$\mu\xi_n \leq \mathbb{E}[f(\theta_n) - f^*] \leq LA_{n-1} - \rho\xi_n$$

From eq 8 and above we get, $A_n = \delta\xi_n + (1-\delta)A_{n-1} \leq \beta A_{n-1}$ where $\beta = \left(\frac{\delta L}{\mu+\rho} + (1-\delta)\right)$

Recursively, we get $A_n \leq \beta^n A_0$ and since, $A_0 = \xi_0$, the expression becomes

$$\mathbb{E}[f(\theta_n) - f^*] \leq LA_{n-1} \leq L\beta^{n-1}A_0$$

where $\beta \leq 1 - \frac{2\delta\mu}{\rho+\mu}$, and thus the promised rate mentioned in eq 7 is proved. Also, it is important to see that majority of proof uses the fact that each of f^t should be smooth and convex. In case of large scale machine learning, that may not always be the case, especially if there are different functions f^t for each t, and T is large. The condition can be relaxed as shown by [5], where sum of smooth and non-smooth convex functions were considered.

4 Implementation

In this section, we will discuss the practical updates we have implemented for the "MISO" algorithm. We will also present the results of the various experiment performed by us.

4.1 Practical updates for Algorithm

In our experiments we would assume unconstrained optimization where the domain $\Theta = \mathbb{R}^p$. In general, when f^t are assumed as L smooth, the updates mentioned in Step 6 of Algorithm 2 would change to

$$\theta_n \leftarrow \frac{1}{T}\sum_{t=1}^{T}\kappa_{n-1}^t - \frac{1}{LT}\sum_{t=1}^{T}\nabla f^t(\kappa_{n-1}^t) \tag{10}$$

The surrogate used in equation 10 is Lipschitz. The L-Lipschitz surrogate $g(\theta)$ of function $f(\theta)$ at point κ is defined as:

$$g(\theta, \kappa) = f(\kappa) + \nabla f(\kappa)^\top(\theta - \kappa) + \frac{L}{2}\|\theta - \kappa\|_2^2$$

Also, κ_{n-1}^t for $n \geq 2$ in eq 10 are computed as

$$\kappa_{n-1}^t = \theta_{n-1} \qquad \text{for } t = \hat{t}_n$$
$$\kappa_{n-1}^t = \theta_{n-2} \qquad \text{for } t \neq \hat{t}_n$$

Thus any iteration n, we will have to keep track of κ_n, which will result in heavy book-keeping, thus having high space complexity of order $\mathcal{O}(T)$. However, the time complexity of each iteration is constant and does not depend upon T, as the updates are made only for the chosen point or group of chosen point (minibatches).

Based on the differences in assumptions, initialization, update rules, we have proposed the three versions of the discussed "MISO" algorithm discussed below:

4.1.1 MISO-1 : This is a sort of vanilla "MISO" algorithm. Here we choose a global $L_1 = 2^{-k}L_0$ by running one epoch for 5% of training dataset and performing line search over the positive values of k. Value of k is chosen which yields the minimum value of objective function. L_0 can be any rough estimate, which upper bounds the true L_1

4.1.2 MISO-2 : MISO-2 is more aggressive in its computations than MISO-1. The L_1 is computed as proposed in section 4.1.1. Compute $L_2 = \eta L_1$ where $\eta = 5\%$. Also, at iteration n, compute $A_n = \sum_{a_n^t}$ and $B_n = \sum_{b_n^t}$, where a_n^t and b_n^t are computed as:

$$a_n^t = f^{\hat{t}_n}(\theta_{n-1}) \quad \text{and} \quad b_n^t = g_{L_2}^{\hat{t}_n}(\theta_{n-1}) \qquad \text{for } t = \hat{t}_n$$
$$a_n^t = a_{n-1}^t \quad \text{and} \quad a_n^t = a_{n-1}^t \qquad \textit{otherwise},$$

where $g_{L_2}^{\hat{t}_n}$ is the $L-2$ Lipschitz surrogate. Check if $A_n \geq B_n$, in that case increase the value of L_2 until you achieve $A_n < B_n$.

Thus MISO-2 helps in choosing tighter L compared to MISO-1. But again, this algorithm require us to keep track of more variables namely A_n and B_n which require the extra bookkeeping in addition to the κ_n from equation 10.

4.1.3 MISO-μ : The updates of MISO-μ algorithm are especially designed for μ strongly and L-smooth convex functions and aims to achieve the desired linear convergence theoretically described in section 3.2.1. The updates for surrogate function at iteration n in this case are given by

$$g(\theta, \kappa_{n-1}^t) = f^t(\kappa_{n-1}^t) + \nabla f^t(\kappa_{n-1}^t)^\top(\theta - \kappa_{n-1}^t)\frac{\mu}{2}\|\theta - \kappa_{n-1}^t\|_2^2$$

The surrogate g in above update is
both $\mu-$ strongly convex and μ smooth. However, an assumption that $2L \leq T\mu$ is taken, while implementing this update. Without this assumption the algorithm will fail to achieve the linear time convergence it claimed.

4.2 Experiments and Results

For our implementation we have chosen the dense dataset `alpha` from the various datasets mentioned. Training dataset `alpha` has T = 0.5 million data points each with feature vector of size $p = 500$ and a corresponding binary label (-1,1). We kept 80% of data for training and 20% for testing. We chose the problem of $l2$ regularized logistic regression whose objective function looks like:

$$\min_{\theta \in \mathbb{R}^p} \frac{1}{T}\sum_{t=1}^{T}\log(1 + \exp(-y_t x_t^\top \theta)) + \frac{\lambda}{2}(\|\theta\|_2^2)$$

We implemented MISO-1, MISO-2 and MISO-μ algorithms as described in section 4.1. For implementing the algorithms, we borrowed `mexIncrementalProx` function from the SPAMS library [3]. We compared these algorithms with state of the art for large scale optimization algorithm: Stochastic Gradient Descent (SGD) [2] and Stochastic Variance Reduced Gradient (SVRG) [4]. The SGD and SVRG is implemented in a straightforward fashion from their libraries.

As clear from the figures 1, with epochs all the methods converge in more or less in a similar manner w.r.t. epochs. However, the MISO-μ algorithm doesn't perform as good as MISO-1/SGD perhaps because of the strong μ convex assumption during it's updates.

The best performance is given by MISO-1 which is again due to its more general nature and aggressive computations of Lipschitz constants while updating the surrogate function.

The difference between the convergence of MISO and SGD/SVRG is not very evident from the figure 1.

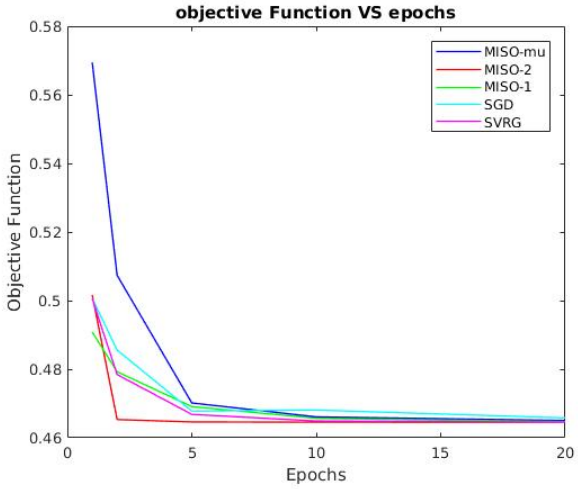

Figure 1: Objective function vs epochs

But, from the Figure 2 the difference between MISO-class algorithms and gradient based algorithms is evident. For similar convergence, the MISO algorithms turned out to be much faster than the gradient based algorithms. The performance on the test data set as seen from the figure 3 provides further proofs that MISO algorithms are producing **slightly better** results than Gradient based algorithms **in much less time** with only exception being the MISO-μ algorithm whose performance is worse than SGD/SVRG, but still it is considerably faster. We were also monitoring the RAM usage while implementing all these algorithms and found that RAM usage was maximum for implementation for MISO-2 while minimum for MISO-μ. This could be due to the extra book keeping we discussed in section 4.1.2

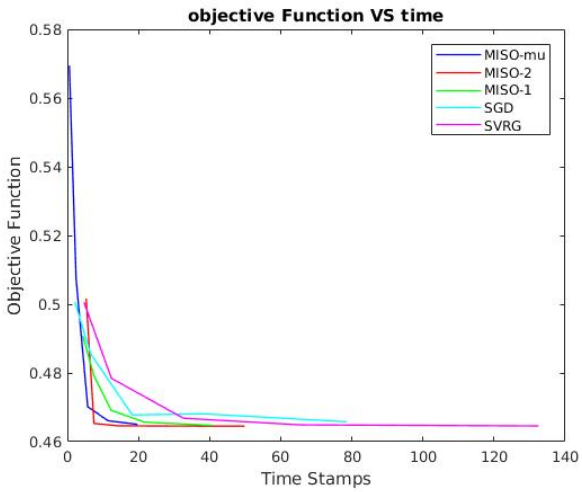

Figure 2: Objective function vs time

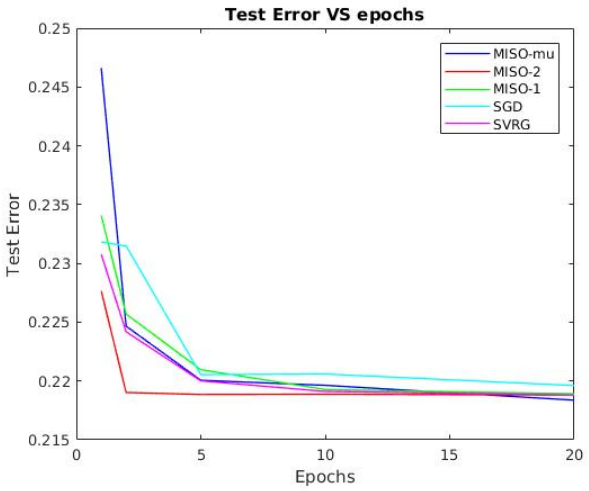

Figure 3: Test Error Vs epochs

We studied the variation of λ on the MISO-μ algorithm. Ideally, the objective function should decrease on decreasing the λ. But, here we saw a very different phenomenon. We ran MISO-μ for various $\lambda's$ for 20 epochs and noticed that below a certain threshold, the objective function diverges instead of converging. One of the important assumptions of MISO-μ updates are that apart from being convex, function should be bounded below, but as we decrease λ, the regularization term vanishes and we are left with only the cross-entropy function which is not bounded below. And, that's the reason for poor performance at the very low values of λ

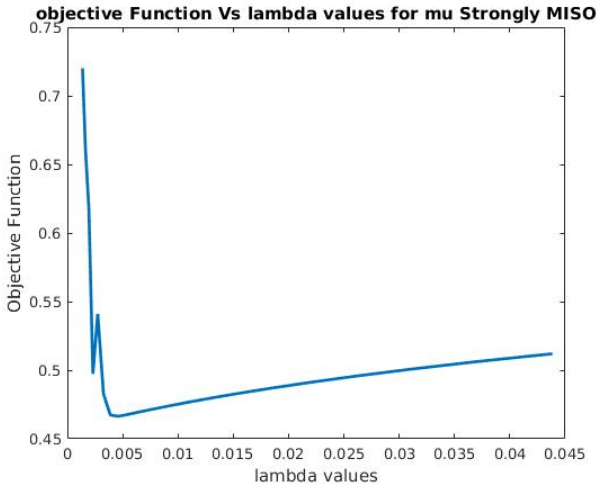

Figure 4: Variation of lambda

Note: *We could only run the code for single dataset and that too only dense one, because of the time and space constraints of our machines. Time constraints were still manageable, as we set the size of the minibatches to 1, but the space constraints were not manageable. Most of the times, our systems were freezing under the load. At the time of running, the maximum RAM usage on our systems was around* 14 *Gigabytes.*

5 Conclusion

We presented the scalable version of majorization-minimization algorithm for machine learning problems. We have covered the theoretical convergence as well as implementation of the presented algorithm both for convex and non-convex objective functions. Overall MISO algorithms were at par with previous gradient based techniques for scalable machine learning. They did *slightly better* on the test data while being trained for *considerably* less time. Essentially, MISO can also be interpreted as a variance reduction technique for SGD which helps in achieving constant convergence and learning rate [10].

References

[1] Nguyen Thai An, Daniel Giles, Nguyen Mau Nam, and R. Blake Rector. 2016. The Log-Exponential Smoothing Technique and Nesterov's Accelerated Gradient Method for Generalized Sylvester Problems. *Journal of Optimization Theory and Applications* 168, 2 (01 Feb 2016), 559–583. https://doi.org/10.1007/s10957-015-0811-z

[2] Léon Bottou. 2010. Large-Scale Machine Learning with Stochastic Gradient Descent. In *Proceedings of COMPSTAT'2010*, Yves Lechevallier and Gilbert Saporta (Eds.). Physica-Verlag HD, Heidelberg, 177–186.

[3] Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis R Bach. 2010. Proximal Methods for Sparse Hierarchical Dictionary Learning.. In *ICML*, Vol. 1. Citeseer, 2. http://spams-devel.gforge.inria.fr/index.html

[4] Rie Johnson and Tong Zhang. 2013. Accelerating Stochastic Gradient Descent using Predictive Variance Reduction. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 315–323.

[5] Mojtaba Kadkhodaie, Maziar Sanjabi, and Zhi-Quan Luo. 2014. On the Linear Convergence of the Approximate Proximal Splitting Method for Non-smooth Convex Optimization. *Journal of the Operations Research Society of China* 2, 2 (01 Jun 2014), 123–141. https://doi.org/10.1007/s40305-014-0047-x

[6] Julien Mairal. 2013. Optimization with first-order surrogate functions. In *International Conference on Machine Learning*. 783–791.

[7] Julien Mairal. 2013. Stochastic majorization-minimization algorithms for large-scale optimization. In *Advances in Neural Information Processing Systems*. 2283–2291.

[8] J. Mairal. 2015. Incremental Majorization-Minimization Optimization with Application to Large-Scale Machine Learning. *SIAM Journal on Optimization* 25, 2 (2015), 829–855. https://doi.org/10.1137/140957639 arXiv:https://doi.org/10.1137/140957639

[9] Herbert Robbins and Sutton Monro. 1951. A Stochastic Approximation Method. *The Annals of Mathematical Statistics* 22, 3 (1951), 400–407. http://www.jstor.org/stable/2236626

[10] Ruiliang Zhang, Shuai Zheng, and James T. Kwok. 2015. Fast Distributed Asynchronous SGD with Variance Reduction. *CoRR* abs/1508.01633 (2015). arXiv:1508.01633 http://arxiv.org/abs/1508.01633