Optimization with First-Order Surrogate Functions

Julien Mairal JULIEN.MAIRAL@INRIA.FR
INRIA LEAR Project-Team, Grenoble, France

Abstract

In this paper, we study optimization methods consisting of iteratively minimizing surrogates of an objective function. By proposing several algorithmic variants and simple convergence analyses, we make two main contributions. First, we provide a unified viewpoint for several first-order optimization techniques such as accelerated proximal gradient, block coordinate descent, or Frank-Wolfe algorithms. Second, we introduce a new incremental scheme that experimentally matches or outperforms state-of-the-art solvers for large-scale optimization problems typically arising in machine learning.

1. Introduction

The principle of iteratively minimizing a majorizing surrogate of an objective function is often called *majorization-minimization* (Lange et al., 2000). Each iteration drives the objective function downhill, thus giving the hope of finding a local optimum. A large number of existing procedures can be interpreted from this point of view. This is for instance the case of gradient-based or proximal methods (see Nesterov, 2007; Beck & Teboulle, 2009; Wright et al., 2009), EM algorithms (see Neal & Hinton, 1998), DC programming (Horst & Thoai, 1999), boosting (Collins et al., 2002; Della Pietra et al., 2001), and some variational Bayes techniques (Wainwright & Jordan, 2008; Seeger & Wipf, 2010). The concept of "surrogate" has also been used successfully in the signal processing literature about sparse optimization (Daubechies et al., 2004; Gasso et al., 2009) and matrix factorization (Lee & Seung, 2001; Mairal et al., 2010).

In this paper, we are interested in generalizing the majorization-minimization principle. Our goal is both to discover new algorithms, and to draw connections with existing methods. We focus our study on "first-order surrogate functions", which consist of approximating a possibly non-smooth objective function up to a smooth error. We present several schemes exploiting such surrogates, and analyze their convergence properties: asymptotic stationary point conditions for non-convex problems, and convergence rates for convex ones. More precisely, we successively study:

- a generic majorization-minimization approach;
- a randomized block coordinate descent algorithm (see Tseng & Yun, 2009; Shalev-Shwartz & Tewari, 2009; Nesterov, 2012; Richtárik & Takáč, 2012);
- an accelerated variant for convex problems inspired by Nesterov (2004); Beck & Teboulle (2009);
- a generalization of the "Frank-Wolfe" conditional gradient method (see Zhang, 2003; Harchaoui et al., 2013; Hazan & Kale, 2012; Zhang et al., 2012);
- a new incremental scheme, which we call MISO.[1]

We present in this work a unified view for analyzing a large family of algorithms with simple convergence proofs and strong guarantees. In particular, all the above optimization methods except Frank-Wolfe have linear convergence rates for minimizing strongly convex objective functions. This is remarkable for MISO, the new incremental scheme derived from our framework; to the best of our knowledge, only two recent incremental algorithms share such a property: the *stochastic average gradient* method (SAG) of Le Roux et al. (2012), and the *stochastic dual coordinate ascent* method (SDCA) of Shalev-Schwartz & Zhang (2012). Our scheme MISO is inspired in part by these two works, but yields different update rules than SAG or SDCA.

After we present and analyze the different optimization schemes, we conclude the paper with numerical experiments focusing on the scheme MISO. We show that in most cases MISO matches or outperforms cutting-edge solvers for large-scale ℓ_2- and ℓ_1-regularized logistic regression (Bradley et al., 2011; Beck & Teboulle, 2009; Le Roux et al., 2012; Fan et al., 2008; Bottou, 2010).

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

[1] *Minimization by Incremental Surrogate Optimization.*

2. Basic Optimization Scheme

Given a convex subset Θ of $\mathbb{R}^p$ and a continuous function $f : \mathbb{R}^p \to \mathbb{R}$, we are interested in solving

$$\min_{\theta \in \Theta} f(\theta),$$

where we assume, to simplify, that f is bounded below. Our goal is to study the majorization-minimization scheme presented in Algorithm 1 and its variants. This procedure relies on the concept of surrogate functions, which are minimized instead of f at every iteration.[2]

Algorithm 1 Basic Scheme

input $\theta_0 \in \Theta$; N (number of iterations).
1: **for** $n = 1, \dots, N$ **do**
2: Compute a surrogate function g_n of f near θ_{n-1};
3: Update solution: $\theta_n \in \arg\min_{\theta \in \Theta} g_n(\theta)$.
4: **end for**
output θ_N (final estimate);

For this approach to be successful, we intuitively need surrogates that approximate well the objective f and that are easy to minimize. In this paper, we focus on "first-order surrogate functions" defined below, which will be shown to have "good" theoretical properties.

Definition 2.1 (**First-Order Surrogate**). *A function $g : \mathbb{R}^p \to \mathbb{R}$ is a first-order surrogate of f near κ in Θ when the following conditions are satisfied:*

- ***Majorization**: we have $g(\theta') \geq f(\theta')$ for all θ' in $\arg\min_{\theta \in \Theta} g(\theta)$. When the more general condition $g \geq f$ holds, we say that g is a **majorant** function;*
- ***Smoothness**: the approximation error $h \triangleq g - f$ is differentiable, and its gradient is L-Lipschitz continuous. Moreover, we have $h(\kappa) = 0$ and $\nabla h(\kappa) = 0$.*

We denote by $\mathcal{S}_L(f, \kappa)$ the set of such surrogates, and by $\mathcal{S}_{L,\rho}(f, \kappa)$ the subset of ρ-strongly convex surrogates.

First-order surrogates have a few simple properties, which form the building block of our analyses:

Lemma 2.1 (**Basic Properties - Key Lemma**). *Let g be in $\mathcal{S}_L(f, \kappa)$ for some κ in Θ. Define $h \triangleq g - f$ and let θ' be in $\arg\min_{\theta \in \Theta} g(\theta)$. Then, for all θ in Θ,*

- $|h(\theta)| \leq \frac{L}{2}\|\theta - \kappa\|_2^2$;
- $f(\theta') \leq f(\theta) + \frac{L}{2}\|\theta - \kappa\|_2^2$.

Assume that g is in $\mathcal{S}_{L,\rho}(f, \kappa)$, then, for all θ in Θ,

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

[2] Note that this concept differs from the machine learning terminology, where a "surrogate" often denotes a fixed convex upper bound of the nonconvex $(0-1)$-loss.

The proof of this lemma is relatively simple but for space limitation reasons, all proofs in this paper are provided as supplemental material. With Lemma 2.1 in hand, we now study the properties of Algorithm 1.

2.1. Convergence Analysis

For general non-convex problems, proving convergence to a global (or local) minimum is out of reach, and classical analyses study instead asymptotic stationary point conditions (see, e.g., Bertsekas, 1999). To do so, we make the mild assumption that for all θ, θ' in Θ, the directional derivative $\nabla f(\theta, \theta' - \theta)$ of f at θ in the direction $\theta' - \theta$ exists. A classical necessary first-order condition (see Borwein & Lewis, 2006) for θ to be a local minimum of f is to have $\nabla f(\theta, \theta' - \theta)$ non-negative for all θ' in Θ. This naturally leads us to consider the following asymptotic condition to assess the quality of a sequence $(\theta_n)_{n \geq 0}$ for non-convex problems:

Definition 2.2 (**Asymptotic Stationary Point**). *A sequence $(\theta_n)_{n \geq 0}$ satisfies an asymptotic stationary point condition if*

$$\liminf_{n \to +\infty} \inf_{\theta \in \Theta} \frac{\nabla f(\theta_n, \theta - \theta_n)}{\|\theta - \theta_n\|_2} \geq 0.$$

In particular, if f is differentiable on $\mathbb{R}^p$ and $\Theta = \mathbb{R}^p$, this condition implies $\lim_{n \to +\infty} \|\nabla f(\theta_n)\|_2 = 0$.

Building upon this definition, we now give a first convergence result about Algorithm 1.

Proposition 2.1 (**Non-Convex Analysis**). *Assume that the surrogates g_n from Algorithm 1 are in $\mathcal{S}_L(f, \theta_{n-1})$ and are majorant or strongly convex. Then, $(f(\theta_n))_{n \geq 0}$ monotonically decreases and $(\theta_n)_{n \geq 0}$ satisfies an asymptotic stationary point condition.*

Convergence results for non-convex problems are by nature weak. This is not the case when f is convex. In the next proposition, we obtain convergence rates by following a proof technique from Nesterov (2007) originally designed for proximal gradient methods.

Proposition 2.2 (**Convex Analysis for $\mathcal{S}_L(f, \kappa)$**). *Assume that f is convex and that for some $R > 0$,*

$$\|\theta - \theta^\star\|_2 \leq R \quad \text{for all } \theta \in \Theta \ \text{ s.t. } \ f(\theta) \leq f(\theta_0), \quad (1)$$

where $\theta^\star$ is a minimizer of f on Θ. When the surrogate g_n in Algorithm 1 are in $\mathcal{S}_L(f, \theta_{n-1})$, we have

$$f(\theta_n) - f^\star \leq \frac{2LR^2}{n+2} \quad \text{for all } n \geq 1,$$

where $f^\star \triangleq f(\theta^\star)$. Assume now that f is μ-strongly convex. Regardless of condition (1), we have

$$f(\theta_n) - f^\star \leq \beta^n (f(\theta_0) - f^\star) \quad \text{for all } n \geq 1,$$

where $\beta \triangleq \frac{L}{\mu}$ if $\mu > 2L$ or $\beta \triangleq \left(1 - \frac{\mu}{4L}\right)$ otherwise.

The result of Proposition 2.2 is interesting in the sense that it provides sharp theoretical results without making strong assumption on the surrogate functions. The next proposition shows that slightly better rates can be obtained when the surrogates are strongly convex.

Proposition 2.3 (Convex Analysis for $\mathcal{S}_{L,\rho}(f,\kappa)$). *Assume that f is convex and let $\theta^\star$ be a minimizer of f on Θ. When the surrogates g_n of Algorithm 1 are in $\mathcal{S}_{L,\rho}(f,\theta_{n-1})$ with $\rho \geq L$, we have for all $n \geq 1$,*

$$f(\theta_n) - f^\star \leq \frac{L\|\theta_0 - \theta^\star\|_2^2}{2n}.$$

When f is μ-strongly convex, we have for all $n \geq 1$,

$$\left\{ \begin{array}{rcl} \|\theta_n - \theta^\star\|_2^2 & \leq & \left(\frac{L}{\rho+\mu}\right)^n \|\theta_0 - \theta^\star\|_2^2 \\ f(\theta_n) - f^\star & \leq & \left(\frac{L}{\rho+\mu}\right)^{n-1} \frac{L\|\theta_0-\theta^\star\|_2^2}{2} \end{array} \right. .$$

Note that the condition $\rho \geq L$ is relatively strong; it can indeed be shown that f is necessarily $(\rho-L)$-strongly convex if $\rho > L$, and convex if $\rho = L$. The fact that making stronger assumptions yields better convergence rates suggests that going beyond first-order surrogates could provide even sharper results. This is confirmed in the next proposition:

Proposition 2.4 (Second-Order Surrogates). *Make similar assumptions as in Proposition 2.2, and also assume that the error functions $h_n \triangleq g_n - f$ are twice differentiable, that their Hessians $\nabla^2 h_n$ are M-Lipschitz, and that $\nabla^2 h_n(\theta_{n-1}) = 0$ for all n. Then,*

$$f(\theta_n) - f^\star \leq \frac{9MR^3}{2(n+3)^2} \quad \text{for all } n \geq 1.$$

If f is μ-strongly convex, the convergence rate is superlinear with order 3/2.

Consistently with this proposition, similar rates were obtained by Nesterov & Polyak (2006) for the Newton method with cubic regularization, which involve second-order surrogates. In the next section, we focus again on first-order surrogates, and present simple mechanisms to build them. The proofs of the different claims are provided in the supplemental material.

2.2. Examples of Surrogate Functions

Lipschitz Gradient Surrogates.
When f is differentiable and ∇f is L-Lipschitz, f admits the following majorant surrogate in $\mathcal{S}_{2L,L}(f,\kappa)$:

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

In addition, when f is convex, g is in $\mathcal{S}_{L,L}(f,\kappa)$, and when f is μ-strongly convex, g is in $\mathcal{S}_{L-\mu,L}(f,\kappa)$. Note also that minimizing g amounts to performing a classical classical gradient descent step $\theta' \leftarrow \kappa - \frac{1}{L}\nabla f(\kappa)$.

Proximal Gradient Surrogates.
Assume that f splits into $f = f_1 + f_2$, where f_1 is differentiable with a L-Lipschitz gradient. Then, f admits the following majorant surrogate in $\mathcal{S}_{2L}(f,\kappa)$:

$$g : \theta \mapsto f_1(\kappa) + \nabla f_1(\kappa)^\top(\theta - \kappa) + \frac{L}{2}\|\theta - \kappa\|_2^2 + f_2(\theta).$$

The approximation error $g - f$ is indeed the same as in the previous paragraph and thus:

- when f_1 is convex, g is in $\mathcal{S}_L(f,\kappa)$. If f_2 is also convex, g is in $\mathcal{S}_{L,L}(f,\kappa)$.
- when f_1 is μ-strongly convex, g is in $\mathcal{S}_{L-\mu}(f,\kappa)$. If f_2 is also convex, g is in $\mathcal{S}_{L-\mu,L}(f,\kappa)$.

Minimizing g amounts to performing a proximal gradient step (see Nesterov, 2007; Beck & Teboulle, 2009).

DC Programming Surrogates.
Assume that $f = f_1 + f_2$, where f_2 is concave and differentiable with a L_2-Lipschitz gradient. Then, the following function g is a majorant surrogate in $\mathcal{S}_{L_2}(f,\kappa)$:

$$g : \theta \mapsto f_1(\theta) + f_2(\kappa) + \nabla f_2(\kappa)^\top(\theta - \kappa).$$

Such a surrogate forms the root of DC- (difference of convex functions)-programming (see Horst & Thoai, 1999). It is also indirectly used in reweighted-ℓ_1 algorithms (Candès et al., 2008) for minimizing on $\mathbb{R}_+^p$ a cost function of the form $\theta \mapsto f_1(\theta) + \lambda \sum_{i=1}^p \log(\theta_i + \varepsilon)$.

Variational Surrogates.
Let f be a real-valued function defined on $\mathbb{R}^{p_1} \times \mathbb{R}^{p_2}$. Let $\Theta_1 \subseteq \mathbb{R}^{p_1}$ and $\Theta_2 \subseteq \mathbb{R}^{p_2}$ be two convex sets. Define $\tilde{f}$ as $\tilde{f}(\theta_1) \triangleq \min_{\theta_2 \in \Theta_2} f(\theta_1, \theta_2)$ and assume that

- $\theta_1 \mapsto f(\theta_1, \theta_2)$ is differentiable for all θ_2 in Θ_2;
- $\theta_2 \mapsto \nabla_1 f(\theta_1, \theta_2)$ is L-Lipschitz for all θ_1 in $\mathbb{R}^{p_1}$;[3]
- $\theta_1 \mapsto \nabla_1 f(\theta_1, \theta_2)$ is L'-Lipschitz for all θ_2 in Θ_2;
- $\theta_2 \mapsto f(\theta_1, \theta_2)$ is μ-strongly convex for all θ_1 in $\mathbb{R}^{p_1}$.

Let us fix κ_1 in Θ_1. Then, the following function is a majorant surrogate in $\mathcal{S}_{2L''}(\tilde{f},\kappa)$ for some $L'' > 0$:

$$g : \theta_1 \mapsto f(\theta_1, \kappa_2^\star) \text{ with } \kappa_2^\star \triangleq \operatorname*{arg\,min}_{\theta_2 \in \Theta_2} \tilde{f}(\kappa_1, \theta_2).$$

When f is jointly convex in θ_1 and θ_2, $\tilde{f}$ is itself convex and we can choose $L'' = L'$. Algorithm 1 becomes a block-coordinate descent procedure with two blocks.

Saddle Point Surrogates.
Let us make the same assumptions as in the previous paragraph but with the following differences:

[3] The notation ∇_1 denotes the gradient w.r.t. θ_1.

- $\theta_2 \mapsto f(\theta_1, \theta_2)$ is μ-strongly concave for all θ_1 in $\mathbb{R}^{p_1}$;
- $\theta_1 \mapsto f(\theta_1, \theta_2)$ is convex for all θ_2 in Θ_2;
- $\tilde{f}(\theta_1) \triangleq \max_{\theta_2 \in \Theta_2} f(\theta_1, \theta_2)$.

Then, $\tilde{f}$ is convex and the function below is a majorant surrogate in $\mathcal{S}_{2L''}(\tilde{f}, \kappa_1)$:

$$g : \theta_1 \mapsto f(\theta_1, \kappa_2^\star) + \frac{L''}{2}\|\theta_1 - \kappa_1\|_2^2,$$

where $L'' \triangleq \max(2L^2/\mu, L')$. When $\theta_1 \mapsto f(\theta_1, \theta_2)$ is affine, we can instead choose $L'' \triangleq L^2/\mu$.

Jensen Surrogates.
Jensen's inequality provides a natural mechanism to obtain surrogates for convex functions. Following the presentation of Lange et al. (2000), we consider a convex function $f : \mathbb{R} \mapsto \mathbb{R}$, a vector $\mathbf{x}$ in $\mathbb{R}^p$, and define $\tilde{f} : \mathbb{R}^p \to \mathbb{R}$ as $\tilde{f}(\theta) \triangleq f(\mathbf{x}^\top \theta)$ for all θ. Let $\mathbf{w}$ be a weight vector in $\mathbb{R}_+^p$ such that $\|\mathbf{w}\|_1 = 1$ and $\mathbf{w}_i \neq 0$ whenever $\mathbf{x}_i \neq 0$. Then, we define for any κ in $\mathbb{R}^p$

$$g : \theta \mapsto \sum_{i=1}^p \mathbf{w}_i f\left(\frac{\mathbf{x}_i}{\mathbf{w}_i}(\theta_i - \kappa_i) + \mathbf{x}^\top \kappa\right),$$

When f is differentiable with an L-Lipschitz gradient, and $\mathbf{w}_i \triangleq |\mathbf{x}_i|^\nu / \|\mathbf{x}\|_\nu^\nu$, then g is in $\mathcal{S}_{L'}(\tilde{f}, \kappa)$ with

- $L' = L\|\mathbf{x}\|_\infty^2 \|\mathbf{x}\|_0$ for $\nu = 0$;
- $L' = L\|\mathbf{x}\|_\infty \|\mathbf{x}\|_1$ for $\nu = 1$;
- $L' = L\|\mathbf{x}\|_2^2$ for $\nu = 2$.

As far as we know, the convergence rates we provide when using such surrogates are new. We also note that Jensen surrogates have been successfully used in machine learning. For instance, Della Pietra et al. (2001) interpret boosting procedures under this point of view through the concept of *auxiliary functions*.

Quadratic Surrogates.
When f is twice differentiable and admits a matrix $\mathbf{H}$ such that $\mathbf{H} - \nabla^2 f$ is always positive definite, the following function is a first-order majorant surrogate:

$$g : \theta \mapsto f(\kappa) + \nabla f(\kappa)^\top (\theta - \kappa) + \frac{1}{2}(\theta - \kappa)^\top \mathbf{H}(\theta - \kappa).$$

The Lipschitz constant of $\nabla(g - f)$ is the largest eigenvalue of $\mathbf{H} - \nabla^2 f(\theta)$ over Θ. Such surrogates appear frequently in the statistics and machine learning literature (Böhning & Lindsay, 1988; Khan et al., 2010).

We have shown that there are many rules to build first-order surrogates. Choosing one instead of another mainly depends on how easy it is to build the surrogate (do we need to estimate an a priori unknown Lipschitz constant?), and on how cheaply it can be minimized.

3. Block Coordinate Scheme

In this section, we introduce a block coordinate descent extension of Algorithm 1 under the assumptions that

- Θ is separable—that is, it can be written as a Cartesian product $\Theta = \Theta^1 \times \Theta^2 \times \ldots \times \Theta^k$;
- the surrogates g_n are separable into k components:

$$g_n(\theta) = \sum_{i=1}^k g_n^i(\theta^i) \quad \text{for} \quad \theta = (\theta^1, \ldots, \theta^k) \in \Theta.$$

We present a randomized procedure in Algorithm 2 following Tseng & Yun (2009); Shalev-Shwartz & Tewari (2009); Nesterov (2012); Richtárik & Takáč (2012).

Algorithm 2 Block Coordinate Descent Scheme

input $\theta_0 = (\theta_0^1, \ldots, \theta_0^k) \in \Theta = (\Theta^1 \times \ldots \times \Theta^k)$; N.
1: **for** $n = 1, \ldots, N$ **do**
2: Choose a separable surrogate g_n of f near θ_{n-1};
3: Randomly pick up one block $\hat{\imath}_n$ and update $\theta_n^{\hat{\imath}_n}$:

$$\theta_n^{\hat{\imath}_n} \in \operatorname*{arg\,min}_{\theta^{\hat{\imath}_n} \in \Theta^{\hat{\imath}_n}} g_n^{\hat{\imath}_n}(\theta^{\hat{\imath}_n}).$$

4: **end for**
output $\theta_N = (\theta_N^1, \ldots, \theta_N^k)$ (final estimate);

As before, we first study the convergence for non-convex problems. The next proposition shows that similar guarantees as for Algorithm 1 can be obtained.

Proposition 3.1 (Non-Convex Analysis). *Assume that the functions g_n are majorant surrogates in $\mathcal{S}_L(f, \theta_{n-1})$. Assume also that θ_0 is the minimizer of a majorant surrogate function in $\mathcal{S}_L(f, \theta_{-1})$ for some θ_{-1} in Θ. Then, the conclusions of Proposition 2.1 hold with probability one.*

Under convexity assumptions on f, the next two propositions give us expected convergence rates.

Proposition 3.2 (Convex Analysis for $\mathcal{S}_L(f, \kappa)$). *Make the same assumptions as in Proposition 2.2 and define $\delta \triangleq \frac{1}{k}$. When the surrogate functions g_n in Algorithm 2 are majorant and in $\mathcal{S}_L(f, \theta_{n-1})$, the sequence $(f(\theta_n))_{n \geq 0}$ almost surely converges to $f^\star$ and*

$$\mathbb{E}[f(\theta_n) - f^\star] \leq \frac{2LR^2}{2 + \delta(n - n_0)} \quad \text{for all } n \geq n_0,$$

where $n_0 \triangleq \left\lceil \log\left(\frac{2(f(\theta_0) - f^\star)}{LR^2} - 1\right) / \log\left(\frac{1}{1-\delta}\right) \right\rceil$ *if* $f(\theta_0) - f^\star > LR^2$ *and* $n_0 \triangleq 0$ *otherwise. Assume now that f is μ-strongly convex. Then, we have instead an expected linear convergence rate*

$$\mathbb{E}[f(\theta_n) - f^\star] \leq ((1 - \delta) + \delta\beta)^n (f(\theta_0) - f^\star),$$

where $\beta \triangleq \frac{L}{\mu}$ *if* $\mu > 2L$ *or* $\beta \triangleq \left(1 - \frac{\mu}{4L}\right)$ *otherwise.*

Proposition 3.3 (Convex Analysis for $\mathcal{S}_{L,\rho}(f,\kappa)$). *Assume that f is convex. Define $\delta \triangleq \frac{1}{k}$. Choose majorant surrogates g_n in $\mathcal{S}_{L,\rho}(f,\theta_{n-1})$ with $\rho \geq L$, then $(f(\theta_n))_{n\geq 0}$ almost surely converges to $f^\star$ and we have*

$$\mathbb{E}[f(\theta_n) - f^\star] \leq \frac{C_0}{(1-\delta)+\delta n} \quad \text{for all} \quad n \geq 1,$$

with $C_0 \triangleq (1-\delta)(f(\theta_0)-f^\star)+\frac{(1-\delta)\rho+\delta L}{2}\|\theta_0-\theta^\star\|_2^2$. Assume now that f is μ-strongly convex, then we have an expected linear convergence rate

$$\left\{ \begin{array}{rcl} \frac{L}{2}\mathbb{E}[\|\theta^\star-\theta_n\|_2^2] & \leq & C_0\left((1-\delta)+\delta\frac{L}{\rho+\mu}\right)^n \\ \mathbb{E}[f(\theta_n)-f^\star] & \leq & \frac{C_0}{\delta}\left((1-\delta)+\delta\frac{L}{\rho+\mu}\right)^{n-1} \end{array} \right. .$$

The quantity $\delta = 1/k$ represents the probability for a block to be updated during an iteration. Note that updating all blocks ($\delta = 1$) gives the same results as in Section 2. Linear convergence for strongly convex objectives with block coordinate descent is classical since the works of Tseng & Yun (2009); Nesterov (2012). Results of the same nature have also been obtained by Richtárik & Takáč (2012) for composite functions.

4. Frank-Wolfe Scheme

In this section, we show how to use surrogates to generalize the Frank-Wolfe method, an old convex optimization technique that has regained some popularity in machine learning (Zhang, 2003; Harchaoui et al., 2013; Hazan & Kale, 2012; Zhang et al., 2012). We present this approach in Algorithm 3.

Algorithm 3 Frank-Wolfe Scheme

input $\theta_0 \in \Theta$; N (number of iterations).

1: **for** $n = 1, \ldots, N$ **do**

2: Let g_n be a majorant surrogate in $\mathcal{S}_{L,L}(f,\theta_{n-1})$.

3: Compute a search direction:

$$\nu_n \in \arg\min_{\theta\in\Theta}\left[g_n(\theta) - \frac{L}{2}\|\theta-\theta_{n-1}\|_2^2\right].$$

4: Line search: $\alpha^\star \triangleq \arg\min_{\alpha\in[0,1]} g_n(\alpha\nu_n+(1-\alpha)\theta_{n-1})$.

5: Update solution: $\theta_n \triangleq \alpha^\star\nu_n + (1-\alpha^\star)\theta_{n-1}$.

6: **end for**

output θ_N (final estimate);

When f is smooth and the "gradient Lipschitz based surrogates" from Section 2.2 are used, Algorithm 3 becomes the classical Frank-Wolfe method.[4] Our point of view is however more general since it allows for example to use "proximal gradient surrogates". The next proposition gives a convergence rate.

[4]Note that the classical Frank-Wolfe algorithm performs in fact the line search over the function f and not g_n.

Proposition 4.1 (Convex Analysis). *Assume that f is convex and that Θ is bounded. Call $R \triangleq \max_{\theta_1,\theta_2\in\Theta}\|\theta_1-\theta_2\|_2$ the diameter of Θ. Then, the sequence $(f(\theta_n))_{n\geq 0}$ provided by Algorithm 3 converges to the minimum $f^\star$ of f over Θ and*

$$f(\theta_n) - f^\star \leq \frac{2LR^2}{n+2} \quad \text{for all } n \geq 1.$$

Other extensions of Algorithm 3 can also easily be designed by using our framework. We present for instance in the supplemental material a randomized block Frank-Wolfe algorithm, revisiting the recent work of Lacoste-Julien et al. (2013).

5. Accelerated Scheme

A popular scheme for convex optimization is the accelerated proximal gradient method (Nesterov, 2007; Beck & Teboulle, 2009). By using surrogate functions, we exploit similar ideas in Algorithm 4. When using the "Lipschitz gradient surrogates" of Section 2.2, Algorithm 4 is exactly the scheme 2.2.19 of Nesterov (2004). When using the "proximal gradient surrogate" and when $\mu = 0$, it is equivalent to the FISTA method of Beck & Teboulle (2009). Algorithm 4 consists of iteratively minimizing a surrogate computed at a point κ_{n-1} extrapolated from θ_{n-1} and θ_{n-2}. It results in better convergence rates, as shown in the next proposition by adapting a proof technique of Nesterov (2004).

Algorithm 4 Accelerated Scheme

input $\theta_0 \in \Theta$; N; μ (strong convexity parameter);

1: Initialization: $\kappa_0 \triangleq \theta_0$; $a_0 = 1$;

2: **for** $n = 1, \ldots, N$ **do**

3: Choose a surrogate g_n in $\mathcal{S}_{L,L+\mu}(f,\kappa_{n-1})$;

4: Update solution: $\theta_n \triangleq \arg\min_{\theta\in\Theta} g_n(\theta)$;

5: Compute $a_n \geq 0$ such that:

$$a_n^2 = (1-a_n)a_{n-1}^2 + \tfrac{\mu}{L+\mu}a_n;$$

6: Set $\beta_n \triangleq \frac{a_{n-1}(1-a_{n-1})}{a_{n-1}^2+a_n}$ and update κ:

$$\kappa_n \triangleq \theta_n + \beta_n(\theta_n - \theta_{n-1});$$

7: **end for**

output θ_N (final estimate);

Proposition 5.1 (Convex Analysis). *Assume that f is convex. When $\mu = 0$, the sequence $(\theta_n)_{n\geq 0}$ provided by Algorithm 4 satisfies for all $n \geq 1$,*

$$f(\theta_n) - f^\star \leq \frac{2L\|\theta_0-\theta^\star\|_2^2}{(n+2)^2}.$$

When f is μ-strongly convex, we have instead a linear

convergence rate: for $n \geq 1$,

$$f(\theta_n) - f^\star \leq \left(1 - \sqrt{\frac{\mu}{L+\mu}}\right)^{n-1} \frac{L\|\theta_0 - \theta^\star\|_2^2}{2}.$$

6. Incremental Scheme

This section is devoted to objective functions f that split into many components:

$$f(\theta) = \frac{1}{T}\sum_{t=1}^{T} f^t(\theta). \tag{2}$$

The most classical method exploiting such a structure when f is smooth is probably the stochastic gradient descent (SGD) and its variants (see Bottou, 2010). It consists of drawing at iteration n an index $\hat{t}_n$ and updating the solution as $\theta_n \leftarrow \theta_{n-1} - \eta_n \nabla f^{\hat{t}_n}(\theta_{n-1})$ with a scalar η_n. Another popular algorithm is the *stochastic mirror descent* (see Juditsky & Nemirovski, 2011) for general non-smooth convex problems, a setting we do not consider in this paper since non-smooth functions do not always admit first-order surrogates.

Recently, it was shown by Shalev-Schwartz & Zhang (2012) and Le Roux et al. (2012) that linear convergence rates could be obtained for strongly convex functions f^t. The SAG algorithm of Le Roux et al. (2012) for smooth unconstrained optimization is an approximate gradient descent strategy, where an estimate of ∇f is incrementally updated at each iteration. The work of Shalev-Schwartz & Zhang (2012) for composite optimization is a dual coordinate ascent method called SDCA which performs incremental updates in the primal (2). Unlike SGD, both SAG and SDCA require storing information about past iterates.

In a different context, incremental EM algorithms have been proposed by Neal & Hinton (1998), where surrogates of a log-likelihood are incrementally updated. By using similar ideas, we present in Algorithm 5 a scheme for solving (2), which we call MISO. In the next propositions, we study its convergence properties.

Algorithm 5 Incremental Scheme MISO

input $\theta_0 \in \Theta$; N (number of iterations).

1: Choose surrogates g_0^t of f^t near θ_0 for all t;
2: **for** $n = 1, \ldots, N$ **do**
3: Randomly pick up one index $\hat{t}_n$ and choose a surrogate $g_n^{\hat{t}_n}$ of $f^{\hat{t}_n}$ near θ_{n-1}. Set $g_n^t \triangleq g_{n-1}^t$ for $t \neq \hat{t}_n$;
4: Update solution: $\theta_n \in \arg\min_{\theta\in\Theta} \frac{1}{T}\sum_{t=1}^{T} g_n^t(\theta)$.
5: **end for**

output θ_N (final estimate);

Proposition 6.1 (Non-Convex Analysis). *Assume that the surrogates* $g_n^{\hat{t}_n}$ *from Algorithm 5 are majorant and are in* $\mathcal{S}_L(f^{\hat{t}_n}, \theta_{n-1})$. *Then, the conclusions of Proposition 2.1 hold with probability one.*

Proposition 6.2 (Convex Analysis). *Assume that* f *is convex. Define* $f^\star \triangleq \min_{\theta\in\Theta} f(\theta)$ *and* $\delta \triangleq \frac{1}{T}$. *When the surrogates* g_n^t *in Algorithm 5 are majorant and in* $\mathcal{S}_{L,\rho}(f^t, \theta_{n-1})$ *with* $\rho \geq L$, *we have*

$$\mathbb{E}[f(\theta_n) - f^\star] \leq \frac{L\|\theta^\star - \theta_0\|_2^2}{2\delta n} \quad \textit{for all } n \geq 1.$$

Assume now that f *is* μ*-strongly convex. For all* $n \geq 1$,

$$\begin{cases} \mathbb{E}[\|\theta^\star - \theta_n\|_2^2] \leq \left((1-\delta) + \delta\frac{L}{\rho+\mu}\right)^n \|\theta^\star - \theta_0\|_2^2 \\ \mathbb{E}[f(\theta_n) - f^\star] \leq \left((1-\delta) + \delta\frac{L}{\rho+\mu}\right)^{n-1} \frac{L\|\theta^\star - \theta_0\|_2^2}{2} \end{cases}.$$

Interestingly, the proof and the convergence rates of Proposition 6.2 are similar to those of the block coordinate scheme. For both schemes, the current iterate θ_n can be shown to be the minimizer of an approximate surrogate function which splits into different parts. Each iteration randomly picks up one part, and updates it. Like SAG or SDCA, we obtain linear convergence for strongly convex functions f, even though the upper bounds obtained for SAG and SDCA are better than ours.

It is also worth noticing that for smooth unconstrained problems, MISO and SAG yield different, but related, update rules. Assume for instance that "Lipschitz gradient surrogates" are used. At iteration n of MISO, each function g_n^t is a surrogate of f^t near some κ_{n-1}^t. The update rule of MISO can be shown to be $\theta_n \leftarrow \frac{1}{T}\sum_{t=1}^{T}\kappa_{n-1}^t - \frac{1}{TL}\sum_{t=1}^{T}\nabla f^t(\kappa_{n-1}^t)$; in comparison, the update rule of SAG is $\theta_n \leftarrow \theta_{n-1} - \frac{1}{TL}\sum_{t=1}^{T}\nabla f^t(\kappa_{n-1}^t)$.

The next section complements the theoretical analysis of the scheme MISO by numerical experiments and practical implementation heuristics.

7. Experiments

In this section, we show that MISO is efficient for solving large-scale machine learning problems.

7.1. Experimental Setting

We consider ℓ_2- and ℓ_1- logistic regression without intercept, and denote by m the number of samples and by p the number of features. The corresponding optimization problem can be written

$$\min_{\theta\in\mathbb{R}^p} \frac{1}{m}\sum_{t=1}^{m} \log(1 + e^{-y_t \mathbf{x}^{t\top}\theta}) + \lambda\psi(\theta), \tag{3}$$

where the regularizer ψ is either the ℓ_1- or squared ℓ_2-norm. The y_t's are in $\{-1,+1\}$ and the $\mathbf{x}^t$'s are vectors in $\mathbb{R}^p$ with unit ℓ_2-norm. We use four classical datasets described in the following table:

name	m	p	storage	size (GB)
alpha	250 000	500	dense	1
rcv1	781 265	47 152	sparse	0.95
covtype	581 012	54	dense	0.11
ocr	2 500 000	1 155	dense	23.1

Three datasets, alpha, rcv1 and ocr were obtained from the 2008 Pascal large scale learning challenge.[5] The dataset covtype is available from the LIBSVM website.[6] We have chosen to test several software packages including LIBLINEAR 1.93 (Fan et al., 2008), the ASGD and SGD implementations of L. Bottou (version 2)[7], an implementation of SAG kindly provided to us by the authors of Le Roux et al. (2012), the FISTA method of Beck & Teboulle (2009) implemented in the SPAMS toolbox[8], and SHOTGUN (Bradley et al., 2011). All these softwares are coded in C++ and were compiled using gcc. Experiments were run on a single core of a 2.00GHz Intel Xeon CPU E5-2650 using 64GB of RAM, and all computations were done in double precision. All the timings reported do not include data loading into memory. Note that we could not run the softwares SPAMS, LIBLINEAR and SHOTGUN on the dataset ocr because of index overflow issues.

7.2. On Implementing MISO

The objective function (3) splits into m components $f^t : \theta \mapsto \log(1 + e^{-y_t \mathbf{x}^{t\top}\theta}) + \lambda\psi(\theta)$. It is thus natural to consider the incremental scheme of Section 6 together with the proximal gradient surrogates of Section 2.2. Concretely, we build at iteration n of MISO a surrogate $g_n^{\hat{t}_n}$ of $f^{\hat{t}_n}$ as follows: $g_n^{\hat{t}_n} : \theta \mapsto l^{\hat{t}_n}(\theta_{n-1}) + \nabla l^{\hat{t}_n}(\theta_{n-1})^\top(\theta - \theta_{n-1}) + \frac{L}{2}\|\theta - \theta_{n-1}\|_2^2 + \lambda\psi(\theta)$, where l^t is the logistic function $\theta \mapsto \log(1 + e^{-y_t \mathbf{x}^{t\top}\theta})$.

After removing the dependency over n to simplify the notation, all the surrogates can be rewritten as $g^t : \theta \mapsto a_t + \mathbf{z}^{t\top}\theta + \frac{L}{2}\|\theta\|_2^2 + \lambda\psi(\theta)$, where a_t is a constant and $\mathbf{z}^t$ is a vector in $\mathbb{R}^p$. Therefore, all surrogates can be "summarized" by the pair $(a_t, \mathbf{z}^t)$, quantities which we keep into memory during the optimization. Then, finding the estimate θ_n amounts to minimizing a function of the form $\theta \mapsto \bar{\mathbf{z}}_n^\top\theta + \frac{L}{2}\|\theta\|_2^2 + \lambda\psi(\theta)$, where $\bar{\mathbf{z}}_n$ is the average value of the quantities $\mathbf{z}^t$ at iteration n. It is then easy to see that obtaining $\bar{\mathbf{z}}_{n+1}$ from $\bar{\mathbf{z}}_n$ can be done in $O(p)$ operations with the following update: $\bar{\mathbf{z}}_{n+1} \leftarrow \bar{\mathbf{z}}_n + (\mathbf{z}_{\text{new}}^{\hat{t}_n} - \mathbf{z}_{\text{old}}^{\hat{t}_n})/m$.

One issue is that building the surrogates g^t requires choosing some constant L. An upper bound on the Lipschitz constants of the gradients ∇l^t could be used here. However, we have observed that significantly faster convergence could be achieved by using a smaller value, probably because a local Lipschitz constant may be better adapted than a global one. By studying the proof of Proposition 6.2, we notice indeed that our convergence rates can be obtained without majorant surrogates, when we simply have: $\mathbb{E}[f^t(\theta_n)] \leq \mathbb{E}[g_n^t(\theta_n)]$ for all t and n. This motivates the following heuristics:

- MISO1: start by performing one pass over $\eta = 5\%$ of the data to select a constant L' yielding the smallest decrease of the objective, and set $L = L'\eta$;
- MISO2: in addition to MISO1, check the inequalities $f^{\hat{t}_n}(\theta_{n-1}) \leq g_{n-1}^{\hat{t}_n}(\theta_{n-1})$ during the optimization. After each pass over the data, if the rate of satisfied inequalities drops below 50%, double the value of L.

Following these strategies, we have implemented the scheme MISO in C++. The resulting software package will be publicly released with an open source license.

7.3. ℓ_2-Regularized Logistic Regression

We compare LIBLINEAR, FISTA, SAG, ASGD, SGD, MISO1, MISO2 and MISO2 with $T = 1000$ blocks (grouping some observations into minibatches). LIBLINEAR was run using the option -s 0 -e 0.000001. The implementation of SAG includes a heuristic line search in the same spirit as MISO2, introduced by Le Roux et al. (2012). Every method was stopped after 50 passes over the data. We considered three regularization regimes, high ($\lambda = 10^{-3}$), medium ($\lambda = 10^{-5}$) and low ($\lambda = 10^{-7}$). We present in Figure 1 the values of the objective function during the optimization for the regime medium, both in terms of passes over the data and training time. The regimes low and high are provided as supplemental material only. Note that to reduce the memory load, we used a minibatch strategy for the dataset rcv1 with $T = 10\,000$ blocks.

Overall, there is no clear winner from this experiment, and the preference for an algorithm depends on the dataset, the required precision, or the regularization level. The best methods seem to be consistently MISO, ASGD and SAG and the slowest one FISTA. Note that this apparently mixed result is a significant achievement. We have indeed focused on state-of-the-art solvers, which already significantly outperform a large number of other baselines (see Bottou, 2010; Fan et al., 2008; Le Roux et al., 2012).

[5] http://largescale.ml.tu-berlin.de.
[6] http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
[7] http://leon.bottou.org/projects/sgd.
[8] http://spams-devel.gforge.inria.fr/.

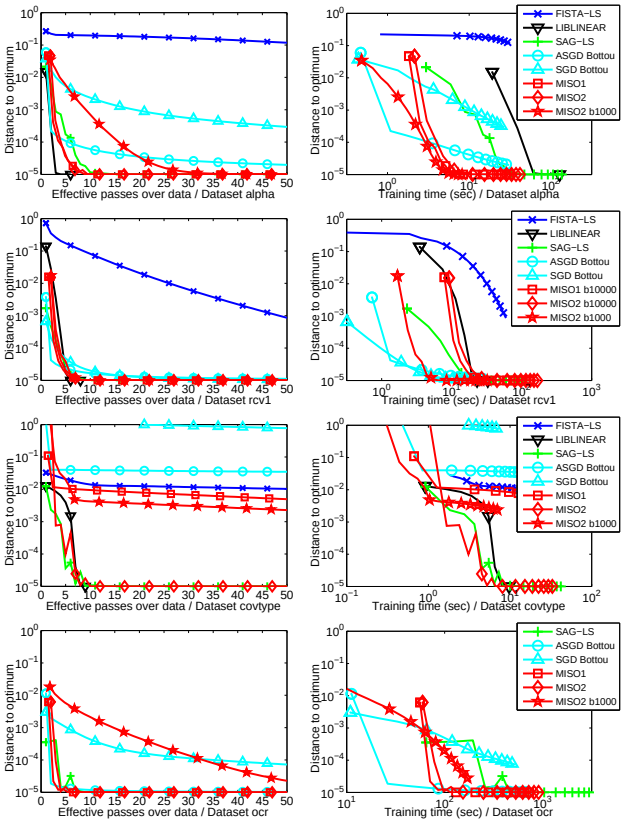

Figure 1. Results for ℓ_2-logistic regression with $\lambda = 10^{-5}$.

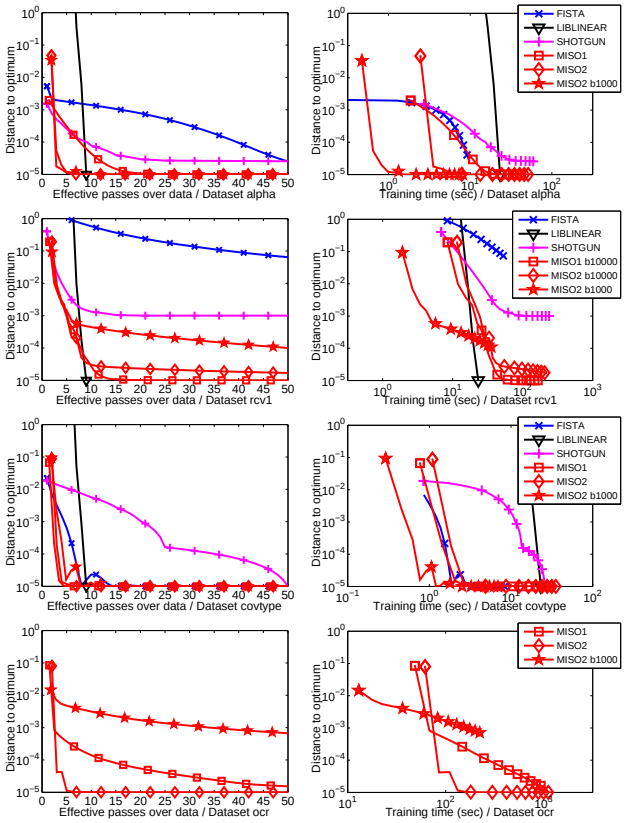

Figure 2. Benchmarks for ℓ_1-logistic regression. λ was chosen to obtain a solution with 10% nonzero coefficients.

7.4. ℓ_1-Regularized Logistic Regression

Since SAG, SGD and ASGD cannot deal with ℓ_1-regularization, we compare here LIBLINEAR, FISTA, SHOTGUN and MISO. We use for LIBLINEAR the option `-s 6 -e 0.000001`. We proceed as in Section 7.3, considering three regularization regimes yielding different sparsity levels. We report the results for one of them in Figure 2 and provide the rest as supplemental material. In this experiment, our method outperforms other competitors, except LIBLINEAR on the dataset `rcv1` when a high precision is required (and the regularization is low). We also remark that a low precision solution is often achieved quickly using the minibatch scheme (MISO2 b1000), but this strategy is outperformed by MISO1 and MISO2 for high precisions.

8. Conclusion

In this paper, we have introduced a flexible optimization framework based on the computation of "surrogate functions". We have revisited numerous schemes and discovered new ones. For each of them, we have studied convergence guarantees for non-convex problems and convergence rates for convex ones. Our methodology led us in particular to the design of an incremental algorithm, which has theoretical properties and empirical performance matching state-of-the-art solvers for large-scale machine learning problems.

In the future, we are planning to study fully stochastic or memoryless variants of our framework. As in the incremental setting, it consists of drawing a single training point at each iteration, but the algorithm does not keep track of all past information. This is essentially a strategy followed by Neal & Hinton (1998) and Mairal et al. (2010) in the respective contexts of EM and sparse coding algorithms. This would be particularly important for processing sparse datasets with a large number of features, where storing (dense) information about the past surrogates is cumbersome.

Acknowledgments

JM would like to thank Zaid Harchaoui, Francis Bach, Simon Lacoste-Julien, Mark Schmidt, Martin Jaggi, and Bin Yu for fruitful discussions. This work was supported by Quaero, (funded by OSEO, the French state agency for innovation), by the Gargantua project (program Mastodons - CNRS), and by the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370.

References

Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

Bertsekas, D.P. *Nonlinear programming*. Athena Scientific Belmont, 1999. 2nd edition.

Böhning, D. and Lindsay, B. G. Monotonicity of quadratic-approximation algorithms. *Ann. I. Stat. Math.*, 40(4): 641–663, 1988.

Borwein, J.M. and Lewis, A.S. *Convex analysis and nonlinear optimization: theory and examples*. Springer, 2006.

Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proc. COMPSTAT*, 2010.

Bradley, J.K., Kyrola, A., Bickson, D., and Guestrin, C. Parallel coordinate descent for l1-regularized loss minimization. In *Proc. ICML*, 2011.

Candès, E.J., Wakin, M., and Boyd, S.P. Enhancing sparsity by reweighted ℓ_1 minimization. *J. Fourier Anal. Appl.*, 14(5):877–905, 2008.

Collins, M., Schapire, R.E., and Singer, Y. Logistic regression, AdaBoost and Bregman distances. *Mach. Learn.*, 48(1):253–285, 2002.

Daubechies, I., Defrise, M., and De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pur. Appl. Math.*, 57 (11):1413–1457, 2004.

Della Pietra, S., Della Pietra, V., and Lafferty, J. Duality and auxiliary functions for Bregman distances. Technical report, CMU-CS-01-109, 2001.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, 2008.

Gasso, G., Rakotomamonjy, A., and Canu, S. Recovering sparse signals with non-convex penalties and DC programming. *IEEE T. Signal Process.*, 57(12):4686–4698, 2009.

Harchaoui, Z., Juditsky, A., and Nemirovski, A. Conditional gradient algorithms for norm-regularized smooth convex optimization. *preprint arXiv:1302.2325v4*, 2013.

Hazan, E. and Kale, S. Projection-free online learning. In *Proc. ICML*, 2012.

Horst, R. and Thoai, N.V. DC programming: overview. *J. Optim. Theory App.*, 103(1):1–43, 1999.

Juditsky, A. and Nemirovski, A. First order methods for nonsmooth convex large-scale optimization, I: General purpose methods. In *Optimization for Machine Learning*. MIT Press, 2011.

Khan, E., Marlin, B., Bouchard, G., and Murphy, K. Variational bounds for mixed-data factor analysis. In *Adv. NIPS*, 2010.

Lacoste-Julien, S., Jaggi, M., Schmidt, M., and Pletscher, P. Block-coordinate Frank-Wolfe optimization for structural SVMs. In *Proc. ICML*, 2013.

Lange, K., Hunter, D.R., and Yang, I. Optimization transfer using surrogate objective functions. *J. Comput. Graph. Stat.*, 9(1):1–20, 2000.

Le Roux, N., Schmidt, M., and Bach, F. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Adv. NIPS*, 2012.

Lee, D.D. and Seung, H.S. Algorithms for non-negative matrix factorization. In *Adv. NIPS*, 2001.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11:19–60, 2010.

Neal, R.M. and Hinton, G.E. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Learning in graphical models*, 89:355–368, 1998.

Nesterov, Y. *Introductory lectures on convex optimization*. Kluwer Academic Publishers, 2004.

Nesterov, Y. Gradient methods for minimizing composite objective functions. Technical report, CORE Discussion Paper, 2007.

Nesterov, Y. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM J. Optimiz.*, 22 (2):341–362, 2012.

Nesterov, Y. and Polyak, B.T. Cubic regularization of Newton method and its global performance. *Math. Program.*, 108(1):177–205, 2006.

Richtárik, P. and Takáč, M. Iteration complexity of randomized block coordinate descent methods for minimizing a composite function. *Math. Program.*, 2012.

Seeger, M.W. and Wipf, D.P. Variational Bayesian inference techniques. *IEEE Signal Proc. Mag.*, 27(6):81–91, 2010.

Shalev-Schwartz, S. and Zhang, T. Proximal stochastic dual coordinate ascent. *preprint arXiv 1211.2717v1*, 2012.

Shalev-Shwartz, S. and Tewari, A. Stochastic methods for ℓ_1 regularized loss minimization. In *Proc. ICML*, 2009.

Tseng, P. and Yun, S. A coordinate gradient descent method for nonsmooth separable minimization. *Math. Program.*, 117:387–423, 2009.

Wainwright, M.J. and Jordan, M.I. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, 2008.

Wright, S., Nowak, R., and Figueiredo, M. Sparse reconstruction by separable approximation. *IEEE T. Signal Process.*, 57(7):2479–2493, 2009.

Zhang, T. Sequential greedy approximation for certain convex optimization problems. *IEEE T. Inform. Theory*, 49 (3):682–691, 2003.

Zhang, X., Yu, Y., and Schuurmans, D. Accelerated training for matrix-norm regularization: a boosting approach. In *Adv. NIPS*, 2012.