Stochastic Dual Coordinate Ascent with Alternating Direction Method of Multipliers

Taiji Suzuki S-TAIJI@IS.TITECH.AC.JP
Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo 152-8552, JAPAN

Abstract

We propose a new stochastic dual coordinate ascent technique that can be applied to a wide range of regularized learning problems. Our method is based on alternating direction method of multipliers (ADMM) to deal with complex regularization functions such as structured regularizations. Although the original ADMM is a batch method, the proposed method offers a stochastic update rule where each iteration requires only one or few sample observations. Moreover, our method can naturally afford mini-batch update and it gives speed up of convergence. We show that, under mild assumptions, our method converges exponentially. The numerical experiments show that our method actually performs efficiently.

1. Introduction

This paper proposes a new stochastic optimization method that shows exponential convergence and can be applied to wide range of regularization functions using the techniques of stochastic dual coordinate ascent with alternating direction method of multipliers. Recently, it is getting more and more important to develop an efficient optimization method which can handle large amount of samples. One of the most successful approaches is a stochastic optimization approach. Indeed, a lot of stochastic methods have been proposed to deal with large amount of samples. Among them, the (online) stochastic gradient method is the most basic and successful one. This can be naturally applied to the regularized learning frame-work. Such a method is called several different names including online proximal gradient descent, forward-backward splitting and online mirror descent (Duchi and Singer, 2009). Basically, these methods are intended to process sequentially coming data. They update the parameter using one new observation and discard the observed sample. Therefore, they don't need large memory space to store the whole observed data. The convergence rate of those methods is $O(1/\sqrt{T})$ for general settings and $O(1/T)$ for strongly convex losses, which are minimax optimal (Nemirovskii and Yudin, 1983).

On the other hand, recently it was shown that, if it is allowed to reuse the observed data several times, it is possible to develop a stochastic method with exponential convergence rate for a strongly convex objective (Le Roux et al., 2013; Shalev-Shwartz and Zhang, 2013c;a). These methods are still stochastic in a sense that one sample or small mini-batch is randomly picked up to be used for each update. The main difference from the stochastic gradient method is that these methods are intended to process data with a fixed number of training samples. stochastic average gradient (SAG) method (Le Roux et al., 2013) utilizes an *averaged* gradient to show an exponential convergence. stochastic dual coordinate ascent (SDCA) method solves the dual problem using a stochastic coordinate ascent technique (Shalev-Shwartz and Zhang, 2013c;a). These methods have favorable properties of both online-stochastic approach and batch approach. That is, they show fast decrease of the objective function in the early stage of the optimization as online-stochastic approaches, and shows exponential convergence after the "burn in" time as batch approaches. However, these methods have some drawbacks. SAG needs to maintain all gradients computed on each training sample in memory which amount to dimension times sample size. SDCA method can be applied only to a simple regularization function for which the dual function is easily computed, thus it is hard to apply the method to a complex regularization function such as structured regularization.

In this paper, we propose stochastic dual coordinate ascent method for alternating direction method of multipliers (SDCA-ADMM). Our method is similar to SDCA, but inherits a favorable property of ADMM. By combining SDCA and ADMM, our method can be applied to a wide range of regularized learning problems. ADMM is an effective optimization method to solve a composite optimization problem described as $\min_x f(x) + g(y)$ s.t. $Ax +$

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

$By = 0$ (Gabay and Mercier, 1976; Boyd et al., 2010; Qin and Goldfarb, 2012). This formulation is quite flexible and fit wide range of applications such as structured regularization, dictionary learning, convex tensor decomposition and so on (Qin and Goldfarb, 2012; Jacob et al., 2009; Tomioka et al., 2011; Rakotomamonjy, 2013). However, ADMM is a batch optimization method. Our approach transforms ADMM to a stochastic one by utilizing stochastic coordinate ascent technique. Our method, SDCA-ADMM, does not require large amount of memory because it observes only one or few samples for each iteration. SDCA-ADMM can be naturally adapted to a sub-batch situation where a block of few samples is utilized for each iteration. Moreover, it is shown that our method shows exponential convergence for risk functions with some strong convexity and smoothness property. The convergence rate is affected by the size of sub-batch. If the samples are not strongly correlated, sub-batch gives a better convergence rate than one-sample update.

2. Structured Regularization and its Dual Formulation

In this section, we give the problem formulation of structured regularization and its dual formulation. The standard regularized risk minimization is described as follows:

$$\min_{w\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n} f_i(z_i^\top w) + \tilde{\psi}(w), \tag{1}$$

where $z_1, z_2, \dots, z_n$ are vectors in $\mathbb{R}^p$, w is the weight vector that we want to learn, f_i is a loss function for the i-th sample, and $\tilde{\psi}$ is the regularization function which is used to avoid over-fitting. For example, the loss function f_i can be taken as a classification surrogate loss $f_i(z_i^\top w) = \ell(y_i, z_i^\top w)$ where y_i is the training label of the i-th sample. With regard to $\tilde{\psi}$, we are interested in a sparsity inducing regularization, e.g., ℓ_1-regularization, group lasso regularization, trace-norm regularization, and so on. Our motivation in this paper is to deal with a "complex" regularization $\tilde{\psi}$ where it is not easy to directly minimize the regularization function (more precisely the proximal operation determined by $\tilde{\psi}$ is not easily computed, see Eq. (5)). This kind of regularization appears in, for example, structured sparsity such as overlapped group lasso and graph regularization (Jacob et al., 2009; Signoretto et al., 2010). In many cases, such a "complex" regularization function can be decomposed into a "simple" regularization ψ and a linear transformation B, that is, $\tilde{\psi}(w) = \psi(B^\top w)$ where $B \in \mathbb{R}^{p\times d}$. Using this formulation, the optimization problem (Eq. (1)) is equivalent to

$$\min_{w\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n} f_i(z_i^\top w) + \psi(B^\top w). \tag{2}$$

The purpose of this paper is to give an efficient stochastic optimization method to solve this problem (2). For this purpose, we employ the *dual formulation*. Using the *Fenchel's duality theorem*, we have the following dual formulation.

Lemma 1.

$$\min_{w\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n} f_i(z_i^\top w) + \psi(B^\top w)$$
$$= -\min_{\substack{x\in\mathbb{R}^n\\ y\in\mathbb{R}^d}} \left\{ \frac{1}{n}\sum_{i=1}^{n} f_i^*(x_i) + \psi^*\left(\frac{y}{n}\right) \mid Zx + By = 0 \right\}, \tag{3}$$

where f_i^* and ψ^* are the convex conjugates of f_i and ψ respectively (Rockafellar, 1970)[1], and $Z = [z_1, z_2, \dots, z_n] \in \mathbb{R}^{p\times n}$. Moreover w^*, x^* and y^* are optimal solutions of both sides if and only if

$$z_i^\top w^* \in \partial f_i^*(x_i^*),\ \frac{1}{n}y^* \in \partial\psi(u)|_{u=B^\top w^*},$$
$$Zx^* + By^* = 0.$$

Proof. By Fenchel's duality theorem (Corollary 31.2.1 of Rockafellar (1970)), we have that

$$\min_{w\in\mathbb{R}^p} \tfrac{1}{n}\textstyle\sum_{i=1}^{n} f_i(z_i^\top w) + \tilde{\psi}(w)$$
$$= -\min_{x\in\mathbb{R}^n} \left\{ \tfrac{1}{n}\textstyle\sum_{i=1}^{n} f_i^*(x_i) + \tilde{\psi}^*(-Zx/n) \right\}. \tag{4}$$

Moreover x^*, w^* are optimal in each side if and only if $z_i^\top w^* \in \partial f_i^*(x_i^*)$ and $-Zx^*/n \in \partial\tilde{\psi}(w^*) = B\partial\psi(u^*)|_{u=B^\top w^*}$ (Corollary 31.3 of Rockafellar (1970)). Now, Theorem 16.3 of Rockafellar (1970) gives that

$$\tilde{\psi}^*(u) = (\psi \circ B^\top)^*(u) = \inf\{\psi^*(y) \mid By = u\}.$$

Thus $\tilde{\psi}^*(-Zx/n) = \inf\{\psi^*(y/n) \mid By = -Zx\}$, and substituting this into the RHS of Eq. (4) we obtain Eq. (3). Now, y^* satisfying $Zx^* + By^* = 0$ is the optimal solution if and only if $\psi^*(y^*/n) = \tilde{\psi}^*(-Zx^*/n)$ for the optimal x^*. Thus, if (w^*, x^*, y^*) is optimal, then we have $-Zx^*/n \in \partial\tilde{\psi}(w^*)$ and thus $\psi^*(y^*/n) = \tilde{\psi}^*(-Zx^*/n) = \langle w^*, -Zx^*/n\rangle - \tilde{\psi}(w^*) = \langle B^\top w^*, y^*/n\rangle - \psi(B^\top w^*)$ which implies $y^*/n \in \partial\psi(u)|_{u=B^\top w^*}$. Contrary, if $y^*/n \in \partial\psi(u)|_{u=B^\top w^*}$, then it is obvious that $-Zx^*/n \in \partial\tilde{\psi}(w^*)$ because $Zx^* + By^* = 0$. Therefore, we obtain the optimality conditions. □

The dual problem is a composite objective function optimization with a linear constraint $Zx + By = 0$. In the next section, we give an efficient stochastic method to solve this dual problem. A nice property of the dual formulation is that, in many machine learning applications, the dual loss

[1] The convex conjugate function f^* of f is defined by $f^*(y) := \sup_x\{x^\top y - f(x)\}$.

function f_i^* becomes strongly convex. For example, for the logistic loss $f_i(x) = \log(1+\exp(-y_i x))$, the dual function is $f_i^*(-u) = y_i u \log(y_i u) + (1 - y_i u)\log(1 - y_i u)$ $(y_i u \in [0,1])$ and its modulus of strong convexity is much better than the primal one. More importantly, each sample (z_i, y_i) directly affects only each coordinate x_i of dual variable. In other words, if x_i is fixed the i-th sample (z_i, y_i) has no influence to the objective value. This enables us to utilize the stochastic coordinate ascent technique in the dual problem because update of single coordinate x_i requires only the information of the i-th sample (z_i, y_i).

Finally, we give the precise notion of the "complex" and "simple" regularizations. This notion is defined by the computational complexity of *proximal operation* corresponding to the regularization function (Rockafellar, 1970). The proximal operation corresponding to a convex function ψ is defined by

$$\mathrm{prox}(q|\psi) := \arg\min_u \left\{ \frac{1}{2}\|q-u\|^2 + \psi(u) \right\}. \tag{5}$$

For example, the proximal operation corresponding to ℓ_1-regularization $\psi(w) = \|w\|_{\ell_1}$ is easily computed as $\mathrm{prox}(q|\psi) = (\mathrm{sign}(w_i)\max\{|w_i|-1, 0\})_i$ which is the so-called soft-thresholding operation. More generally, the proximal operation for group lasso regularization with non-overlapped groups can also be analytically computed. On the other hand, for overlapped group regularization, the proximal operation is no longer analytically obtained. However, by choosing B appropriately, we can split the overlap and obtain ψ for which the proximal operation is easily computed (see Section 6 for concrete examples).

3. Proposed Method: Stochastic Dual Coordinate Ascent with ADMM

In this section, we present our proposal, stochastic dual coordinate ascent type ADMM. For a positive semidefinite matrix S, we denote by $\|x\|_S := \sqrt{x^\top S x}$. Z_i denotes the i-th column of Z, which is z_i, and $Z_{\setminus i}$ is a matrix obtained by subtracting i-th column from Z. Similarly, for a vector x, $x_{\setminus i}$ is a vector obtained by subtracting i-th component from x.

3.1. One Sample Update of SDCA for ADMM

The basic update rule of our proposed method in the t-th step is given as follows: Each update step, choose $i \in \{1, \dots, n\}$ uniformly at random, and update as

$$y^{(t)} \leftarrow \arg\min_y \Big\{ n\psi^*(y/n) - \langle w^{(t-1)}, Zx^{(t-1)} + By \rangle + \frac{\rho}{2}\|Zx^{(t-1)} + By\|^2 + \frac{1}{2}\|y - y^{(t-1)}\|_Q^2 \Big\}, \tag{6a}$$

$$x_i^{(t)} \leftarrow \arg\min_{x_i} \Big\{ f_i^*(x_i) - \langle w^{(t-1)}, Z_i x_i + By^{(t)} \rangle + \frac{\rho}{2}\|Z_i x_i + Z_{\setminus i} x_{\setminus i}^{(t-1)} + By^{(t)}\|^2 + \frac{1}{2}\|x_i - x_i^{(t-1)}\|_{G_{ii}}^2 \Big\}, \tag{6b}$$

$$w^{(t)} \leftarrow w^{(t-1)} - \gamma\rho\{ n(Zx^{(t)} + By^{(t)}) - (n-1)(Zx^{(t-1)} + By^{(t-1)}) \}, \tag{6c}$$

where $w^{(t)} \in \mathbb{R}^p$ is the primal variable at the t-th step, Q and G are arbitrary positive semidefinite matrices, and $\gamma, \rho > 0$ are parameters we give beforehand.

The optimization procedure looks a bit complicated, To simplify the procedure, we set Q as

$$Q = \rho(\eta_B \mathrm{I}_d - B^\top B) \tag{7}$$

where η_B are chosen so that $\eta_B \mathrm{I}_d \succ B^\top B$. Then, by carrying out simple calculations and denoting $\eta_{Z,i} = G_{ii}/\rho + \|z_i\|^2$, the update rule of $x^{(t)}$ and $y^{(t)}$ is rewritten as

$$y^{(t)} \leftarrow \mathrm{prox}\Big(y^{(t-1)} + \frac{B^\top}{\rho\eta_B}\{ w^{(t-1)} - \rho(Zx^{(t-1)} + By^{(t-1)})\} \Big| \frac{n\psi^*(\cdot/n)}{\rho\eta_B} \Big), \tag{8a}$$

$$x_i^{(t)} \leftarrow \mathrm{prox}\Big(x_i^{(t-1)} + \frac{Z_i^\top}{\rho\eta_{Z,i}}\{ w^{(t-1)} - \rho(Zx^{(t-1)} + By^{(t)})\} \Big| \frac{f_i^*}{\rho\eta_{Z,i}} \Big). \tag{8b}$$

Note that the update (8b) of $x^{(t)}$ is just a one dimensional optimization, thus it is quite easily computed. Moreover, for some loss functions such as the smoothed hinge loss used in Section 6, we have an analytic form of the update.

The update rule (8a) of $y^{(t)}$ can be rewritten by the proximal operation corresponding to the primal function ψ while the rule (8a) is given by that corresponding to the dual function ψ^*. Indeed, there is a clear relation between primal and dual (Theorem 31.5 of Rockafellar (1970)):

$$\mathrm{prox}(q|\psi) + \mathrm{prox}(q|\psi^*) = q.$$

Using this, for $q^{(t)} = y^{(t-1)} + \frac{B^\top}{\rho\eta_B}\{w^{(t-1)} - \rho(Zx^{(t-1)} + By^{(t-1)})\}$, we have that

$$y^{(t)} \leftarrow q^{(t)} - \mathrm{prox}(q^{(t)} | n\psi(\rho\eta_B \ \cdot\ )/(\rho\eta_B)), \tag{9}$$

because $(cf(\cdot))^*(y) = cf^*(y/c)$ for a convex function f and $c > 0$. This is efficiently computed because we assumed the proximal operation corresponding to ψ can be efficiently computed.

During the update, we need $Zx^{(t-1)}$ which seems to require $O(n)$ computation at the first glance. However, it can be incrementally updated as $Zx^{(t)} = Zx^{(t-1)} + Z_i(x_i^{(t)} -$

$x_i^{(t-1)}$). Thus we don't need to load all the samples to compute $Zx^{(t-1)}$ at each iteration.

In the above, the update rule of our algorithm is based on one sample observation. Next, we give a mini-batch extension of the algorithm where more than one samples could be used for each iteration.

3.2. Mini-Batch Extension

Here, we generalize our method to mini-batch situation where, at each iteration, we observe a small number of samples $\{(x_{i_1}, y_{i_1}), \dots, (x_{i_k}, y_{i_k})\}$ instead of one sample observation. At each iteration, we randomly choose an index set $I \subseteq \{1, \dots, n\}$ so that each index i is included in I with probability $1/K$; $P(i \in I) = 1/K$ for all $i = 1, \dots, n$. To do so, we suggest the following procedure. We split the index set $\{1, \dots, n\}$ into K groups $(I_1, I_2, \dots, I_K)$ beforehand, and then pick up uniformly $k \in \{1, \dots, K\}$ and set $I = I_k$ for each iteration. Each sub-batch I_k can have different cardinality from others, but the probability $P(i \in I)$ should be uniform for all $i = 1, \dots, n$. The update rule using sub-batch is given as follows: Update $y^{(t)}$ as before (6a), and update $x^{(t)}$ and $w^{(t)}$ by

$$x_I^{(t)} \leftarrow \arg\min_{x_I} \Big\{ \sum_{i \in I} f_i^*(x_i) - \langle w^{(t-1)}, Z_I x_I + By^{(t)} \rangle + \frac{\rho}{2}\|Z_I x_I + Z_{\setminus I} x_{\setminus I}^{(t-1)} + By^{(t)}\|^2 + \frac{1}{2}\|x_I - x_I^{(t-1)}\|_{G_{I,I}}^2 \Big\}, \tag{10a}$$

$$w^{(t)} \leftarrow w^{(t-1)} - \gamma\rho\{n(Zx^{(t)} + By^{(t)}) - (n - n/K)(Zx^{(t-1)} + By^{(t-1)})\}. \tag{10b}$$

Using Q given in Eq. (7), the update rule of $y^{(t)}$ can be replaced by Eq. (9) as in one-sample update situation. The update rule of $x^{(t)}$ can also be simplified by choosing G appropriately. Because sub-batches have no overlap between each other, we can construct a positive semi-definite matrix G such that the block-diagonal element $G_{I,I}$ has the form

$$G_{I,I} = \rho(\eta_{Z,I} \mathrm{I}_{|I|} - Z_I^\top Z_I) \tag{11}$$

where $\eta_{Z,I}$ is a positive real satisfying $\eta_{Z,I} \geq \|Z_I^\top Z_I\|$. The reason why we split the index sets into K sets is to construct this kind of G which "diagonalizes" the quadratic function in (10a). The choice of I and G could be replaced with another one for which we could compute the update efficiently, as long as $P(i \in I)$ is uniform for all $i = 1, \dots, n$. Using G given in (11), the update rule (10a) of $x^{(t)}$ is rewritten as

$$x_I^{(t)} \leftarrow \mathrm{prox}\Big(x_I^{(t-1)} + \frac{Z_I^\top}{\rho\eta_{Z,I}}\{w^{(t-1)} - \rho(Zx^{(t-1)} + By^{(t)})\} \,\Big|\, \frac{\sum_{i \in I} f_i^*}{\rho\eta_{Z,I}}\Big), \tag{12}$$

where x_I is a vector consisting of components with indexes $i \in I$, $x_I = (x_i)_{i \in I}$, and Z_I is a sub-matrix of Z consisting of columns with indexes $i \in I$, $Z_I = [Z_{i_1}, \dots, Z_{i_{|I|}}]$. Note that, since $\sum_{i \in I} f_i^*(x_i)$ is sum of single variable convex functions $f_i^*(x_i)$, the proximal operation in Eq. (12) can be split into the proximal operation with respect to each single variable x_i. This is advantageous for not only the simpleness of the computation but also parallel computation. That is, for $p_I = x_I^{(t-1)} + \frac{Z_I^\top}{\rho\eta_{Z,I}}\{w^{(t-1)} - \rho(Zx^{(t-1)} + By^{(t)})\}$, the update rule (12) is reduced to $x_i^{(t)} \leftarrow \mathrm{prox}(p_i | \frac{f_i^*}{\rho\eta_{Z,I}})$ for each $i \in I$, which is easily parallelizable. In summary, our proposed algorithm is given in Algorithm 1.

Algorithm 1 SDCA-ADMM

```
Input: $\rho, \eta > 0$
Initialize $x_0 = \mathbf{0}$, $y_0 = \mathbf{0}$, $w_0 = \mathbf{0}$ and $\{I_1, \dots, I_K\}$.
for $t = 1$ to $T$ do
    Choose $k \in \{1, \dots, K\}$ uniformly at random, set $I = I_k$, and observe the training samples $\{(x_i, y_i)\}_{i \in I}$.
    Set $q^{(t)} = y^{(t-1)} + \frac{B^\top}{\rho\eta_B}\{w^{(t-1)} - \rho(Zx^{(t-1)} + By^{(t-1)})\}$.
    Update $y^{(t)} \leftarrow q^{(t)} - \mathrm{prox}(q^{(t)} | n\psi(\rho\eta_B\ \cdot\ )/(\rho\eta_B))$
    Update $x_I^{(t)} \leftarrow \mathrm{prox}\Big(x_I^{(t-1)} + \frac{Z_I^\top}{\rho\eta_{Z,I}}\{w^{(t-1)} - \rho(Zx^{(t-1)} + By^{(t)})\} \,\Big|\, \frac{\sum_{i \in I} f_i^*}{\rho\eta_{Z,I}}\Big)$.
    Update $w^{(t)} \leftarrow w^{(t-1)} - \gamma\rho\{n(Zx^{(t)} + By^{(t)}) - (n - n/K)(Zx^{(t-1)} + By^{(t-1)})\}$.
end for
Output: $w^{(T)}$.
```

Finally, we would like to highlight the connection between our method and the original batch ADMM (Hestenes, 1969; Powell, 1969; Rockafellar, 1976). The batch ADMM utilizes the following update rule

$$y^{(t)} \leftarrow \arg\min_y \Big\{ n\psi^*\left(\frac{y}{n}\right) - \langle w^{(t-1)}, Zx^{(t-1)} + By \rangle + \frac{\rho}{2}\|Zx^{(t-1)} + By\|^2 \Big\}, \tag{13a}$$

$$x^{(t)} \leftarrow \arg\min_x \{ \textstyle\sum_{i=1}^n f_i^*(x_i) - \langle w^{(t-1)}, Zx + By^{(t)} \rangle + \frac{\rho}{2}\|Zx + By^{(t)}\|^2 \}, \tag{13b}$$

$$w^{(t)} \leftarrow w^{(t-1)} - \gamma\rho(Zx^{(t)} + By^{(t)}). \tag{13c}$$

One can see that the update rule of our algorithm is reduced to that of the batch ADMM (13) if we set $K = 1$ except the term related to G and Q (the terms $\frac{1}{2}\|\cdot\|_Q^2$ and $\frac{1}{2}\|\cdot\|_{G_{I,I}}^2$). These terms related to G and Q are used also in batch situation to eliminate cross terms in $BB^\top$ and $ZZ^\top$. This technique is called linearization. The linearization technique makes the update rule simple and parallelizable, and in some situations makes it possible to obtain an analytic form of the update.

4. Linear Convergence of SDCA-ADMM

In this section, the convergence rate of our proposed algorithm is given. Indeed, the convergence rate is exponential (R-linear). To show the convergence rate, we assume some conditions. First, we assume that there exits an unique optimal solution w^* and $B^\top$ is injective (on the other hand, B is not necessarily injective). Moreover, we assume the uniqueness of the dual solution x^*, but don't assume the uniqueness of y^*. We denote by the set of dual optimum of y as $\mathcal{Y}^*$ and assume that $\mathcal{Y}^*$ is compact. Then, by Lemma 1, we have that

$$z_i^\top w^* \in \partial f_i^*(x_i^*), \ \ y^*/n \in \partial\psi(u)|_{u=B^\top w^*}. \qquad (14)$$

By the convex duality arguments, this implies that $x_i^* \in \partial f_i(u)|_{u=z_i^\top w^*}, \ \ B^\top w^* \in \partial\psi^*(u)|_{u=y^*/n}$.

Moreover, we suppose that each (dual) loss function f_i is locally v-strongly convex and ψ, h-smooth around the optimal solution and ψ^* is also locally strongly convex in a weak sense as follows.

Assumption 1. *There exits $v > 0$ such that, $\forall x_i \in \mathbb{R}$,*

$$f_i^*(x_i) - f_i^*(x_i^*) \geq \langle z_i^\top w^*, x_i - x_i^*\rangle + \frac{v\|x_i - x_i^*\|^2}{2}.$$

There exit $h > 0$ and $v_\psi > 0$ such that, for all y, u and $y^* \in \mathcal{Y}^*$, there exists $\widehat{y}^* \in \mathcal{Y}^*$ (depending on y) and we have

$$\begin{aligned}\psi^*(y/n) - \psi^*(\widehat{y}^*/n) &\geq \langle B^\top w^*, y/n - \widehat{y}^*/n\rangle \\ &\quad + \tfrac{v_\psi}{2}\|P_{\mathrm{Ker}(B)}(y/n - \widehat{y}^*/n)\|^2, \qquad (15)\\ \psi(u) - \psi(B^\top w^*) &\geq \langle y^*/n, u - B^\top w^*\rangle \\ &\quad + \tfrac{h}{2}\|u - B^\top w^*\|^2, \qquad (16)\end{aligned}$$

where $P_{\mathrm{Ker}(B)}$ is the projection matrix to the kernel of B.

Note that these conditions should be satisfied only around the optimal solutions (x^*, y^*) and w^*. It does not need to hold for every point, thus is much weaker than the ordinary strong convexity. Moreover, the inequalities need to be satisfied only for the solution sequence $(w^{(t)}, x^{(t)}, y^{(t)})$ of our algorithm. The strong convexity of the dual loss f_i^* implies that the primal loss f_i is smooth around the optimal. The condition (15) is satisfied, for example, by ℓ_1-regularization because the dual of ℓ_1-regularization is an indicator function with a compact support and, outside the optimal solution set $\mathcal{Y}^*$, the indicator function is lower bounded by a quadratic function. In addition, the quadratic term in the right hand side of this condition (15) is restricted on $\mathrm{Ker}(B)$. This makes it possible to include several types of regularization functions. Indeed, if $B = \mathrm{I}_p$, this condition is always satisfied. The assumption (16) is the strongest assumption. This is satisfied for elastic-net regularization. ℓ_1-regularization could satisfy this condition depending on the optimum w^* and the solution sequence. If one wants to make this condition always hold, just adding a small square term, then the condition is satisfied and we obtain an approximated solution which is sufficiently close to the true one within a precision.

Define the primal and dual objectives as

$$\begin{aligned}F_P(w) &:= \tfrac{1}{n}\textstyle\sum_{i=1}^n f_i(z_i^\top w) + \psi(B^\top w),\\ F_D(x, y) &:= \tfrac{1}{n}\textstyle\sum_{i=1}^n f_i^*(x_i) + \psi^*(\tfrac{y}{n}) - \langle w^*, Z\tfrac{x}{n} - B\tfrac{y}{n}\rangle.\end{aligned}$$

Note that, by Eq. (14), $F_P(w) - F_P(w^*)$ and $F_D(x, y) - F_D(x^*, y^*)$ are always non-negative. Define the block diagonal matrix H as $H_{I,I} = \rho Z_I^\top Z_I + G_{I,I}$ for all $I \in \{I_1, \dots, I_K\}$ and $H_{i,j} = 0$ for $(i, j) \notin I_k \times I_k$ $(\forall k)$. Let $\|y - \mathcal{Y}^*\|_Q := \min\{\|y - y^*\|_Q \mid y^* \in \mathcal{Y}^*\}$. We define $R_D(x, y, w)$ as

$$\begin{aligned}R_D(x, y, w) &:= F_D(x, y) - F_D(x^*, y^*) + \frac{\|w - w^*\|^2}{2n^2\gamma\rho}\\ &+ \tfrac{\rho(1-\gamma)}{2n}\|Zx + By\|^2 + \tfrac{1}{2n}\|x - x^*\|^2_{v\mathrm{I}_p + H} + \tfrac{\|y - \mathcal{Y}^*\|_Q^2}{2nK}.\end{aligned}$$

For a symmetric matrix S, we define $\sigma_{\max}(S)$ and $\sigma_{\min}(S)$ as the maximum and minimum singular value respectively.

Theorem 2. *Suppose that $\gamma = \frac{1}{4n}$, $\eta_{Z,I} > \{1 + 2\gamma n(1 - 1/K)\}\sigma_{\max}(Z_I^\top Z_I)$ for all $I \in \{I_1, \dots, I_K\}$ and $B^\top$ is injective. Then, under Assumption 1, the dual objective function converges R-linearly: We have that, for $C_1 = R_D(x^{(0)}, y^{(0)}, w^{(0)})$,*

$$\mathrm{E}[R_D(x^{(T)}, y^{(T)}, w^{(T)})] \leq \left(1 - \frac{\mu}{K}\right)^T C_1,$$

where $\mu = \min\left\{\frac{v}{4(v + \sigma_{\max}(H))}, \frac{h\rho\sigma_{\min}(B^\top B)}{2\max\{1/n, 4h\rho, 4h\sigma_{\max(Q)}\}}, \frac{Kv_\psi/n}{4\sigma_{\max}(Q)}, \frac{Kv\sigma_{\min}(BB^\top)}{4\sigma_{\max(Q)}(\rho\sigma_{\max}(Z^\top Z) + 4v)}\right\}$. In particular,

$$\mathrm{E}[\|w^{(T)} - w^*\|^2] \leq \frac{n\rho}{2}\left(1 - \frac{\mu}{K}\right)^T C_1.$$

If we further assume $\psi(B^\top w) \leq \psi(B^\top w^*) + \langle y^*/n, B^\top(w - w^*)\rangle + l_1\|w - w^*\| + l_2\|w - w^*\|^2$ $(\forall w)$, then this implies that

$$\begin{aligned}&\mathrm{E}[F_P(w^{(T)}) - F_P(w^*)]\\ &\leq \left(\tfrac{\sigma_{\max}(Z^\top Z/n)}{2v} + l_2\right)\frac{n\rho}{2}\left(1 - \frac{\mu}{K}\right)^T C_1\\ &\quad + l_1\sqrt{\frac{n\rho}{2}\left(1 - \frac{\mu}{K}\right)^T C_1}.\end{aligned}$$

Since the proof is technical, it is deferred to the supplementary material. This theorem shows that the primal and dual objective values converge R-linearly. Moreover, the primal variable w also converges R-linearly to the optimal value. The number K of sub-batches controls the convergence rate. If all samples are nearly orthogonal to each other, $\sigma_{\max}(H)$ is bounded by a constant for all K, and thus convergence rate gets faster and faster as K decreases (the size of each sub-batch grows up). On the other hand, if samples are strongly correlated to each other, $\sigma_{\max}(H)$ grows linearly against $1/K$ and then the convergence rate is not improved by decreasing K. As for batch settings, the linear convergence of batch ADMM has been shown by Deng and Yin (2012). However, their proof can not be directly applied to our stochastic setting. Our proof requires a technique specialized to stochastic coordinate ascent technique. We would like to point out that the exponential convergence is not guaranteed if the choice of index set I at each update is cyclic. The index I is supposed to be chosen randomly. It is reported in Shalev-Shwartz and Zhang (2013a) that cyclic choice of I yields slower convergence. As described in the introduction, the mini-max optimal rate of stochastic optimization is at least $O(1/T)$. This corresponds to the convergence rate of the test loss in machine learning terminology. However, our analysis focuses on the training loss. Thus we can obtain the linear convergence.

The statement can be described in terms of the number of iterations required to achieve a precision ϵ, i.e. smallest T satisfying $\mathrm{E}[F_P(w^{(T)}) - F_P(w^*)] \leq \epsilon$:

$$T \leq C'K \max\left\{ \tfrac{4(v+\sigma_{\max}(H))}{v}, \tfrac{2\max\{1/(nh), 4\rho, 4\sigma_{\max(Q)}\}}{\rho\sigma_{\min}(B^\top B)}, \right.$$
$$\left. \tfrac{4\sigma_{\max}(Q)}{Kv_\psi/n}, \tfrac{4\sigma_{\max}(Q)(\rho\sigma_{\max}(Z^\top Z)+4v)}{Kv\sigma_{\min}(BB^\top)} \right\} \log\left(\tfrac{nC''}{\epsilon}\right),$$

where C' and C'' are an absolute constant. This says that dependency of ϵ is log-order. An interesting point is that the influence of h, the modulus of local strong convexity of ψ. Usually the regularization function is made weaker as the number of samples increases. In that situation, h decreases as n goes up. However, even if we set $h = 1/n$ (and $v_\psi \geq \frac{n}{K}$), we still have $T = O(K \log(n/\epsilon))$ instead of $O(nK \log(n/\epsilon))$. Thus, the convergence rate is hardly affected by the setting of h. This point is same as the ordinary SDCA algorithm (Shalev-Shwartz and Zhang, 2013a).

5. Related Works

In this section, we present some related works and discuss differences from our method.

The most related work is a recent study by Shalev-Shwartz and Zhang (2013c) in which stochastic dual coordinate ascent (SDCA) method for a regularized risk minimization is proposed. Their method also deals with the dual problem (3) with $B = \mathrm{I}_p$ in our setting, and apply a stochastic coordinate ascent technique. This method converges linearly. At each iteration, the method solves the following one-dimensional optimization problem, $\Delta x_i^{(t)} \leftarrow \arg\min_{\Delta x_i \in \mathbb{R}} \ f_i^*(\Delta x_i + x_i^{(t-1)}) + z_i^\top w^{(t-1)} \Delta x_i + \frac{1}{2n}\|z_i \Delta x_i\|^2$, and updates $x_i^{(t)} \leftarrow \Delta x_i^{(t)} + x_i^{(t-1)}$ and $w^{(t)} \leftarrow \partial\tilde{\psi}^*(-Zx^{(t)})$. The most important difference from our method is the computation of $\partial\psi^*$. In a “simple” regularization function, it is often easy to compute the (sub-)gradient of $\tilde{\psi}^*$. However, in a “complex” regularization such as structured regularization, the computation is not efficiently carried out. To overcome this difficulty, our method utilizes a linearly transformed one $\psi(B\cdot) = \tilde{\psi}(\cdot)$, and split the optimization with respect to f_i^* and ψ^* by applying ADMM technique. Thus, our method is applicable to much more general regularization functions. A mini-batch extension of SDCA is a recent hot topic (Takáč et al., 2013; Shalev-Shwartz and Zhang, 2013b). Our approach realizes the mini-batch extension using the linearlization technique in ADMM which is naturally derived in the frame-work of ADMM. Although the proof technique is quite different, the convergence analysis of normal mini-batch SDCA given by Shalev-Shwartz and Zhang (2013b) is parallel to our theorem.

The second method related to ours is stochastic average gradient (SAG) method (Le Roux et al., 2013). The method is a modification of stochastic gradient descent method, but utilizes an *averaged* gradient. A good point of their method is that we only need to deal with the primal problem. Thus the computation is easy, and we don’t need to look at the convex conjugate function. Moreover, their method also converges linearly. However, the linear convergence of SAG is guaranteed for smoothed loss and regularization functions. Thus non-smooth structured regularization is not included in the scope of the naive SAG procedure. It is conjectured by Schmidt et al. (2013) that SAG could be combined with proximal gradient frame-work and even ADMM. Our work gives a particular answer to this question in the setting of SDCA.

The third method is online version of ADMM. Recently some online variants of ADMM have been proposed by Wang and Banerjee (2012); Suzuki (2013); Ouyang et al. (2013). These methods are effective for complex regularizations as discussed in this paper. Thus they are applicable to wide range of situations. However, those methods are basically online methods, thus they discard the samples once observed. They are not adapted to a situation where the training samples are observed several times. Therefore, the convergence rate is $O(1/\sqrt{T})$ in general and $O(\log(T)/T)$ for a strongly convex loss (possibly $O(1/T)$ with some modification). On the other hand, our method converges linearly.

6. Numerical Experiments

In this section, we give numerical experiments on artificial and real data to demonstrate the effectiveness of our proposed algorithm[2]. We compare our SDCA-ADMM with the existing stochastic optimization methods such as regularized dual averaging (RDA) (Duchi and Singer, 2009; Xiao, 2009), online ADMM (OL-ADMM) (Wang and Banerjee, 2012), online proximal gradient descent ADMM (OPG-ADMM) (Ouyang et al., 2013; Suzuki, 2013) and RDA-ADMM (Suzuki, 2013). We also compared our method with batch ADMM (Batch-ADMM) in the artificial data sets. We used sub-batch with size 50 for all the methods including ours ($|I_k| = 50$, but $|I_K|$ could be less than 50). We employed the parameter settings $\gamma = 1/n$ and $\rho = 0.1$[3]. As for $\eta_{Z,I}$ and η_B, we used $\eta_{Z,I} = 1.1\sigma_{\max}(Z_I^\top Z_I)$ and $\eta_B = \sigma_{\max}(BB^\top) + 1$. All of the experiments are classification problems with structured sparsity. We employed the *smoothed hinge loss*:

$$f_i(u) = \begin{cases} 0, & (y_i u \geq 1), \\ \frac{1}{2} - y_i u, & (y_i u < 0), \\ \frac{1}{2}(1 - y_i u)^2, & (\text{otherwise}). \end{cases}$$

Then the proximal operation with respect to its dual function is analytically given as follows (see the supplementary material for the derivation):

$$\mathrm{prox}(u|f_i^*/C) = \begin{cases} \frac{Cu - y_i}{1+C} & (-1 \leq \frac{Cuy_i - 1}{1+C} \leq 0), \\ -y_i & (-1 > \frac{Cuy_i - 1}{1+C}), \\ 0 & (\text{otherwise}). \end{cases}$$

6.1. Artificial Data

Here we execute numerical experiments on artificial data sets. The problem is a classification problem with overlapped group regularization as performed in Suzuki (2013). We generated n input feature vectors $\{z_i\}_{i=1}^n$ with dimension $d = 32 \times 32 = 1024$ where each feature is generated from i.i.d. standard normal distribution. Then the true weight vector w_0 is generated as follows: First we generate a random matrix which has non-zero elements on its first column (distributed from i.i.d. standard normal) and zeros on other columns, and vectorize the matrix to obtain w_0. The training label y_i is given by $y_i = \mathrm{sign}(z_i^\top w_0 + \epsilon_i)$ where ϵ_i is distributed from normal distribution with mean 0 and standard deviation 0.1.

The group regularization is given as $\tilde{\psi}(x) = C(\sum_{i=1}^{32} \|X_{:,i}\| + \sum_{j=1}^{32} \|X_{j,:}\| + 0.01 \times \sum_{i,j} X_{i,j}^2/2)$ where X is the 32×32 matrix obtained by reshaping x. The quadratic term is added to make the regularization function strongly convex[4]. Since there exist overlaps between groups, the proximal operation can not be straightforwardly computed (Jacob et al., 2009). To deal with this regularization function in our frame-work, we let $B^\top x = [x; x](= [x^\top x^\top]^\top)$, that is $B = [\mathrm{I}_p \mathrm{I}_p]$, and $\psi([x; x']) = C(\sum_{i=1}^{32} \|X_{:,i}\| + \sum_{j=1}^{32} \|X'_{j,:}\|)$. Then we can see that $\tilde{\psi}(x) = \psi(B^\top x)$ and the proximal operation with respect to ψ is analytically obtained; indeed it is easily checked that $\mathrm{prox}([q; q']|\psi) = [\mathrm{ST}_{C'}(Q_{:,1}/(1 + 0.01C)); \dots; \mathrm{ST}_{C'}(Q_{:,32}/(1 + 0.01C)); \mathrm{ST}_{C'}(Q'_{1,:}/(1 + 0.01C)); \dots; \mathrm{ST}_{C'}(Q'_{32,:}/(1 + 0.01C))]$ where $\mathrm{ST}_C(q) = q \max(1 - C/\|q\|, 0)$ and $C' = C/(1 + 0.01C)$.

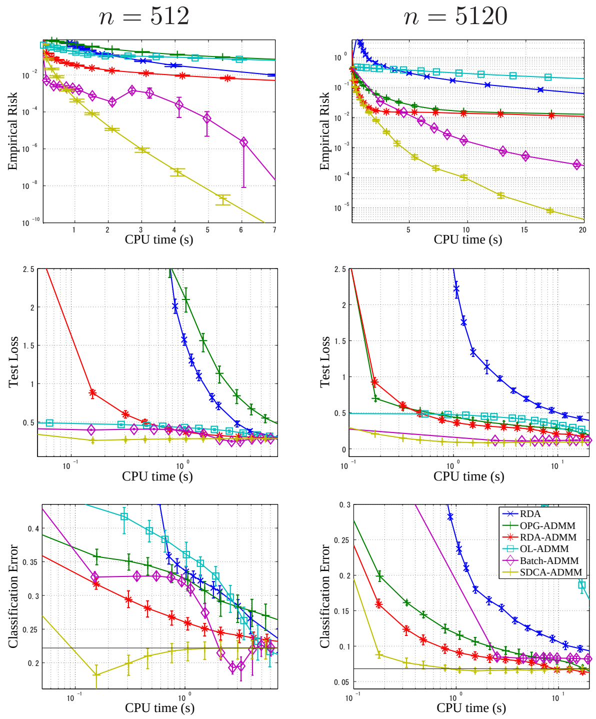

Figure 1. Excess empirical risk, exected loss on the test data and test classification error averaged over 10 independent iteration against CPU time in artificial data with $n = 512, 5120$. The error bar indicates the standard deviation.

The original RDA requires a direct computation of the proximal operation for the overlapped group penalty. To compute that, we employed the dual formulation proposed by Yuan et al. (2011).

We independently repeated the experiments 10 times and averaged the excess empirical risk $(F_P(w^{(t)}) - \min_w F_P(w))$, the expected loss on the test data $(\mathrm{E}_{(z,y)}[f(y, z^\top w^{(t)})])$ and the classification error $(\mathrm{E}_{(z,y)}[1\{y \neq \mathrm{sign}(z^\top w^{(t)})\}])$. Figure 1 shows these three values against CPU time with the standard deviation for $n = 512$ and $n = 5120$. We employed $C_1 = 0.1/\sqrt{n}$.

We observe that the excess empirical risk of our method,

[2] All the experiments were carried out on Intel Core i7 2.93GHz with 8GB RAM.

[3] In our experiments, $\rho = 0.1$ gave nice performances on all datasets. Too large or too small rho does not give a proper performance, but $\rho = 0.1$ gave stable performances in our experiments.

[4] Even if there is no quadratic term, our method converged with almost the same speed.

SDCA-ADMM, actually converges linearly while other stochastic methods don't show linear convergence. Although Batch-ADMM also shows linear convergence and its convergence speed is comparable to SDCA-ADMM for small sample situation ($n = 512$), SDCA-ADMM is much faster than Batch-ADMM when the number of samples is large ($n = 5120$). As for the classification error, existing stochastic methods also show nice performances despite the poor convergence of the empirical risk. On the other hand, SDCA-ADMM rapidly converges to a stable state and shows comparable or better classification accuracy than existing methods.

6.2. Real Data

Here we execute numerical experiments on real data; '20 Newsgroups'[5] and 'a9a'[6]. '20 Newsgroups' contains 100 dimensional 12,995 training samples and 3,247 test samples. 'a9a' contains 123 dimensional 32,561 training samples and 16,281 test samples. We constructed a similarity graph between features using graph Lasso and applied graph guided regularization as in Ouyang et al. (2013). That is, we applied graph Lasso to the training samples, and obtain a sparse inverse variance-covariance matrix $\hat{F}$. Based on the similarity matrix $\hat{F}$, we connect all index pairs (i, j) with $\hat{F}_{i,j} \neq 0$ on edges. We denote by E the set of edges. Then we impose the following graph guided regularization:

$$\tilde{\psi}(w) = C_1 \textstyle\sum_{i=1}^{p} |w_i| + C_2 \sum_{(i,j)\in E} |w_i - w_j| \\ + 0.01 \times (C_1 \textstyle\sum_{i=1}^{p} |w_i|^2 + C_2 \sum_{(i,j)\in E} |w_i - w_j|^2).$$

Now let F be $|E|\times p$ matrix where $F_{e,i} = 1$ and $F_{e,j} = -1$, if $(i, j) = e \in E$, and $F_{e,i} = 0$ otherwise. Then by letting $B^\top = [\mathrm{I}_p; F]$ and $\psi(u) = C_1 \sum_{i=1}^{p} |u_i| + C_2 \sum_{i=p+1}^{|E|} |u_i| + 0.01(C_1 \sum_{i=1}^{p} |u_i|^2 + C_2 \sum_{i=p+1}^{|E|} |u_i|^2)$ for $u \in \mathbb{R}^{p+|E|}$, we have $\tilde{\psi}(w) = \psi(B^\top w)$. Note that the proximal operation with respect to ψ is just the soft-thresholding operation. In our experiments, we employed $C_2 = C_1|E|/p$ and $C_1 = 0.01/\sqrt{n}$.

We computed the empirical risk on the training data, the averaged loss on the test data, and the test classification error (Figure 2). We observe that the empirical risk on the training data of SDCA-ADMM converges much faster than other methods. Although other methods also performs well on the test loss and the classification error, SDCA-ADMM still converges faster than existing methods with respect to the two quantities measured on the test data.

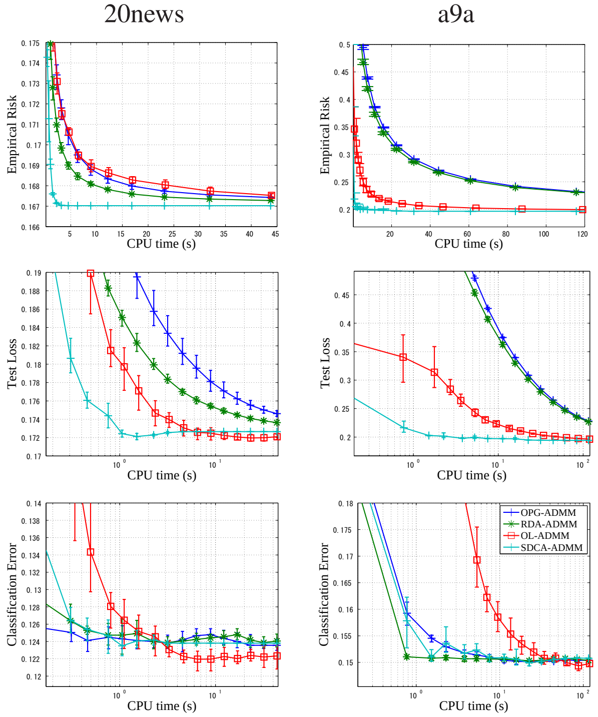

Figure 2. Empirical risk, average loss on the test data and test classification error averaged over 5 independent iteration against CPU time in real data. The error bar indicates the standard deviation.

7. Conclusion

We proposed a new stochastic dual coordinate ascent technique with alternating direction method of multipliers. The proposed method can be applied to wide range of regularization functions. Moreover, we proposed a mini-batch extension of our method. It is shown that, under some strong convexity conditions, our method converges exponentially. According to our analysis, the mini-batch method improves the convergence rate if the input features don't have strong correlation between each other. The numerical experiments showed that our method actually converges exponentially, and the convergence is fast in terms of both empirical and expected risk.

Future work includes that the determination of $\eta_{Z,I}$. In Theorem 2, the exponential convergence is guaranteed if $\eta_{Z,I} >= \{1 + 2\gamma n(1 - 1/K)\}\sigma_{\max}(Z_I^\top Z_I)$. However, in our preliminary numerical experiments, an aggressive method like the one suggested in Takáč et al. (2013) performed effectively in some data sets. Developing more sophisticated determination of $\eta_{Z,I}$ (and G) would be a potentially promising future work.

Acknowledgement TS was partially supported by MEXT Kakenhi 25730013, and the Aihara Project, the FIRST program from JSPS, initiated by CSTP.

[5] Available at http://www.cs.nyu.edu/˜roweis/data.html. We converted the four class classification task into binary classification by grouping category 1,2 and category 3,4 respectively.

[6] Available at 'LIBSVM data sets' http://www.csie.ntu.edu.tw/˜cjlin/libsvmtools/datasets.

References

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2010.

W. Deng and W. Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. Technical report, Rice University CAAM TR12-14, 2012.

J. Duchi and Y. Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2873–2908, 2009.

D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite-element approximations. *Computers & Mathematics with Applications*, 2:17–40, 1976.

M. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory & Applications*, 4:303–320, 1969.

L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

N. Le Roux, M. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for strongly-convex optimization with finite training sets. In *Advances in Neural Information Processing Systems 25*, 2013.

A. Nemirovskii and D. Yudin. *Problem complexity and method efficiency in optimization*. John Wiley, New York, 1983.

H. Ouyang, N. He, L. Q. Tran, and A. Gray. Stochastic alternating direction method of multipliers. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

M. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, London, New York, 1969.

Z. Qin and D. Goldfarb. Structured sparsity via alternating direction methods. *Journal of Machine Learning Research*, 13:1435–1468, 2012.

A. Rakotomamonjy. Applying alternating direction method of multipliers for constrained dictionary learning. *Neurocomputing*, 106:126–136, 2013.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, 1970.

R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1:97–116, 1976.

M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. Technical report, 2013. arXiv:1309.2388.

S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14:567–599, 2013a.

S. Shalev-Shwartz and T. Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In *Advances in Neural Information Processing Systems 26*, 2013b.

S. Shalev-Shwartz and T. Zhang. Proximal stochastic dual coordinate ascent. Technical report, 2013c. arXiv:1211.2717.

M. Signoretto, L. D. Lathauwer, and J. Suykens. Nuclear norms for tensors and their use for convex multilinear estimation. Technical Report 10-186, ESAT-SISTA, K.U.Leuven, 2010.

T. Suzuki. Dual averaging and proximal gradient descent for online alternating direction multiplier method. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

M. Takáč, A. Bijral, P. Richtárik, and N. Srebro. Mini-batch primal and dual methods for SVMs. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

R. Tomioka, T. Suzuki, K. Hayashi, and H. Kashima. Statistical performance of convex tensor decomposition. In *Advances in Neural Information Processing Systems 25*, 2011.

H. Wang and A. Banerjee. Online alternating direction method. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. In *Advances in Neural Information Processing Systems 23*, 2009.

L. Yuan, J. Liu, and J. Ye. Efficient methods for overlapping group lasso. In *Advances in Neural Information Processing Systems 24*, 2011.