Parallel Majorization Minimization with Dynamically Restricted Domains for Nonconvex Optimization

Yan Kaganovsky[*]
Duke University

Ikenna Odinaka[*]
Duke University

David Carlson
Columbia University

Lawrence Carin
Duke University

Abstract

We propose an optimization framework for nonconvex problems based on majorization-minimization that is particularity well-suited for parallel computing. It reduces the optimization of a high dimensional nonconvex objective function to successive optimizations of locally tight and convex upper bounds which are additively separable into low dimensional objectives. The original problem is then broken into simpler *parallel* sub-problems while still guaranteeing the monotonic reduction of the original objective function and convergence to a local minimum. Due to the low dimensionality of each sub-problem, second-order optimization methods become computationally feasible and can be used to accelerate convergence. In addition, the upper bound can be restricted to a *local dynamic* convex domain, so that it is better matched to the local curvature of the objective function, resulting in accelerated convergence.

1 Introduction

We consider the following optimization problem

$$\theta^* = \arg\min_{\theta \in \Omega_F} \mathcal{C}(\theta), \tag{1}$$

where Ω_F is a closed convex set and $\mathcal{C}$ is generally *non-convex* and has the following form

$$\begin{aligned}\mathcal{C}(\theta) &= \sum_{m=1}^{M} F(\theta; x_m, y_m) + \lambda R(\theta) \\ &= \sum_{m=1}^{M} f_m(\theta^T x_m) + \lambda R(\theta),\end{aligned} \tag{2}$$

where $F(\theta; x_m, y_m)$ represents the data-mismatch for the mth data point $(x_m, y_m) \in \mathbb{R}^p \times \mathbb{R}^d$ and $\theta \in \mathbb{R}^p$ are the parameters (latent variables) to be learned. f_m is a twice continuously differentiable function $\mathbb{R} \to \mathbb{R}$ (possibly nonconvex) that depends on y_m. Here R represents a penalty due to prior knowledge (e.g., the 2-norm penalty) that is separable and twice continuously differentiable (possibly non-convex). The class of optimization problems in (1) appears in various fields such as machine learning, signal/image processing, imaging and bioinformatics, to name a few.

Often, these problems can be of huge size (large p) and involve big datasets (large M). There is a growing need for faster algorithms that can deal with large problems in a numerically efficient way. The availability of high performance multi-core computing platforms makes it increasingly desirable to develop parallel optimization methods.

A general optimization principle that often leads to parallelizable algorithms is *majorization-minimization* (**MM**) (Lange, 2000), also known as optimization transfer. The idea is essentially to replace a difficult optimization problem by a sequence of simpler problems. At each iteration, the original objective function is approximated by a locally tight upper bound surrogate so that minimizing the surrogate ensures that the objective function is decreased. The motivation for using MM can be to simplify the optimization process, make it more suitable for parallel computing, or both. The global convergence of this type of algorithms has been established for several cases (Lanckriet and Sriperumbudur, 2009; Mairal, 2015; Ahn et al., 2006; Jacobson and Fessler, 2007). Various well known approaches can be interpreted from an MM point of view, such as expectation-maximization (EM) algorithms in statistics (Dempster, 1977; Neal and Hinton, 1998) and difference-of-convex programming (Lanckriet and Sriperumbudur, 2009). The MM procedure has been successfully used in tomographic imaging (Ahn et al., 2006; Kaganovsky et al., 2015), image processing (Sotthivirat and Fessler, 2002), and matrix factorization (Lee and Seung, 2001), to name a few.

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51. Copyright 2016 by the authors. * Indicates equal contributions.

A related principle is *Successive Convex Approximations* (**SCA**) (Razaviyayn et al., 2013; Razaviyayn et al., 2014), where as opposed to MM, the surrogate is not necessarily a global upper bound of the objective so that the monotonic reduction of the objective is not guaranteed. However, due to additional assumptions such as Lipchitz smoothness, convergence can be guaranteed if the step sizes are chosen to satisfy certain criteria related to the Lipchitz constant of the gradient. Well known examples that use this approach are gradient-based proximal methods (Beck and Teboulle, 2009; Combettes and Pesquet, 2011). Recently, Razaviyayn et al. (2014) proposed an SCA method for *non-convex* optimization with *parallel* block updates (**PSCA**) while still guaranteeing convergence.

A class of parallelizable MM algorithms using partitioned-separable paraboloidal surrogates for nonconvex optimization was proposed by Erdogan and Fessler (1999) and also Sotthivirat and Fessler (2002). In this approach, the objective is approximated by a high-dimensional *quadratic* global upper bound and then an additional upper bound is created by using Jensen's inequality to obtain a block coordinate descent algorithm. This approach allows one to update blocks of variables *in parallel* while still guaranteeing the *monotonic* decrease of the original objective and also convergence. A limitation of this method is that it relies on positivity constraints to construct the quadratic upper bound.

The contributions of this work are as follows:

- We propose a new type of parallelizable *local* majorization-minimization for *nonconvex* optimization which uses successive optimizations of convex surrogates. The domain of the surrogate is controlled at each iteration to better match it to the local curvature of the objective function, resulting in accelerated convergence when using second-order methods.
- Similar to Erdogan and Fessler (1999) and Sotthivirat and Fessler (2002), and in contrast to Razaviyayn (2014), our method guarantees *monotonic* decrease of the objective function. This has the advantage that deviation from monotonicity serves as indication of error in the code.
- In contrast to Erdogan and Fessler (1999), Sotthivirat and Fessler (2002), and Razaviyayn (2014), our method uses surrogates that are matched to a local compact region of the objective function rather than globally. Also, in contrast to the first two methods, our method does not require positivity constraints.
- The proposed method leads to parallel block coordinate descent algorithms, where blocks are updated simultaneously on different processors while still automatically guaranteeing the reduction of the original objective function. Each of the blocks can be reduced to a small enough size such that second-order methods are practical and can be used to accelerate convergence.
- The proposed method can be used with an *arbitrary* block size. This enables more flexibility to better explore the trade-off between convergence rate per iteration, and the time per iteration (including parallelization costs such as communication between different processors), at the price of a more restrictive class of problems than Razaviyayn (2014). Note that the method by Razaviyayn (2014) is practically limited to small block sizes since it requires computing the lowest eigenvalue of the feature/system sub-matrix corresponding to each block, which becomes computationally expensive as the block size increases.
- We derive guarantees for global convergence of the proposed class of algorithms.
- We test the proposed framework on nonconvex support vector machines based on a sigmoid loss function (similar to the "ramp-loss" used by Ertekin et al. (2011)) and on logistic regression with nonconvex penalties (Mairal, 2015).
- The proposed method is shown to yield comparable or superior classification accuracy relative to many prior methods for non-convex optimization such as PSCA, and popular stochastic optimization methods, e.g., AdaGrad, and RMSProp, *but without the need for tuning learning rates.*

2 Majorization Minimization with Dynamically Restricted Domains

2.1 Overview of Majorization Minimization

We begin with some basic definitions of the majorization-minimization procedure (**MMP**) that will be considered in this paper. Unless otherwise stated, all functions are $\mathbb{R}^p \to \mathbb{R}$.

Definition $\mathcal{S}(\theta; \hat{\theta}, \Omega_M)$ is said to be a **local first-order surrogate** of a function $\mathcal{C}$ around an expansion point $\hat{\theta}$ if the following conditions are satisfied

$$\mathcal{S}(\theta; \hat{\theta}) \geq \mathcal{C}(\theta) \qquad \forall \theta \in \Omega_M, \tag{3}$$

$$\mathcal{S}(\hat{\theta}; \hat{\theta}) = \mathcal{C}(\hat{\theta}), \tag{4}$$

$$\nabla_\theta \mathcal{S}|_{\theta=\hat{\theta}} = \nabla_\theta \mathcal{C}|_{\theta=\hat{\theta}}, \tag{5}$$

$$\Omega_M \cap \Omega_F \neq \emptyset, \tag{6}$$

$$\hat{\theta} \in \mathrm{Int}(\Omega_M), \tag{7}$$

where Ω_M is called the majorization domain, $\text{Int}(\cdot)$ denotes the interior of a set and Ω_F is the feasible set in the original problem in (1). If $\mathcal{S}(\theta;\hat{\theta})$ is also convex with respect to θ on some convex set containing Ω_M, it is said to be a **convex local first-order surrogate**.

The condition in (3) states that $\mathcal{S}$ is an upper bound of $\mathcal{C}$ in Ω_M. The conditions in (4)–(5) state consistency of the values of the functions and their gradients at the expansion point $\hat{\theta}$. Condition (6) is added to ensure that a feasible solution exists. The condition in (7) ensures that $\hat{\theta}$ is in Ω_M and not a boundary point.

The basic idea in MMP is to replace the hard problem in (1) by a sequence of easier subproblems

$$\theta^{(t+1)} \in \arg\min_{\theta}\{\mathcal{S}(\theta;\theta^{(t)},\Omega_M^{(t)}) : \theta \in \Omega_F \cap \Omega_M^{(t)}\}, \quad (8)$$

where $\mathcal{S}(\theta;\hat{\theta},\Omega_M)$ is the surrogate for the cost function $\mathcal{C}$ in (1) with the expansion point $\hat{\theta}$ chosen according to the last iteration $\hat{\theta} = \theta^{(t)}$ and the majorization domain $\Omega_M = \Omega_M^{(t)}$ can also change with iteration. Also note that we do not require $\mathcal{S}$ to be strongly convex, so it can have multiple minima; accordingly, in the definition of (8) we use $\in$ instead of equality. This MM procedure is guaranteed to decrease the original objective function (or leave it unchanged) if the surrogate is decreased (or unchanged), since

$$\mathcal{C}(\theta^{(t+1)}) \leq \mathcal{S}(\theta^{(t+1)};\theta^{(t)}) \leq \mathcal{S}(\theta^{(t)};\theta^{(t)}) = \mathcal{C}(\theta^{(t)}), \quad (9)$$

where the first inequality follows from (3), the second inequality follows from (8) and the equality follows from (4). We will also show in Sec. 3 that under some mild conditions, this procedure is guaranteed to converge to a local minimum of $\mathcal{C}$.

The definition in (3)–(7) includes, as a special case, the paraboloidal surrogates used by Erdogan and Fessler (1999) and also Sotthivirat and Fessler (2002), where the feasible set is assumed to be $\Omega_F = \mathbb{R}_+^p$, the majorization is on the entire feasible region $\Omega_M = \Omega_F$ and the surrogate $\mathcal{S}$ is restricted to a particular quadratic form. However, to the best of our knowledge, the possibility of changing the majorization domains s.t. $\Omega_M \subset \Omega_F$ has not been significantly explored. It should be noted that this idea has been mentioned by Jacobson and Fessler (2007), albeit without any specific algorithm that actually uses this approach. As we shall show below, there are important practical advantages to using *local* convex surrogates with dynamic majorization domains.

2.2 Convex Local Majorization

The first step of our MM procedure is to construct a convex *local* first-order surrogate to the nonconvex cost function by utilizing the specific structure in (2). Specifically, our approach relies on the observation that in most models of the form in (2) the scalar functions f_m and R and their derivatives are given in closed-form. Thus, we first construct a convex surrogate for the scalar functions $f_m(z)$, where later we will substitute $z = \theta^T x_m$. We note that since the set of functions f_m generally depend on y_m they can be different for different m.

Let $f_m : \mathbb{R} \to \mathbb{R}$ be a twice continuously differentiable nonconvex function with its second derivative bounded below, i.e.,

$$\min f_m''(z) > -\infty \qquad \forall m, \quad (10)$$

where f_m'' is the second derivative of f_m. Define the minimum second derivative on the interval $[a_m, b_m]$

$$c_m(a_m, b_m) := \min_{a_m \leq z \leq b_m} f_m''(z). \quad (11)$$

We now define $\bar{f}_m$ as

$$\bar{f}_m(z;\hat{z}) := f_m(z) + [-c_m(a_m, b_m)]_+(z-\hat{z})^2/2, \quad (12)$$

where $[x]_+ = max(0, x)$. It is easy to verify that for any particular m, the function $\bar{f}_m(z;\hat{z})$ in (12) is a global first-order surrogate for $f_m(z)$ (see definition in Sec. 2.1), i.e., $\bar{f}_m(z;\hat{z}) \geq f_m(z)$ with equality at $z = \hat{z}$ and $\bar{f}_m' = f_m'$ at $z = \hat{z}$. Also, convexity, i.e., $\bar{f}_m'' \geq 0$, is guaranteed only on the domain $[a_m, b_m]$, so in accordance with the definitions in Sec. 2.1, to obtain a convex surrogate the majorization domain should be chosen as $\Omega_M = \{\theta \,|\, a_m \leq \theta^T x_m \leq b_m, \forall m\}$. Note that for $(a_m, b_m) = (-\infty, \infty)$, $[-c_m(a_m, b_m)]_+$ in (12) equals the absolute value of the global minimum of f_m'' if the minimum is negative and zero otherwise, so $\bar{f}_m$ in (12) is convex on the entire real line. Figure 1 illustrates the need for the local surrogate introduced in (12) and why one should consider choosing $a_m > -\infty$ and $b_m < \infty$. When the expansion point $\hat{z}$ is located at a convex region ($f_m'' > 0$), adding $[-\min_{z\in\mathbb{R}} f_m''(z)]_+(z-\hat{z})^2$ to the function $f_m(z)$ can lead to very high positive curvature of the surrogate and to small steps when minimizing the surrogate using second-order methods. In order to achieve faster convergence, we restrict the surrogate to the interval $[a_m, b_m]$ and choose c_m in (11) according to the *local* minimum of f_m'' (if it is negative), which can result in a much lower curvature of the surrogate. The proposed modification leads to wider surrogates which enable taking larger steps (if needed) when using second-order methods, as illustrated by the red curve in Fig. 1. In fact, if $[a_m, b_m]$ is contained in a convex region, then the quadratic term in (12) is dropped and the original function can be used in this region.

To appreciate the difficulty in choosing the majorization domains Ω_M or $[a_m, b_m]$, note that the Hessian

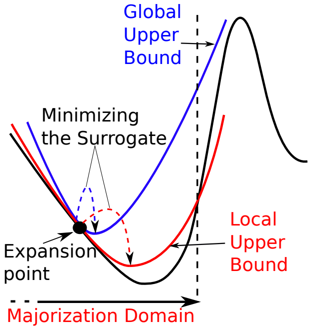

Figure 1: Illustration of first-order convex global (blue) and local (red) surrogates for a nonconvex cost function (black). The minimum of the local surrogate is seen to be closer to the minimum of the nonconvex function than the global surrogate. All surrogates are constraint to have the same function value and gradient as the nonconvex function at the expansion point.

of the nonconvex objective in (2) with respect to θ has the form $\mathcal{H}(\theta) = X\Lambda_1 X^T + \lambda\Lambda_2$, where $X \in \mathbb{R}^{p\times M}$ is the column-wise concatenation of the vectors x_m for different m, $\Lambda_1 = \text{Diag}(\{f''_m(\theta^T x_m)\}_{m=1}^M)$ with $\text{Diag}(\{v_m\}_{m=1}^M)$ denoting a diagonal matrix in which the entries on the main diagonal are specified by $v_1, v_2, ..., v_M$, and Λ_2 is the Hessian of the penalty term. Accordingly, it is generally possible that a local minimum point θ_{min}, where $\mathcal{H}(\theta_{min}) \succeq 0$, can also satisfy $f''_m(\theta_{min}^T x_m) < 0$ for *some* m. Therefore, choosing $[a_m, b_m]$ in (12) **is not a trivial task** as solutions with $f''_m < 0$ must also be allowed. We propose a principled approach to tackle this problem in Sec. 4.

Based on the scalar surrogates in (12), we construct a multivariate convex surrogate to a function of the form $\sum_m f_m(\theta^T x_m)$ (see (2)) by substituting

$$z_m = \theta^T x_m, \qquad \hat{z}_m = \hat{\theta}^T x_m, \tag{13}$$

for each m separately, where z_m and $\hat{z}_m$ are the current and reference values of the argument of f_m, respectively. This is summarized in the following lemma.

Lemma 2.1 *Let $F(\theta) = \sum_m f_m(\theta^T x_m)$ with f_m a twice continuously differentiable (possibly nonconvex) function. Define*

$$\bar{F}(\theta; \hat{\theta}) := F(\theta) + \frac{1}{2}(\theta - \hat{\theta})^T X D X^T (\theta - \hat{\theta}), \tag{14}$$

where $\hat{\theta} \in \Omega_F$ is an expansion point, $X \in \mathbb{R}^{p\times M}$ is a matrix with the mth column equal to x_m and D is a diagonal matrix

$$D := Diag(\{[-c_m(a_m, b_m)]_+\}_{m=1}^M), \tag{15}$$

*where $c_m(a_m, b_m)$ is defined in (11). The function $\bar{F}(\theta; \hat{\theta})$ is a **convex local first-order surrogate** (see (3)–(5)) for F on the majorization domain $\Omega_M := \{\theta \,|\, a_m \le \theta^T x_m \le b_m, \forall m\}$ when a_m, b_m are chosen such that $\hat{\theta} \in Int(\Omega_M)$. (See proof in supplementary material).*

$\bar{F}$ in (14) is generally not convex outside Ω_M. In order to allow the use of any general algorithm for convex functions, we extend the surrogate to be globally convex. This extension will also be essential for our decomposition method in Sec. 2.3 that relies on Jensen's inequality and requires the function to be globally convex. First, we introduce an extension to $\bar{f}$ from (12) in Eq. (16) (see next page), where $\bar{f}'_m(z; \hat{z})$ and $\bar{f}''_m(z; \hat{z})$ denote the first and second derivatives of $\bar{f}_m$ with respect to z. Note that $\tilde{f}_m$ in Eq. (16) is globally convex even in regions where it is not an upper bound of f_m and it is also twice continuously differentiable. The extension of $\bar{F}$ from (14) can now be written as

$$\tilde{F}(\theta; \hat{\theta}) = \sum_m \tilde{f}_m(\theta^T x_m; \hat{\theta}^T x_m). \tag{17}$$

The proposed MMP with *Dynamically Restricted majorization Domains* (**DRD**) is summarized in Algorithm 2.1. It is important to note that it is not necessary to solve the surrogate subproblem in (8) exactly. It is sufficient just to decrease the value of the surrogate which will result in the decrease of $\mathcal{C}$ in (1). Since the surrogate is a local approximation of $\mathcal{C}$ it might explore a small region in θ space and it is more beneficial to re-expand the surrogate at a new point than to minimize the current surrogate exactly. In addition, the extra constraint due to Ω_M in (8) might make the problem more difficult to solve. Instead, we first minimize the surrogate in Algorithm 2.1 without constraining the solution to lie inside the majorization domain Ω_M and then check if it lies in Ω_M. If not, we project the solution to the boundary of Ω_M along the line connecting the current solution $\theta^{(t+1)}$ and the previous one $\theta^{(t)}$. The following lemma asserts that this projection always exists and necessarily leads to a decrease of the surrogate which implies the decrease of $\mathcal{C}$. We also note that Ω_M can be modified as needed, so in cases where these projections slow down the convergence, one can expand Ω_M to allow larger step sizes.

Lemma 2.2 *Let Ω_F and Ω_M be non-empty convex sets with $\Omega_F \cap \Omega_M \neq \emptyset$ and $\Omega_M := \{\theta \,|\, a_m \le \theta^T x_m \le b_m, \forall m\}$. Let $\theta_1 \in \Omega_F \cap \Omega_M$ and let $\theta_2 \in \Omega_F$ with $\theta_2 \notin \Omega_M$ s.t. $F(\theta_2) < F(\theta_1)$ with F a convex function. In addition, assume a_m, b_m are chosen such that $\theta_1 \in int(\Omega_M)$. Then there exists $\lambda \in (0, 1)$ s.t. $\theta^* := \lambda\theta_2 + (1-\lambda)\theta_1$ satisfies $\theta^* \in \Omega_F \cap \Omega_M$. Furthermore, any such point satisfies $F(\theta^*) < F(\theta_1)$. (Proof is provided in the supplementary material).*

$$\tilde{f}_m(z;\hat{z}) := \begin{cases} \bar{f}_m(a_m) + \bar{f}'_m(a_m)(z-a_m) + \bar{f}''_m(a_m)(z-a_m)^2/2, & z < a_m \\ \bar{f}_m(z;\hat{z}), & a_m \leq z \leq b_m \\ \bar{f}_m(b_m) + \bar{f}'(b_m)(z-b_m) + \bar{f}''(b_m)(z-b_m)^2/2, & z > b_m \end{cases} \tag{16}$$

```
Algorithm 2.1: MM-DRD
 for $t \leftarrow 1$ to $T$   (or until converged)
   for $m \leftarrow 1$ to $M$
     Choose $[a_m^{(t)}, b_m^{(t)}]$
     comment: see Sec. 4 for details
     Compute $c_m^{(t)}$ in (11) using $[a_m^{(t)}, b_m^{(t)}]$
   Find $\theta^{(t+1)} \in \arg\min_{\theta \in \Omega_F} \{\tilde{F}(\theta; \theta^{(t)}, \Omega_M^{(t)})\}$
   comment: $\tilde{F}$ is given in Eq. (17)
   Let $\Omega_M^{(t)} := \{\theta \,|\, a_m^{(t)} \leq \theta^T x_m \leq b_m^{(t)}, \forall m\}$
   if $\theta^{(t+1)} \notin \Omega_M^{(t)}$ then
   Find $\lambda \in (0,1)$ s.t.
     $\theta^* = \lambda\theta^{(t+1)} + (1-\lambda)\theta^{(t)} \in \Omega_M^{(t)} \cap \Omega_F$
   set $\theta^{(t+1)} = \theta^*$
 return $(\theta^{(t+1)})$
```

One practical solution to performing the projection is to use the point on the boundary of Ω_M along the line connecting the previous and current solution, which has the following simple solution

$$\lambda = \min\left(\left\{\frac{b_m^{(t)} - x_m^T\theta^{(t)}}{x_m^T(\theta^{(t+1)} - \theta^{(t)})}, \frac{a_m^{(t)} - x_m^T\theta^{(t)}}{x_m^T(\theta^{(t+1)} - \theta^{(t)})}\right\}_{m=1}^{M} \cap (0,1)\right). \tag{18}$$

It should be noted that the convex surrogate in (17) can be minimized by any standard software package for convex optimization which is already more convenient than minimizing the original nonconvex cost function.

2.3 Jensen-Type Decomposition for Parallel Block Coordinate Descent

As shown in Sec. 3, a necessary condition for Algorithm 2.1 to converge is that the surrogate is decreased. This requires one to repeatedly evaluate the surrogate in order to determine the step-size (line-search). However, for high dimensional problems, with a large number of variables and data points, this operation is extremely time consuming. In order to alleviate this difficulty, we make use of a powerful decomposition procedure that is based on Jensen's inequality (Boyd and Vandenberghe, 2004) and originally proposed by De Pierro (1994) that results in a surrogate that is **additively separable** with respect to blocks of variables (Erdogan and Fessler, 1999). This leads to a **parallel** block coordinate descent algorithm where all blocks are updated simultaneously while still guaranteeing the decrease of the objective function. Since the optimization using each block potentially involves a small number of variables, it allows the use of second-order methods. Note that naively minimizing the objective with respect to blocks of variables in parallel without using the decomposition we present here would **not** guaranty the decrease of the objective.

We can now introduce the above mentioned decomposition. Separate $\theta = [\theta^1;\ \theta^2;\ \ldots\ ; \theta^K]$ into K arbitrary blocks. Define $\mathbb{S}^k$ as the set of all indexes used in the kth block with the corresponding variables denoted by the column vector θ^k and $|\mathbb{S}^k|$ as the number of variables in θ^k. We define the following function

$$\begin{aligned} \mathcal{S}_m(\theta;\hat{\theta}) &:= \sum_{k=1}^{K} \mathcal{S}_m^k(\theta^k; \hat{\theta}, r_m^k, x_m^k) \\ &:= \sum_{k=1}^{K} r_m^k \tilde{f}_m(\hat{\theta}^T x_m + (\theta^k - \hat{\theta}^k)^T x_m^k / r_m^k), \end{aligned} \tag{19}$$

where $\tilde{f}_m$ is defined in (16), $x_m^k \in \mathbb{R}^{|\mathbb{S}_k|}$ is a subset of x_m corresponding to θ^k and r_m^k is defined as

$$r_m^k = \|x_m^k\|_1 / \|x_m\|_1. \tag{20}$$

Note that $\mathcal{S}_m$ is an additively separable function with respect to the blocks ($\mathcal{S}_m^k$ corresponds to the kth block), so it naturally leads to a *parallel* block coordinate descent approach where all blocks are updated simultaneously while still guaranteeing the decrease of the objective function.

Lemma 2.3 *The function $\mathcal{S} := \sum_m \mathcal{S}_m(\theta;\hat{\theta})$ with $\mathcal{S}_m$ defined in* (19) *is a twice continuously differentiable* ***convex local first-order surrogate*** *(see* (3)–(5)*) for $F = \sum_m f_m(\theta^T x_m)$ on the majorization domain $\Omega_M := \{\theta \,|\, a_m \leq \theta^T x_m \leq b_m, \forall m\}$ when a_m and b_m are chosen such that $\hat{\theta} \in int(\Omega_M)$. (Proof is provided in the supplementary material).*

Remark The function in (19) is a surrogate for F on Ω_M *only if* $\tilde{f}$ is *globally* convex since Jensen's inequality only holds in this case. More specifically, consider the argument of $\tilde{f}_m(\hat{\theta}^T x_m + (\theta^k - \hat{\theta}^k)^T x_m^k / r_m^k)$ and note that the argument can be outside of $[a_m, b_m]$ even if $\theta \in \Omega_M$, so in that case, Jensen's inequality would not hold if $\tilde{f}$ was not convex outside $[a_m, b_m]$. This provides another motivation for (16).

It is worth noting that for a smaller number of groups K, the surrogate in (19) will be a tighter bound to

```
Algorithm 2.2: PARALLEL BLOCK CD
for $k \leftarrow 1$ to $K$    (In parallel)
    $\theta^{k(t+1)} \in \arg\min_{\theta^k \in \Omega_F^k} \{\mathcal{S}(\theta^k; \theta^{(t)}, \Omega_M^{(t)})\}$
comment: $\mathcal{S} = \sum_m \mathcal{S}_m$
comment: $\mathcal{S}_m$ is given by Eq. (19)
```

the original objective function with the trivial case of $\mathcal{S}_m(\theta) \equiv \tilde{f}_m(\theta^T x_m)$ for all m when $K = 1$. Decreasing K will also result in harder optimization problems involving more variables for each group. On the other hand, increasing K leads to better parallelization, while resulting in a less tight bound, which can slow down the convergence per iteration but can also reduce computation time per iteration. Note that in this approach, there is complete freedom to choose any group size and to explore the above tradeoff according to the available computational resources. The resulting block-coordinate algorithm is described by Algorithm 2.1 by replacing the minimization of $\tilde{F}$ by the minimization described in Algorithm 2.2 where Ω_F^k is the projection of Ω_F onto the subspace of θ^k. We call this algorithm *Parallel Majorization-Minimization with Dynamically Restricted Domains* (**PMM-DRD**).

3 Convergence Analysis

In order for the proposed iterative procedure to be generally useful, it must converge to a local optimum or a stationary point from all initialization states without exhibiting divergence or oscillation. To understand this analysis, we first introduce the notion of a *point-to-set mapping*. A point-to-set map $\mathcal{A}$ from a set X into a set Y is defined as $\mathcal{A} : X \to \mathcal{M}(Y)$, which assigns a subset of Y to each point of X, where $\mathcal{M}(Y)$ denotes the power set of Y. Next, we introduce several definitions related to the properties of point-to-set mappings. A point-to-set map is said to be closed at $x \in X$ if $x^{(t)} \to x$ as $t \to \infty$ with $x^{(t)} \in X$ and $y^{(t)} \to y$ as $t \to \infty$ with $y^{(t)} \in \mathcal{A}(x^{(t)})$ implies $y \in \mathcal{A}(x)$. A point-to-set map is said to be closed on $W \subset X$ if it is closed at every point of W. A fixed point of the map $\mathcal{A}$ is a point x for which $x = \mathcal{A}(x)$. A *generalized* fixed point of a map is a point x for which $x \in \mathcal{A}(x)$. $\mathcal{A}$ is said to be *monotonic* with respect to ϕ whenever $y \in \mathcal{A}(x)$ implies $\phi(y) \leq \phi(x)$. Many iterative algorithms (including the ones presented here) can be described using this notion of point-to-set maps.

Let X be a set and $x^{(0)} \in X$ a given initial point. Then an algorithm $\mathcal{A}$ with initial point $x^{(0)}$ is a point-to-set map $\mathcal{A} : X \to \mathcal{M}(X)$ which generates a sequence $\{x^{(t)}\}_{t=1}^{\infty}$ via the rule $x^{(t+1)} = \mathcal{A}(x^{(t)})$, $t = 0, 1, 2, \ldots$. $\mathcal{A}$ is said to be *globally convergent* if for *any chosen initial point* $x^{(0)}$, the sequence $\{x^{(t)}\}_{t=1}^{\infty}$ converges to a point for which a necessary condition of optimality holds. The property of global convergence expresses, in a sense, the certainty that the algorithm works. It is important to stress the fact that it does not imply (contrary to what the term might suggest) convergence to a global optimum for all initial points $x^{(0)}$. We analyze Algorithm 2.1 using the above tools. We start by stating Zangwill's global convergence theorem (Zangwill, 1969, Convergence theorem A page 91).

Theorem 3.1 (Zangwill, 1969) *Let $\mathcal{A} : X \to \mathcal{M}(X)$ be a point-to-set map that given a point $x_0 \in X$ generates a sequence $\{x^{(t)}\}_{t=1}^{\infty}$ through $x^{(t+1)} \in \mathcal{A}(x^{(t)})$. Also, let a solution set $\Gamma \subset X$ be given and*

1. *All points $\{x^{(t)}\}_{t=1}^{\infty}$ are in a compact set $W \subset X$.*
2. *There is a continuous function ϕ: $X \to \mathbb{R}$ s.t.*
   (a) $x^{(t)} \notin \Gamma \Rightarrow \phi(y) < \phi(x^{(t)})$, $\forall y \in \mathcal{A}(x^{(t)})$
   (b) $x^{(t)} \in \Gamma \Rightarrow \phi(y) \leq \phi(x^{(t)})$, $\forall y \in \mathcal{A}(x^{(t)})$
   (c) $\mathcal{A}$ is closed at $x^{(t)}$ if $x^{(t)} \notin \Gamma$

Then all limit points[1] *of $\{x^{(t)}\}_{t=1}^{\infty}$ are in Γ. Furthermore, $\lim_{t\to\infty} \phi(x^{(t)}) = \phi(x^*)$ for all limit points x^*.*

The general idea in showing the global convergence of an algorithm $\mathcal{A}$ is to invoke Theorem 3.1 by appropriately defining ϕ and Γ. For an algorithm that solves the minimization problem $\min\{f(x) : x \in \Omega\}$, the solution Γ is usually chosen to be the set of corresponding stationary points and ϕ can be chosen to be the objective function itself if it is continuous. Our approach would be to define Γ as the set of generalized fixed points and then use the fact that any generalized fixed point is also a stationary point, as established by the lemma below.

Lemma 3.2 *Suppose x^* is a generalized fixed point of $\mathcal{A}$ described by Algorithm 2.1 (or Algorithm 2.1 with the minimization of $\tilde{F}$ replaced by Algorithm 2.2). Assume that the constraints as specified by Ω_F in* (1) *are qualified at x^*. Furthermore, assume that $\Omega_F = \{g_i(\theta) \leq 0, h_j(\theta) = 0,\ i = 1, ..., I,\ j = 1, ..., J\}$ where $g_i(\theta)$ and $h_j(\theta)$ are differentiable convex functions. Then, x^* is a stationary point of the program in* (1)*. (See proof in the supplementary material).*

We are now ready to state our main result.

Theorem 3.3 (Global convergence) *Let $\{\theta^{(t)}\}_{t=1}^{\infty}$ be any sequence generated by the algorithm described by*

[1] A limit point is defined as the limit of a convergent subsequence.

Algorithm 2.1 (or Algorithm 2.1 with the minimization of $\tilde{F}$ replaced by Algorithm 2.2) using any initialization point $\theta^{(0)} \in \Omega_F$. Assume Ω_F in (1) is closed and bounded and given by $\Omega_F = \{g_i(\theta) \leq 0, h_j(\theta) = 0,\ i = 1, ..., I,\ j = 1, ..., J\}$ where $g_i(\theta)$ and $h_j(\theta)$ are differentiable convex functions. Assume f and R in (2) are twice continuously differentiable functions. Then assuming suitable constraint qualifications (given by Ω_F and Ω_M), any sequence $\{\theta^{(t)}\}_{t=1}^{\infty}$ has a limit point which is a stationary point of the program in (1), i.e., it satisfies the first-order KKT optimality conditions (Proof is provided in the supplementary material).

4 Choosing the Majorization Domains

Here we describe the algorithm for choosing the majorization domains $[a_m, b_m]$ in (11). A preprocessing step involves storing all the inflection points $\mathcal{I}_m = \{z \,|\, f_m''(z) = 0\}$, points with minimum negative 2nd derivatives $\mathcal{M}_m = \{z \,|\, z \in \arg\min f_m''(z),\ f_m''(z) < 0\}$, and the corresponding values of the second derivative $\chi_m = \{f_m''(z) | z \in \mathcal{M}_m\}$, for each $m = 1, 2, \ldots, M$. We denote $\mathcal{M}_m[q]$ as the qth element in the ordered set $\{\mathcal{M}_m[q]\}_{q=1}^{Q}$ consisting of elements of $\mathcal{M}_m$ and satisfying $\mathcal{M}_m[1] \leq \mathcal{M}_m[2] \leq .... \leq \mathcal{M}_m[Q]$. We define a similar ordered set for $\mathcal{I}$. For simplicity, we assume that χ_m is a singleton set, but $\mathcal{M}_m$ can still contain multiple points. Algorithm 4.1 describes the procedure for selecting the majorization domain, where two parameters $0 < \alpha, \beta < 1$ need to be chosen. For each m, Algorithm 4.1 returns an interval $[a_m, b_m]$ for which the minimal curvature is computed and used in (11). An example for using this procedure is shown in the supplementary material. In case there are multiple local minimum values of $f''(z)$, i.e., χ_m contains multiple different values, Algorithm 4.1 is repeated for each minimum value, after which $[a_m, b_m]$ are obtained by taking the intersection of all the majorization domains.

```
Algorithm 4.1: DOMAINS($\alpha, \beta$)
  for $m \leftarrow 1$ to $M$
      $a_m = -\infty$
      $b_m = \infty$
      Compute $z_m = \theta^T x_m$
      Compute $\psi_m = f_m''(z_m)$
      if $0 < |\psi_m| < \alpha|\chi_m|$
      comment: low curved region
          $[a_m, b_m] = \text{Localize}(z_m, \beta, \mathcal{M}_m)$
      if $\alpha|\chi_m| \leq \psi_m \leq |\chi_m|$
      comment: medium curved convex region
          $[a_m, b_m] = \text{Localize}(z_m, \beta, \mathcal{I}_m)$
  return $(\{a_m, b_m\}_{m=1}^{M})$
```

```
Algorithm 4.2: LOCALIZE($z, \beta, \mathcal{P}$)
  if $z < \mathcal{P}[1]$
      $a = -\infty$
      $b = (1 - \beta)z + \beta\mathcal{P}[1]$
  else if $z > \mathcal{P}[end]$
      $b = \infty$
      $a = (1 - \beta)z + \beta\mathcal{P}[end]$
  else
      Find $i$ s.t. $\mathcal{P}[i] < z < \mathcal{P}[i + 1]$
      $a = (1 - \beta)z + \beta\mathcal{P}[i]$
      $b = (1 - \beta)z + \beta\mathcal{P}[i + 1]$
  return $(a, b)$
```

5 Numerical Results

As an application of our proposed framework for non-convex optimization, we consider two binary linear classification problems that follow the general form in Eq. (2). The first example we consider is the sigmoid-loss linear support vector machine (SVM). Standard SVM utilizes a hinge loss function as a surrogate to the 0-1 loss, which is easier to optimize and introduces a margin that allows good generalization. However, to retain a convex loss function it overly penalizes examples with large negative margins. To remedy this, Ertekin et al. (2011) proposed a ramp loss function (difference of two hinge functions). The sigmoid loss considered here is a smooth version of the ramp-loss and is given by $\ell(\alpha) = 1 - \tanh(\alpha)$ (see Fig. 1 in the supplementary material). The second example we consider is logistic regression with a log penalty (Mairal, 2015) that produces sparser solutions than the L1 penalty but makes the overall problem non-convex. The regularizer is given by $R(\theta) = \lambda \sum_{j=1}^{p-1} \log\left(\theta_j^2 + \epsilon\right)$ where $\epsilon > 0$ is chosen to prevent the log penalty from blowing up when $\theta_j = 0$. We compare the proposed method PMM-DRD, which is described by Algorithm 2.1 with Algorithm 2.2 replacing the minimization therein, against previously proposed methods in the literature for solving non-convex optimization problems such as Gradient Descent, L-BFGS (Schmidt, 2005), RMSProp (Tieleman and Hinton, 2012), AdaGrad (Duchi et al., 2011), and PSCA (Razaviyayn et al., 2014). We also compare to the case where the proposed method does not use restricted majorization domains, denoted PMM.

In the following examples we consider binary classification tasks of distinguishing between digits 3 and 5 in the MNIST[2] dataset, newsgroups 1 and 20 in the 20Newsgroup[3] dataset, and HIV positive and negative patients in the TB[4] dataset. The L-BFGS algorithm

[2] http://yann.lecun.com/exdb/mnist/

[3] http://qwone.com/ jason/20Newsgroups/

[4] ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE39941

Table 1: Classification accuracy (%) on the **training** set using SVM. LIBLINEAR (Fan et al., 2008) uses the standard linear SVM, and the other methods use the nonconvex sigmoid-loss. For the latter, we used 10 random initializations and presented the mean and standard deviation. GD=Gradient Descent, RProp=RMSProp, AGrad=AdaGrad, PMM=Parallel Majorization-Minimization, DRD=Dynamically Restricted Domain.

Method	MNIST	20 News	TB
LIBLIN	97.13	100	86.55
LBFGS	96.92 ± 0.02	99.63 ± 0.21	65.99 ± 0
CG	96.91 ± 0.026	98.36 ± 0.72	65.99 ± 0
GD	95.13 ± 1.04	95.16 ± 5.61	65.99 ± 0
PSCA	89.23 ± 0.07	79.29 ± 0.30	65.99 ± 0
RProp	96.74 ± 0.32	99.87 ± 0.09	65.99 ± 0
AGrad	94.99 ± 0.24	81.41 ± 0.63	65.99 ± 0
PMM	97.02 ± 0.02	100 ± 0	**100** ± 0
PMM-DRD	97 ± 0.01	100 ± 0	**100** ± 0

was used with cross validation to learn the regularization coefficient λ in (2) for the nonconvex model, which was then fixed for all methods. All algorithms ran till convergence, i.e., till one of the stopping criteria was met (see supplementary for details). Due to possible local minima, we used 10 random initializations and presented the mean and standard deviation.

Table 1 shows the classification accuracy of the sigmoid-loss SVM on the training set. One can see that among the methods for nonconvex optimization, the highest accuracy for all datasets is achieved by PMM-DRD. For the TB dataset, PMM significantly outperforms all the other methods. Figure 2 shows the nonconvex objective vs training time for the TB dataset, where it is seen that all methods get stuck in a bad local minimum, except for the proposed PMM. This demonstrates the susceptibility of gradient descent methods to the choice of learning rates and feature scaling, while PMM *does not require learning rates* and is also *scale-invariant*. Note that the restricted majorization domain in PMM-DRD leads to a convergence speed-up by a factor $\gtrsim 5$. Table 2 shows the classification accuracy of SVM on the test set. Table 3 shows the training accuracy for logistic regression (see Table 1 in the supplementary for test accuracy).

All experiments use a 64-core machine. In RMSProp and AdaGrad we use mini-batches which are 10% of the examples. The rest of the methods use the entire dataset (batch). PMM methods use blocks of 500 features. For PMM-DRD we use $(\alpha, \beta) = (0.1, 0.8)$ in algorithm 4.1. The minimization of the surrogates in PMM methods was done using the trust region conjugate gradient algorithm (Lin, 2008). We implemented PMM in MATLAB using C/C++ mex with OpenMP enabled. PSCA uses the cyclic version (Razaviyayn et al., 2014) with 64 processors, 40 scalar features per processor, and a learning rate that is initially 1 and decreases as $1/k^{1.01}$, where k is the iteration number.

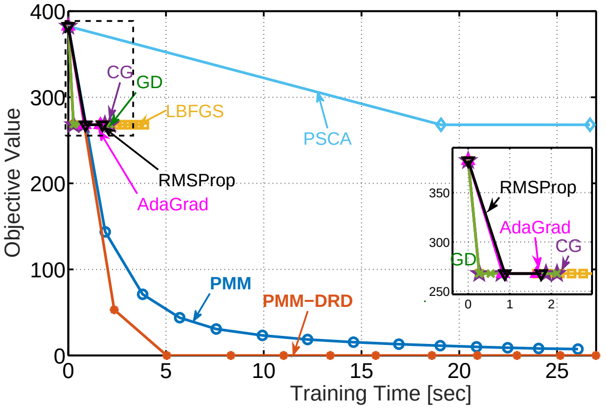

Figure 2: Objective for linear SVM with sigmoid-loss vs training time using the optimization methods mentioned in Table 1 and the TB dataset. Inset shows an enlarged view of the region inside the dashed square.

Table 2: Same as Table 1 but for the **test** set.

Method	MNIST	20 News	TB
LIBLIN	96.37	77.86	90.72
LBFGS	96.65 ± 0.05	77.19 ± 0.9	72.16 ± 0
CG	96.63 ± 0.035	75.45 ± 1.873	72.16 ± 0
GD	95.7 ± 0.67	75.82 ± 3.12	72.16 ± 0
PSCA	90.89 ± 0.07	68.82 ± 0.35	72.16 ± 0
RProp	96.38 ± 0.3	81.92 ± 0.81	72.16 ± 0
AGrad	94.99 ± 0.24	81.41 ± 0.63	72.16 ± 0
PMM	96.95 ± 0	71.81 ± 0.09	91.34 ± 0.5
PMM-DRD	96.98 ± 0.03	78.2 ± 0.08	**91.75** ± 0

Table 3: Same as Table 1 but for Logistic regression. LIBLIN uses an L1 penalty and the rest of the methods use a non-convex log-penalty.

Method	MNIST	20 News	TB
LIBLIN	97.33	98.48	81.44
LBFGS	97.14 ± 0.03	100± 0	100 ± 0
CG	97.81 ± 0.11	100 ± 0	100 ± 0
GD	95.2 ± 1.26	97.82 ± 3.68	89.09 ± 0.77
PSCA	88.4 ± 0	97.58 ± 0.11	53.55 ± 0
RProp	96.74 ± 0.32	100 ± 0	56.43 ± 17.43
AGrad	94.54 ± 0.16	100 ± 0	56.18 ± 17.21
PMM	98.29 ± 0	100 ± 0	100 ± 0
PMM-DRD	98.64 0 ± 0.01	100 ± 0	100 ± 0

Acknowledgements

This research was supported in part by ARO, DARPA, DOE, NGA, ONR and NSF.

References

K. Lange, D. R. Hunter, and I. Yang (2000). Optimization Transfer Using Surrogate Objective Functions. *Journal of Computational and Graphical Statistics* 9(1):1–20.

G. R. Lanckriet, and B. K. Sriperumbudur (2009). On the Convergence of the Concave-Convex Procedure. *Advances in Neural Information Processing Systems* 22:1759–1767.

J. Mairal (2015). Incremental Majorization-Minimization Optimization with Application to Large-Scale Machine Learning. *SIAM Journal on Optimization* 25(2):829-855.

S. Ahn, J. A. Fessler, D. Blatt, and A. O. Hero (2006). Convergent Incremental Optimization Transfer Algorithms: Application to Tomography. *IEEE Trans. on Medical Imaging* 25(3):283–296.

M .W. Jacobson, J. A. Fessler (2007). An Expanded Theoretical Treatment of Iteration-Dependent Majorize-Minimize Algorithms. *IEEE Trans. on Image Processing* 16(10):2411–2422.

A. P. Dempster, N. M. Laird, D. B. Rubin (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society: Series B* 39:1–38.

R. M. Neal and G. E. Hinton (1998). A View of the EM Algorithm that Justifies Incremental, Sparse and Other Variants. *Learning in Graphical Models*, Springer Netherlands, 355-368.

Y. Kaganovsky, S. Han, S. Degirmenci, D. G. Politte, D. J. Brady, J. A. O'Sullivan, and L. Carin (2015). Alternating Minimization Algorithm with Automatic Relevance Determination for Transmission Tomography under Poisson Noise. *SIAM Journal on Imaging Sciences* 8(3):2087-2132.

S. Sotthivirat and J. A. Fessler (2002). Image Recovery Using Partitioned-Separable Paraboloidal Surrogate Coordinate Ascent Algorithms. *IEEE Trans. on Image Processing* 11(3):306-317.

D. D. Lee and H. S. Seung (2001). Algorithms for Non-Negative Matrix Factorization, *Advances in Neural Information Processing Systems* 13:556–562.

M. Razaviyayn, M. Hong,Z. Q. Luo (2013). A Unified Convergence Analysis of Block Successive Minimization Methods for Nonsmooth Optimization. *SIAM Journal on Optimization* 23(2):1126–1153.

M. Razaviyayn, M. Hong, Z. Q. Luo, and J. S. Pang (2014). Parallel Successive Convex Approximation for Nonsmooth Nonconvex Optimization. *Advances in Neural Information Processing Systems* 27:1440–1448.

A. Beck and M. Teboulle (2009). A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences* 2:183-202.

L. Combettes and J. C. Pesquet (2011). Proximal Splitting Methods in Signal Processing. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Springer, New York, 185-212.

H. Erdogan, J. A. Fessler (1999). Ordered Subsets Algorithms for Transmission Tomography. *Physics in Medicine and Biology* 44(11):2835.

S. Ertekin, L. Leon, and C. L. Giles (2011). Nonconvex Online Support Vector Machines. *IEEE Trans, on Pattern Analysis and Machine Intelligence* 33(2):368–381.

S. P. Boyd and L. Vandenberghe (2004). *Convex Optimization*, Cambridge University Press.

A. R. De Pierro (1994). A Modified Expectation Maximization Algorithm for Penalized Likelihood Estimation in Emission Tomography. *IEEE Trans. Medical Imaging* 14(1):132–137.

W. I. Zangwill (1969). *Nonlinear Programming: A Unified Approach.* Prentice-Hall, Englewood Cliffs, N.J..

M. Schmidt (2005). MinFunc: Unconstrained Differentiable Multivariate Optimization in Matlab. Software available at `http://www.cs.ubc.ca/~schmidtm/Software/minFunc.htm`.

T. Tieleman, and G. Hinton (2012). Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning.*

J. Duchi, E. Hazan, and Y. Singer (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization, *Journal of Machine Learning Research* 12(39):2121–2159.

R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin (2008). LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research* 9:1871–1874. Software available at `https://www.csie.ntu.edu.tw/~cjlin/liblinear/`.

C. J. Lin, R. C. Weng, and S. .S Keerthi (2008). Trust Region Newton Method for Large-Scale Logistic Regression. *Journal of Machine Learning Research* 9:627–650.