STOCHASTIC ALTERNATING DIRECTION METHOD OF MULTIPLIERS FOR STRUCTURED REGULARIZATION

Taiji Suzuki*†

ABSTRACT

In this paper, we present stochastic optimization variants of the alternating direction method of multipliers (ADMM). ADMM is a useful method to solve a regularized risk minimization problem where the regularization term is complicated and not easily dealt with in an ordinary manner. For example, structured regularization is one of the typical applications of such regularization in which ADMM is effective. It includes group lasso regularization, low rank tensor regularization, and fused lasso regularization. Since ADMM is a general method and has wide applications, it is intensively studied and refined these days. However, ADMM is not suited to optimization problems with huge data. To resolve this problem, online stochastic optimization variants and a batch stochastic optimization variant of ADMM are presented. All the presented methods can be easily implemented and have wide applications. Moreover, the theoretical guarantees of the methods are given.

1. Introduction

This paper reviews the recently developed stochastic optimization techniques that are based on the Alternating Direction Method of Multipliers (ADMM) (Suzuki, 2013; 2014). Recently the size of data is increasing in various domains. It often happens that the data can not fit in the memory. To process such large size data, stochastic optimization is a powerful technique. The stochastic optimization techniques basically utilize one or a few observations at each iteration. Therefore, we do not need to load all data at each update. This feature enables us to execute an optimization problem on a huge data set. The stochastic optimization methods proposed so far are well suited for regularized risk minimization. In particular, sparsity-inducing regularization is often utilized to estimate a high dimensional parameter. Due to the non-differentiability of sparse regularization such as $L_1$-regularization, the optimization is not as easy as that of a smooth objective function. The so-called proximal mapping techniques are standard solutions to overcome this difficulty, and they are utilized in various stochastic optimization methods such as online (stochastic) proximal Gradient descent (OPG) and regularized dual averaging method (RDA) (Duchi and Singer, 2009; Xiao, 2009). The proximal mapping technique works well only on a “simple” regularization function. If the regularization is “complicated” as in structured regularization, the proximal mapping often requires heavy computation. Structured sparsity is an important notion to capture complex structures of data. Examples of it include the overlapped group lasso, low rank tensor estimation, and graph lasso (Jacob et al., 2009;
SUZUKI

Signoretto et al., 2010; Tomioka et al., 2011). ADMM is a nice approach to deal with complicated structured regularization (Gabay and Mercier, 1976; Boyd et al., 2010; Qin and Goldfarb, 2012). Thanks to its generality, it is applied to various tasks in statistics, machine learning, and image processing (see the survey of Boyd et al. (2010)). However, ADMM is basically non-stochastic optimization, and thus is not suited to optimization on huge data.

In this paper, we introduce stochastic optimization techniques that are given as an extension of ADMM. Roughly speaking the stochastic optimization techniques are divided into two categories: online stochastic optimization and batch stochastic optimization. As for online methods, we present the online proximal gradient descent type method (OPG-ADMM) and the regularized dual averaging type method (RDA-ADMM) proposed by Suzuki (2013), and as for batch methods, we present stochastic dual coordinate ascent type ADMM (SDCA-ADMM) proposed by Suzuki (2014). An online method updates the parameter sequentially as one (or a few) new observations are obtained. The method looks at each observation just one time, and discards the observations once used. This kind of method is useful when we do not want to wait until the all observations are observed. An online stochastic optimization version of ADMM is proposed by Wang and Banerjee (2012), and its “linearized version” is proposed by Suzuki (2013). On the other hand, a batch method works in a setting in which the whole sample has already been observed and we know the number of observations. In that setting, if the objective function has a nice property such as strong convexity, we can construct a stochastic optimization method that achieves exponential convergence rate as shown in Le Roux et al. (2013); Shalev-Shwartz and Zhang (2013b;a). The method SDCA-ADMM is a stochastic version of ADMM that possesses this property.

We show the convergence rate of those methods. The online methods, OPG-ADMM and RDA-ADMM, achieve the mini-max optimal rate $O(1/\sqrt{T})$ for a non-strongly convex objective and $O(\log(T)/T)$ for a strongly convex objective where $T$ is the number of iterations. The batch method, SDCA-ADMM, achieves the exponential rate $O(\exp(-CT))$ where $C$ is a constant depending on the strong convexity and smoothness of the objective and the number of observations.

2. Regularized Risk Minimization

In statistics and machine learning, we often encounter the following stochastic regularized risk minimization problem:

$$\min_{x \in \mathcal{X}} \mathbb{E}_w[f(x, w)] + \psi(x),$$

where $x$ is the optimization variable (usually called weight vector) contained in a closed convex set $\mathcal{X} \subset \mathbb{R}^m$, $w$ is an observation generated from an (unknown) underlying distribution (for example, $w$ is the explanatory variable), $f(x, w)$ is a loss function that measures error of $x$ for an observation $w$, and $\psi(x)$ is a regularization function that is a penalty on complexity of $x$. We assume both $f(\cdot, w)$ and $\psi(\cdot)$ are convex. In the typical settings, we have only a finite number of observations, and consider an empirical approximation of the expected risk:
Stochastic ADMM for Structured Regularization

\[
\min_{x \in \mathcal{X}} \frac{1}{n} \sum_{i=1}^{n} f(x, w_i) + \tilde{\psi}(x),
\]

(1)

where \(\{w_i\}_{i=1}^{n}\) are i.i.d. observations drawn from the (unknown) distribution. This formulation includes several tasks such as ridge regression, Lasso, logistic regression and support vector machine.

3. Stochastic Optimization for Regularized Risk Minimization

To solve the regularized risk minimization problem, a lot of non-stochastic optimization algorithms have been proposed (Beck and Teboulle, 2009; Figueiredo and Nowak, 2003; Combettes and Wajs, 2005; Tomioka et al., 2012). Since these methods utilize all observations at each iteration, such methods are not efficient when data are so large. Therefore we need an alternative approach to tackle large-size problems.

Stochastic optimization is a promising approach to deal with such large scale data. First we show the online stochastic optimization technique. This approach sequentially draws observations, one at a time, and adequately updates the weight vector based on the single observation obtained at the latest iteration. Here we introduce two representative online stochastic optimization methods. The first method has several different names including online proximal gradient descent, forward-backward splitting (FOBOS) and online mirror descent (Duchi and Singer, 2009; Duchi et al., 2010). Here we utilize the terminology online proximal gradient descent (OPG). Let \(x_t\) be the weight vector at the \(t\)-th step, and \(g_t\) be a member of the sub-gradient of \(f(\cdot, w_t)\) evaluated at \(x_t\): \(g_t \in \nabla_x f(x, w_t)|_{x=x_t}\). Then the update rule of OPG at the \(t\)-th step is as follows:

\[(\text{OPG})\]

\[x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ g_t^\top x + \tilde{\psi}(x) + \frac{1}{2\eta_t} \|x - x_t\|^2 \right\},\]

where \(\eta_t\) is a step size parameter. OPG achieves the minimax optimal regret bound. Typically \(\eta_t\) is set to be decreasing, thus the step size shrinks as the iteration proceeds. The second method, Regularized Dual Averaging (RDA), is developed with an opposite spirit. Let \(\bar{g}_t := \frac{1}{t} \sum_{\tau=1}^{t} g_\tau\). Then the update rule of RDA at the \(t\)-th step is as follows:

\[(\text{RDA})\]

\[x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ \bar{g}_t^\top x + \tilde{\psi}(x) + \frac{1}{2\eta_t} \|x\|^2 \right\}.
\]

In this approach \(\eta_t\) is typically increasing, and the regularization for the new step is vanishing. Thus RDA does not down-grade the importance of newly obtained observations. RDA also achieves the minimax optimal regret, and it is reported that RDA well captures the regularization effect, that is, for sparse learning, RDA usually produces a sparser solution than OPG (Xiao, 2009).

Along with the online stochastic optimization, the batch stochastic optimization techniques such as stochastic average gradient (SAG) (Le Roux et al., 2013), stochastic dual coordinate descent (SDCA) (Shalev-Shwartz and Zhang, 2013b;a), and stochastic variance reduced gradient descent (SVRG) (Johnson and Zhang, 2013; Xiao and Zhang, 2014) have been studied.

The efficiency of these algorithms heavily relies on the fact that the proximal mapping corresponding to the regularization function \(\tilde{\psi}\) can be efficiently computed. Here the proximal mapping corresponding to a function \(\tilde{\psi}\) is the map defined by the following display.
(Rockafellar, 1970):

\[ q \mapsto \arg\min_x \{ \|x - q\|^2 / 2 + \tilde{\psi}(x) \} =: \text{prox}(q|\tilde{\psi}). \]

For example, if the regularization function is \( L_1 \)-norm \( \tilde{\psi}(x) = C \sum_{j=1}^n |x_j| \), then the corresponding proximal mapping is the well-known soft-thresholding operation: \( \hat{x} = \text{prox}(q|\tilde{\psi}) \) is given as \( \hat{x}_j = \text{sign}(q_j) \max(|q_j| - C, 0) \).

However, the proximal mapping can not be efficiently computed for structured regularizations unless we develop a specifically tailored optimization method for each regularization function. We overcome this problem utilizing the idea of the Alternating Direction Multiplier Method (ADMM).

4. Alternating Direction Multiplier Method (ADMM) for Structured Regularization

Here we describe the concept of ADMM and how to apply it to structured regularization problems. Instead of considering the naive optimization problem Eq. (1), we transform the problem into the following linear constraint optimization problem:

\[
\min_{x \in \mathcal{X}, y \in \mathcal{Y}} \frac{1}{n} \sum_{i=1}^n f(x, w_i) + \psi(y), \quad \text{s.t.} \quad Bx = y, \tag{2}
\]

where \( \tilde{\psi}(x) = \psi(Bx) \) with a matrix \( B \in \mathbb{R}^{l \times m} \) and \( \mathcal{Y} \subset \mathbb{R}^l \) is a convex set such that \( Bx \in \mathcal{Y} \) for all \( x \in \mathcal{X} \). Here we assume it is easy to compute the proximal mapping corresponding to \( \psi \). ADMM splits the optimizations with respect to \( x \) and \( y \) utilizing the augmented Lagrangian technique. The iterative scheme of ADMM for the problem (2) is as follows:

\[
x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ \frac{1}{n} \sum_{i=1}^n f(x, w_i) - \lambda_t^\top (Bx - y_t) + \frac{\rho}{2} \|Bx - y_t\|^2 \right\}, \tag{3a}
\]

\[
y_{t+1} = \arg\min_{y \in \mathcal{Y}} \left\{ \psi(y) - \lambda_t^\top (Bx_{t+1} - y) + \frac{\rho}{2} \|Bx_{t+1} - y\|^2 \right\}, \tag{3b}
\]

\[
\lambda_{t+1} = \lambda_t - \rho(Bx_{t+1} - y_{t+1}), \tag{3c}
\]

where \( \lambda_t \) is the dual variable and \( \rho \) is a given parameter. One can see that in ADMM the optimizations with respect to \( x \) and \( y \) are separated into (3a) and (3b). ADMM can be seen as an approximated version of the method of multiplier that minimizes the augmented Lagrangian instead of executing (3a),(3b) (Hestenes, 1969; Powell, 1969; Rockafellar, 1976):

\[
\min_{x,y} \frac{1}{n} \sum_{i=1}^n f(x, w_i) + \psi(y) - \lambda_t^\top (Bx - y) + \frac{\rho}{2} \|Bx - y\|^2,
\]

where the optimizations for \( x \) and \( y \) are not split, but are jointly optimized. On the other hand, in ADMM, thanks to the splitting technique, the update is easily carried out. As for the convergence properties of ADMM, \( O(1/t) \) convergence was proven by He and Yuan (2012) and linear convergence for strongly convex functions was shown by Deng and Yin (2012).
Stochastic ADMM for Structured Regularization

There are several examples of structured regularizations where ADMM is effective. Here, we present some examples of structured sparse regularizations 1.

**Overlapped group lasso** The group lasso imposes a group sparsity as

\[ \hat{\psi}(x) = C \sum_{g \in G} \|x_g\| =: C\|x\|_G, \]

where \( G \) is a set of subsets (groups) of indexes, and \( x_g \) is a restriction of \( x \) onto the index set \( g \) (\( x_g = (x_i)_{i \in g} \)). If groups \( \{g\}_{g \in G} \) have no overlap, then the proximal mapping corresponding to the group lasso regularization is analogous to the soft-thresholding operation. However, if there are overlaps, the proximal mapping can not be straightforwardly computed (Jacob et al., 2009; Yuan et al., 2011). This difficulty can be simply avoided by setting \( B \) and \( \psi \) as follows. Divide \( G \) into its subsets \( G_1, \ldots, G_m \) each of which is a set of non-overlapped groups, let \( Bx \) be a concatenation of \( m \)-repetitions of \( x \), that is, \( Bx = [x; \ldots; x] \), and let \( \psi([x_1; \ldots; x_m]) = C \sum_{i=1}^{m} \|x_i\|_{\psi_i} \). Then one can check that \( \psi(Bx) = C \sum_{i=1}^{m} \|x\|_{G_i} = C \|x\|_G = \hat{\psi}(x) \). Here the proximal mapping corresponding to \( \psi \) can be efficiently computed because, for \( q = [x_1; \ldots; x_m] \),

\[
\arg\min_{y} \left\{ \frac{\|y - q\|^2}{2} + \psi(y) \right\} = \arg\min_{y = [y_1; \ldots; y_m]} \left\{ \sum_{i=1}^{m} \left( \frac{\|y_i - q_i\|^2}{2} + C\|y_i\|_{\psi_i} \right) \right\} = \left( \begin{array}{c} \text{prox} \left( q_1 |C\cdot\psi_1 \right) \\
\vdots \\
\text{prox} \left( q_m |C\cdot\psi_m \right) \end{array} \right).
\]

Thus we can apply the ADMM scheme. See Qin and Goldfarb (2012) for applications of the batch ADMM to the overlapped group lasso.

**Low rank tensor estimation** Recently low rank tenor estimation has been studied by several authors (Signoretto et al., 2010; Tomioka et al., 2011; Kolda and Bader, 2009). A tensor is a multi-way extension of matrix. An example of its application is an estimation of relations in multi-modal data. In collaborative filtering, such data are obtained as users’ preferences on products varying over time (user \( \times \) product \( \times \) time). Suppose the weight vector \( x \) is a vectorization of \( d_1 \times \cdots \times d_m \) tensor \( \mathbf{X} \). We write the tensorization of \( x \) as \( \mathbf{X} \in \mathbb{R}^{d_1 \times \cdots \times d_m} \). The Schatten 1-norm of \( \mathbf{X} \) is the concatenation of the trace norm of the \( k \)-th mode matrix (Signoretto et al., 2010; Kolda and Bader, 2009):

\[ \hat{\psi}(x) = \psi(\mathbf{X}) = C \sum_{k=1}^{m} \|\mathbf{X}_{(k)}\|_{\text{trace}}, \]

where the \( k \)-th mode matrix \( \mathbf{X}_{(k)} \) is the \( d_k \times d_{\backslash k} \) matrix obtained by unfolding all coordinate except the \( k \)-th coordinate (see Kolda and Bader (2009) for the precise definition). The Schatten 1-norm regularization of the tensor imposes low rankness on each mode. In that sense, an estimation with the Schatten 1-norm regularization is called low rank tensor estimation. Here, we may also utilize a similar technique to group lasso: \( Bx \) is the \( m \)-times duplication of \( x \) (\( Bx = [x; \ldots; x] \)) and \( \psi([y_1; \ldots; y_m]) = C \sum_{k=1}^{m} \|\mathbf{Y}_{(k)}\|_{\text{trace}} \) where \( \mathbf{Y}_{(k)} \) is the tensorization of \( y_k \) and \( \mathbf{Y}_{(k)} \) is its \( k \)-th mode. Then \( \psi(Bx) = \hat{\psi}(x) = \)

---

1As for the loss functions \( f \), we may chose any functions depending on the applications. If the problem is regression, we would employ the squared loss as \( f \) and combine that with the regularization functions listed here. Otherwise, if the problem is classification, then the logistic loss or smoothed hinge loss (14) could be employed.
Graph regularization

In graph regularization, we make variables on adjacent vertices similar. To do so, we consider the following type of regularization:

$$C \sum_{k=1}^{m} \|x(k)\|.$$ Therefore by the same reasoning as group lasso, the proximal mapping with respect to $\psi$ is a concatenation of the proximal mapping corresponding to the trace norm: for $y = [y_1; \ldots; y_m]$

$$\text{pro}
\text{x}(y|\psi) = \begin{pmatrix}
\text{vec}(\text{prox}(\mathcal{G}^{(1)}_{(1)} | C \parallel \cdot \parallel_{\text{trace}})) \\
\vdots \\
\text{vec}(\text{prox}(\mathcal{G}^{(m)}_{(m)} | C \parallel \cdot \parallel_{\text{trace}}))
\end{pmatrix},$$

where vec is the vectorization. Here note that the proximal mapping of a matrix $X$ with respect to the trace norm is given as the soft-thresholding operation on the singular values $X$.

Graph regularization

Assume that we are given a graph $\mathcal{G}$. We put each coordinate of the weight vector $x$ on each vertex of $\mathcal{G}$. In graph regularization, we make variables on adjacent vertices similar. To do so, we consider the following type of regularization: $\tilde{\psi}(x) = \sum_{(i,j) \in \mathcal{E}} h(x_i - x_j)$, where $\mathcal{E}$ is the set of edges in the graph $\mathcal{G}$ and $h$ is a penalty function on the discrepancy between adjacent variables. Fused lasso (Tibshirani et al., 2005) and Graph lasso (Jacob et al., 2009) are special cases of this formulation. Here we set $B$ as the adjacent matrix which is a $|\mathcal{E}| \times \dim(x)$ matrix where each row of $B$ corresponds to an edge $(i, j) \in \mathcal{E}$ and has 1 at the $i$-th component, $-1$ at the $j$-th component and 0 otherwise, i.e., $Bx = (x_i - x_j)_{(i,j) \in \mathcal{E}}$. Then, for $\psi(y) = \sum_{e \in \mathcal{E}} h(y_e)$ ($y \in \mathbb{R}^{|\mathcal{E}|}$), we have $\tilde{\psi}(x) = \sum_{e \in \mathcal{E}} h(x_i - x_j) = \psi(Bx)$. One can also observe that the proximal mapping corresponding to $\psi$ can be carried out by a concatenation of the proximal mapping for $h$ on each edge $e \in \mathcal{E}$ as in the previous examples.

5. Online Stochastic ADMM: RDA-ADMM and OPG-ADMM

In this section, we present two algorithms, RDA-ADMM and OPG-ADMM (Suzuki, 2013), as online stochastic versions of ADMM. We give the convergence rates of these methods. For a positive definite matrix $Q$, let $\|x\|_Q$ be $\sqrt{x^\top Q x}$.

5.1. RDA-ADMM

We first introduce Regularized Dual Averaging ADMM (RDA-ADMM) which is a combination of online RDA and ADMM. Here we denote by $G_t$ an arbitrary matrix that is used in the $t$-th update in RDA-ADMM. $G_t$ can depend on any information observed until the $t$-th step. We define $\bar{x}_t, \bar{y}_t$ and $\bar{\lambda}_t$ as $\bar{x}_t = \frac{1}{t} \sum_{\tau=1}^{t} x_\tau, \bar{y}_t = \frac{1}{t} \sum_{\tau=1}^{t} y_\tau$ and $\bar{\lambda}_t = \frac{1}{t} \sum_{\tau=1}^{t} \lambda_\tau$. Then the procedure of RDA-ADMM is summarized in Algorithm 1.

The only difference from the batch-version of ADMM is the update rule of $x_t$ (Eq. (4)). The loss function $\frac{1}{T} \sum_{\tau=1}^{T} f(x, w_\tau)$ is replaced with a linear function $\tilde{g}_t^\top x$. This can be seen as a linear approximation of the loss function, which makes the computation of the update much easier. $y_t$ and $\lambda_t$ are replaced with their averaged versions $\bar{y}_t$ and $\bar{\lambda}_t$. The averaging technique producing $\tilde{g}_t, \bar{y}_t, \bar{\lambda}_t$ works like smoothing and allows us to access the past information at the current update. There is a regularization term $\frac{1}{\sqrt{t}} \|x\|_{G_t}$ that controls the step size. The term $\frac{\rho}{\sqrt{t}} \|Bx\|^2$ is a bit tricky. Unlike the original ADMM, there is a $1/t$ factor in front of $\|Bx\|^2$. This is needed for a technical reason to show $O(1/\sqrt{T})$ convergence. The update rules of $y_t$ and $\lambda_t$ are same as the original ADMM.

At the first glance, it seems that we need to solve a linear equation to obtain $x_{t+1}$. However, by setting $G_t = \gamma I - \frac{\rho m}{t} B^\top B$ with a sufficiently large $\gamma$ such that $G_t$ is positive...
Algorithm 1 RDA-ADMM

Input: $\rho > 0$, $\{\eta_t\}_{t=1}^{T-1}$

Initialize $x_1 = 0$, $y_1 = 0$, $\lambda_1 = 0$.

for $t = 1$ to $T - 1$ do

Observe $w_t$, and compute sub-gradient $g_t \in \nabla_x f(x, w_t)|_{x=x_t}$.

$x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ \bar{y}_t^\top x - \lambda_t^\top B x + \frac{\rho}{2t} \|Bx\|^2 + \rho (Bx_t - \bar{y}_t)^\top B x + \frac{1}{2\eta_t} \|x\|^2_G \right\}$ (4)

Update $y_{t+1}$ and $\lambda_{t+1}$ using the rule (3b) and (3c) respectively.

end for

return $\bar{x}_T := \frac{1}{T} \sum_{t=1}^{T} x_t$ and $\bar{y}_T := \frac{1}{T} \sum_{t=1}^{T} y_t$

definite, the update rule becomes drastically simple:

$$x_{t+1} = \Pi_{\mathcal{X}} \left[ -\frac{\eta_t}{\gamma} \{ \bar{y}_t - B^\top (\lambda_t - \rho B \bar{x}_t + \rho \bar{y}_t) \} \right],$$

where $\Pi_{\mathcal{X}}(x)$ is a projection of $x$ onto the convex set $\mathcal{X}$. The technique to employ a special $G_t$ to cancel the term $\|Bx\|^2$ is called linearization of ADMM and has been used also for the non-stochastic ADMM (Zhang et al., 2011). This is a quite advantageous point compared with the existing online ADMM method (Wang and Banerjee, 2012).

The update rule of $y_t$ is just a proximal mapping corresponding to $\psi$. Indeed, that can be rewritten as

$$y_{t+1} = \arg\min_{y \in \mathcal{Y}} \left\{ \psi(y) + \frac{\rho}{2} \|y - Bx_{t+1} + \lambda_t/\rho\|^2 \right\}$$

$$= \text{prox}(Bx_{t+1} - \lambda_t/\rho \psi/\rho).$$

Therefore, under the assumption that the proximal mapping corresponding to $\psi$ is easily computed, the update of $y_t$ is also efficiently carried out.

5.2. Convergence Analysis of RDA-ADMM

Here, we give convergence analysis of RDA-ADMM. To derive convergence rates, we assume the following conditions.

Assumption 1

(A1) $\mathcal{X}$ and $\mathcal{Y}$ are compact convex sets with radius $R$, i.e., $\forall x, x' \in \mathcal{X}$, $\|x - x'\| \leq R$ and $\forall y, y' \in \mathcal{Y}$, $\|y - y'\| \leq R$. Moreover we assume that $Bx \in \mathcal{Y}$ for all $x \in \mathcal{X}$.

(A2) The sub-gradients of $f(\cdot, w)$ are bounded by $G$, i.e., $\forall a \in \nabla_x f(x, w)$, $\|a\| \leq G$ for all $x, w$.

(A3) The sub-gradients of $\psi$ are bounded by $L_\psi$, i.e., $\forall a \in \nabla \psi(y)$, $\|a\| \leq L_\psi$ at any point $y \in \mathcal{Y}$.

Define the objective function as

$$F(x, y) := E_w[f(x, w)] + \psi(y).$$

In this section, we suppose that $G_t = \gamma I - \rho \eta_t B^\top B/\rho$, and $\gamma, \rho, \eta_t$ are chosen so that $G_t \succeq I$ for simplicity. The particular choice of $G_t$ is not essential, but just for simplicity.
For general $G_t$, we also obtain a similar bound with slight changes of expressions. Moreover we suppose that $\eta_t/t$ is non-increasing.

$(x_t, y_t)$ does not necessarily satisfy the constraint. Therefore we need to modify it. A naive approach is to use $y'_t := Bx_t$ instead of $y_t$. On the other hand, if $B$ is invertible, it is also a natural strategy to use $x'_t := B^{-1}y_t$ instead of $x_t$. We can show that both strategies achieve the minimax optimal rate for the expected risk.

The pair $(x_t, y'_t)$ achieves the following convergence rate of the expected risk. Now let $y'_t := \frac{1}{t} \sum_{\tau=1}^{t} y'_\tau$ and $w_{1:t}$ be the concatenation $(w_1, \ldots, w_t)$.

**Theorem 1 (Convergence rate of RDA-ADMM)** Assume that (A2) and (A3) in Assumption 1 are satisfied. For all $x^* \in X, y^* \in Y$ such that $Bx^* = y^*$, there exists a constant $D$ depending on $\|x^*\|, L_\psi, \rho, \eta_1, B$ such that, the expected risk of RDA-ADMM is bounded as

$$E_{w_{1:T-1}}[F(\bar{x}_T, \bar{y}'_T) - F(x^*, y^*)] + \frac{\rho}{2} E_{w_{1:T}}[\|B\bar{x}_T - \bar{y}'_T\|^2] \leq \frac{1}{T} \sum_{t=2}^{T} \frac{\eta_{t-1}}{2(t-1)} C^2 + \frac{\gamma}{\eta T} \|x^*\|^2 + \frac{D'}{T}.$$

In the same way as the convergence analysis for the pair $(x_t, y'_t)$, we also obtain the convergence rate for the pair $(x'_t, y_t)$.

**Theorem 2** Assume that (A2) and (A3) in Assumption 1 are satisfied. Suppose that $B$ is invertible. For all $x^* \in X, y^* \in Y$ such that $Bx^* = y^*$, there exists a constant $D'$ depending on $\|x^*\|, G, B, \rho, \eta_1$ such that

$$E_{w_{1:T-1}}[F(x'_T, \bar{y}_T) - F(x^*, y^*)] \leq \frac{1}{T} \sum_{t=2}^{T} \frac{\eta_{t-1}}{2(t-1)} C^2 + \frac{\gamma}{\eta T} \|x^*\|^2 + \frac{D'}{T}.$$

In particular, for $\eta_t = \eta_0 \sqrt{t}$, the RHS is further bounded by $C_2/\sqrt{T}$ where $C_2$ is a constant depending on $\|x^*\|, G, B, L_\psi, \rho, \eta_0, \gamma$.

For the proof, see Suzuki (2013). Note that the main terms (the first two terms) are independent of $L_\psi$. If the regularization term is the $L_1$-norm $\psi(y) = C\|y\|_1$, then $L_\psi \leq C \sqrt{p}$ where $p$ is the dimension of $y$, and thus could be large in high dimensional settings. However, the main terms of our bound are free from that. This nice property is due to the fact that our algorithm deals with the regularization term via the proximal mapping. Moreover, the main terms are similar to the convergence rate of the ordinary RDA that is given like

$$\frac{1}{T} \sum_{t=2}^{T} \frac{\eta_{t-1}}{2(t-1)} C^2 + \frac{\|x^*\|^2}{2\eta T}.$$

Thus even though we involve an ADMM structure, the convergence rate is still similar to the ordinary one. It is known that $O(1/\sqrt{T})$ is the minimax optimal rate (Agarwal et al., 2012).

### 5.3. OPG-ADMM

Here we describe the second method, OPG-ADMM, that is a combination of OPG and ADMM. The procedure is summarized in Algorithm 2.
Stochastic ADMM for Structured Regularization

Algorithm 2 OPG-ADMM

Input: $\rho > 0$, $\{\eta_t\}_{t=1}^{T-1}$
Initialize $x_1 = 0$, $y_1 = 0$, $\lambda_1 = 0$.

for $t = 1$ to $T - 1$ do

Observe $w_t$, compute sub-gradient $g_t \in \nabla_x f(x, w_t)|_{x=x_t}$ and calculate $G_t$.

$$x_{t+1} = \text{argmin}_{x \in \mathcal{X}} \left\{ g_t^T x - \lambda_t^T (Bx - y_t) + \frac{\rho}{2} \|Bx - y_t\|^2 + \frac{1}{2\eta_t} \|x - x_t\|^2_{G_t} \right\} \quad (5)$$

Update $y_{t+1}$ and $\lambda_{t+1}$ using the rule (3b) and (3c) respectively.

end for

return $\bar{x}_T := \frac{1}{T} \sum_{t=1}^{T} x_t$ and $\bar{y}_T := \frac{1}{T} \sum_{t=1}^{T} y_t$

The only difference from RDA-ADMM is the update rule of $x_t$ (Eq. (5)). Instead of utilizing the averaged gradient $\bar{g}_t^T x$, the gradient at the current state $g_t^T x$ is used for the linearized loss function, and there is a proximal term $\frac{\rho}{2\eta_t} \|x - x_t\|^2_{G_t}$ that works as a smoothing penalty to get $x_{t+1}$ close to the previous state $x_t$. The update rule of OPG-ADMM is more similar to the original non-stochastic ADMM than that of RDA-ADMM.

Here again, we can avoid solving a linear equation in the update of $x_t$ by choosing $G_t$ appropriately. If we set $G_t = \gamma I - \rho \eta_l B^T B$ with a sufficiently large $\gamma$ such that $G_t$ is positive definite, the update rule becomes as follows:

$$x_{t+1} = \Pi_{\mathcal{X}} \left[ \frac{1}{T} \frac{\eta_t}{\gamma} \{ g_t - B^T (\lambda_t - \rho B x_t + \rho y_t) \} + x_t \right].$$

As in the case of RDA-ADMM, we have a similar convergence rate also for OPG-ADMM. Moreover, if we assume strong convexity on the loss function, we can show a tighter bound for OPG-ADMM. To incorporate strong convexity, we introduce a modulus of strong convexity, $\sigma$, that is a non-negative real such that

$$f(x', w) \geq f(x, w) + (x' - x)^T \nabla_x f(x, w) + \frac{\sigma}{2} \|x - x'\|^2,$$

for all $w$ and $x, x' \in \mathcal{X}$.

**Theorem 3 (Convergence rate of OPG-ADMM)** Assume that all conditions in Assumption 1 are satisfied. Suppose $G_t = \gamma I - \rho \eta_l B^T B$, and $\gamma, \rho, \eta_l$ are chosen so that $G_t \succeq I$. Then, there exists a constant $D$ depending on $R, G, L, \psi, \rho, \eta_l, B$ such that, for all $x^* \in \mathcal{X}, y^* \in \mathcal{Y}$ such that $B x^* = y^*$, the expected risk of OPG-ADMM is bounded as

$$\mathbb{E}_{w_{1:T}}[F(\bar{x}_T, \bar{y}_T) - F(x^*, y^*)]$$

$$\leq \frac{1}{2T} \sum_{t=2}^{T} \max \left\{ \frac{\gamma}{\eta_t} - \frac{\gamma}{\eta_{t-1}} - \sigma, 0 \right\} R^2 + \frac{1}{T} \sum_{t=1}^{T} \frac{\eta_t}{2} \lambda^2 + \frac{D}{T}.$$
We again would like to point out that the main terms are not dependent on the regularization term $\psi$, and are similar to those for the ordinary OPG. Moreover, if $\sigma > 0$, by letting $\eta_t = \gamma_t \sigma$, we have that there exists a constant $C''$ such that

$$E_{w_1,T}[F(\bar{x}_T, \bar{y}_T') - F(x^*, y^*)] \leq C'' \frac{\log(T)}{T}.$$  

As for the pair $(\bar{y}_T, \bar{y}_T')$, we also have analogous convergence results as in the case of RDA-ADMM. See Suzuki (2013) for more details. The convergence rate $O(\log(T)/T)$ is near minimax optimal up to $\log(T)$ order (Agarwal et al., 2012).

6. Stochastic ADMM for batch setting: Stochastic Dual Coordinate Ascent with ADMM

When the number of observations is already known and fixed, the batch type stochastic optimization is effective because that can achieve exponential convergence in contrast to the online type methods presented in the previous section. Here we slightly change the notation. From now on, we solve the following optimization problem:

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

where $z_1, z_2, \ldots, z_n$ 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 observation, and $\psi$ is the regularization function which is used to avoid over-fitting. The Legendre conjugates of convex functions $f$ and $\psi$ are denoted by $f^*$ and $\psi^*$. We define $Z = [z_1, z_2, \ldots, z_n] \in \mathbb{R}^{p \times n}$.

Here, $Z$ is not a random variable but a fixed variable. In a supervised learning setting, the label $y_i$ of the $i$-th observation is hidden in the notation $f_i$. For example, in regression, $f_i(z_i^\top w) = (y_i - z_i^\top w)^2$ where $y_i$ is the response variable.

6.1. Stochastic Dual Coordinate Ascent with ADMM

In this section, we present the stochastic dual coordinate ascent type ADMM (SDCA-ADMM) (Suzuki, 2014). $Z_i$ denotes the $i$-th column of $Z$, which is $z_i$, and $Z_{i \setminus i}$ is a matrix obtained by subtracting the $i$-th column from $Z$. Similarly, for a vector $x$, $x_{i \setminus i}$ is a vector obtained by subtracting the $i$-th component from $x$.

At each iteration, we randomly choose an index set $I \subseteq \{1, \ldots, n\}$ so that each index $i$ is included in $I$ with probability $1/K$; $P(i \in I) = 1/K$ for all $i = 1, \ldots, n$. To do so, we suggest the following procedure. We split the index set $\{1, \ldots, n\}$ into $K$ groups $(I_1, I_2, \ldots, I_K)$ beforehand, and then pick up uniformly $k \in \{1, \ldots, K\}$ 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 $i = 1, \ldots, n$. The update rule using sub-batch is given as follows:
where (7b). The choice of \( I \) the update efficiently, as long as \( x \)

\[
y(t) \leftarrow \arg \min_y \left\{ n\psi(y/n) - \langle w(t-1), Zx(t-1) + By \rangle \right.
\]

\[+ \frac{\rho}{2} \| Zx(t-1) + By \|^2 + \frac{1}{2} \| y - y(t-1) \|^2_Q \}, \tag{7a}
\]

\[
x_I(t) \leftarrow \arg \min_{x_I} \left\{ \sum_{i \in I} f_i^*(x_i) - \langle w(t-1), Z_I x_I + B y(t) \rangle \right.
\]

\[+ \frac{\rho}{2} \| Z_I x_I + Z_{I \setminus I} x_{I \setminus I}(t-1) + B y(t) \|^2 + \frac{1}{2} \| x_I - x_I(t-1) \|^2_{G_{I,I}} \}, \tag{7b}
\]

\[
w(t) \leftarrow w(t-1) - \gamma \rho \left\{ n(Z_I x_I(t) + B y(t)) \right.
\]

\[- (n - n/K)(Z x(t-1) + B y(t-1)) \}. \tag{7c}
\]

Using \( Q \) given as

\[
Q = \rho (\eta_B I_d - B^\top B)
\]

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

\[
y(t) \leftarrow \text{prox} \left( y(t-1) + \frac{B^\top}{\rho \eta_B} \left\{ w(t-1) - \rho(Z x(t-1) + B y(t-1)) \right\} \right| \frac{n\psi^*(y/n)}{\rho \eta_B} \). \tag{9}
\]

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

\[
G_{I,I} = \rho (\eta_{Z,I} I_{|I|} - Z_I^\top Z_I)
\]

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 (7b). 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 \( i = 1, \ldots, n \). Using \( G \) given in (10), the update rule (7b) of \( x(t) \) is rewritten as

\[
x_I(t) \leftarrow \text{prox} \left( x_I(t-1) + \frac{Z_I^\top}{\rho \eta_{Z,I}} \left\{ w(t-1) - \rho(Z x(t-1) + B y(t)) \right\} \right| \frac{\sum_{i \in I} f_i^*(x_i)}{\rho \eta_{Z,I}}, \tag{11}
\]

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 \( i \in I \), \( Z_I = [Z_{i_1}, \ldots, Z_{i_{|I|}}] \). Note that, since \( \sum_{i \in I} f_i^*(x_i) \) is sum of single variable convex functions \( f_i^*(x_i) \), the proximal mapping in Eq. (11) can be split into the proximal mapping 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 \( p_I = x_I(t-1) + \frac{Z_I^\top}{\rho \eta_{Z,I}} \left\{ w(t-1) - \rho(Z x(t-1) + B y(t)) \right\} \), the update rule (11) is reduced to \( x_i(t) \leftarrow \text{prox}(p_i \left| \frac{f_i^*}{\rho \eta_{Z,I}} \right) \) for each \( i \in I \), which is easily parallelizable. In summary, SDCA-ADMM algorithm is given in Algorithm 3.

6.2. Linear Convergence of SDCA-ADMM

In this section, the convergence rate of SDCA-ADMM 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 exists a 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 the set of dual optimum of $y$ as $\mathcal{Y}^*$ and assume that $\mathcal{Y}^*$ is compact.

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

**Assumption 2** There exists $v > 0$ such that, $\forall x_i \in \mathbb{R}$,

\[
f^*_i(x_i) - f^*_i(x_i^*) \geq \langle z_i^* w^*, x_i - x_i^* \rangle + \frac{v\|x_i - x_i^*\|^2}{2}.
\]

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

\[
\psi^*(y/n) - \psi^*(\tilde{y}^*/n) \geq \langle B^T w^*, y/n - \tilde{y}^*/n \rangle + \frac{v_\psi}{2}\|P_{\ker(B)}(y/n - \tilde{y}^*/n)\|^2, \quad (12)
\]

\[
\psi(u) - \psi(B^T w^*) \geq \langle y^*/n, u - B^T w^* \rangle + \frac{h}{2}\|u - B^T w^*\|^2, \quad (13)
\]

where $P_{\ker(B)}$ is the projection matrix to the kernel of $B$. ²

Note that these conditions should be satisfied only around the optimal solutions $(x^*, y^*)$ and $w^*$. They do not need to hold for every point, thus they are 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 the algorithm. The strong convexity of the dual loss $f^*_i$ implies that the primal loss $f_i$ is smooth around the optimal. The condition (12) is satisfied, for example, by $\ell_1$-regularization because the dual of $\ell_1$-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 (12) is restricted on $\ker(B)$. This makes it possible to include several types of regularization functions. Indeed, if $B=I_p$, this condition is always satisfied. The assumption (13) is the strongest assumption. This is satisfied for elastic-net regularization.

²The assumption (12) is rather strong. This can be relaxed like $\psi^*(y/n) - \psi^*(\tilde{y}^*/n) + \frac{v_\psi}{2}\|B(y/n - \tilde{y}^*/n)\| \geq \langle B^T w^*, y/n - \tilde{y}^*/n \rangle + \frac{v_\psi}{2}\|y/n - \tilde{y}^*/n\|_Q^2$ for a sufficiently small $\epsilon > 0$. 

---

---
\(\ell_1\)-regularization could satisfy this condition depending on the optimum \(w^*\) and the solution sequence. If one wants to make this condition always hold, one can just add a small square term, then the condition is satisfied and we obtain an approximated solution which is sufficiently close to the true one within certain precision.

Define the primal and dual objectives as

\[
F_P(w) := \frac{1}{n} \sum_{i=1}^{n} f_i(z_i^* w) + \psi(B^T w),
\]

\[
F_D(x, y) := \frac{1}{n} \sum_{i=1}^{n} f_i^*(x_i) + \psi^*(y) - \langle w^*, Z_n - B \frac{2}{n} \rangle.
\]

Note that, by Fenchel’s duality theorem, \(F_P(w) - F_P(w^*)\) and \(F_D(x, y) - F_D(x^*, y^*)\) are always non-negative. Define the block diagonal matrix \(H\) as \(H_{i,j} = \rho Z_n^T Z_I + G_{i,j}\) for all \(I \in \{I_1, \ldots, I_K\}\) and \(H_{i,j} = 0\) for \((i, j) \notin I_k \times I_k\) \((\forall k)\). Let \(\|y - Y^*\|_Q := \min\{\|y - y^*\|_Q \mid y^* \in Y^*\}\). We define \(R_D(x, y, w)\) as

\[
R_D(x, y, w) := F_D(x, y) - F_D(x^*, y^*) + \frac{\|w - w^*\|_2^2}{2n^2\gamma \rho} + \frac{n(1-\gamma)}{2n^2} \|Z x + B y\|_2^2 + \frac{1}{2n^2} \|x - x^*\|_2^2 \psi_n + H + \frac{\|y - Y^*\|_Q^2}{2nK}.
\]

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

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

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

where

\[
\mu = \min \left\{ \frac{v}{4(v + \sigma_{\max}(H))}, \frac{h \rho \sigma_{\min}(B^T B)}{2 \max\{1/n, 4h \rho, 4h \sigma_{\max}(Q)\}}, \frac{K v \sigma_{\min}(BB^T)}{4 \sigma_{\max}(Q)}, \frac{K v \sigma_{\min}(BB^T)}{4 \sigma_{\max}(Q)(\rho \sigma_{\max}(Z^T Z) + 4v)} \right\}.
\]

In particular,

\[
E[\|w^{(T)} - w^*\|_2^2] \leq \frac{np}{2} \left(1 - \frac{\mu}{K}\right)^T C_1.
\]

If we further assume \(\psi(B^T w) \leq \psi(B^T w^*) + \langle y^*/n, B^T (w - w^*)\rangle + l_1 \|w - w^*\|_2^2 + l_2 \|w - w^*\|^2\) \((\forall w)\), then this implies that

\[
E[F_P(w^{(T)}) - F_P(w^*)] \leq \left(\frac{\sigma_{\max}(Z_n^T Z_n)}{2v} + l_2\right) \frac{np}{2} \left(1 - \frac{\mu}{K}\right)^T C_1 + l_1 \sqrt{\frac{np}{2} \left(1 - \frac{\mu}{K}\right)^T C_1}.
\]

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 observations 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 observations are strongly correlated to each other, $\sigma_{\text{max}}(H)$ 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 the stochastic setting. The proof requires a technique specialized to the stochastic coordinate ascent technique (see Suzuki (2014) for the proof). Here, we would like to emphasize that the convergence is just about the (regularized) empirical risk not for the expected risk. The convergence of the online stochastic methods was shown about the expected risk. This point is critical, and thus should be noted.

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

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

where $C'$ and $C''$ are absolute constants. This says that the dependency of $\epsilon$ is log-order. An interesting point is the influence of $h$, the modulus of local strong convexity of $\psi$. Usually the regularization function is made weaker as the number of observations increases. In that situation, $h$ decreases as $n$ goes up. However, if we set $\rho = v$ and $K = n$, we have

$$T = O\left( \left( n + \frac{1}{\sqrt{h}} \right) \log(n/\epsilon) \right)$$

while the ordinary batch ADMM requires $T = O\left( \frac{n}{\sqrt{h}} \log(n/\epsilon) \right)$. Even if $h = 1/n$, we still have $T = O(n \log(n/\epsilon))$ instead of $O(n^2 \log(n/\epsilon))$. Thus, the convergence rate is hardly affected by the setting of $h$. This point is the same as the ordinary SDCA algorithm (Shalev-Shwartz and Zhang, 2013a). Compared with the iteration number of the ordinary SDCA (Shalev-Shwartz and Zhang, 2013a), that of ADMM involves some additional parameters such as $Q$, $\rho$ and $B$.

7. Numerical Experiments

In this section, we give numerical experiments on artificial and real data to demonstrate the effectiveness of the presented algorithms. All the experiments were carried out on Intel Core i7 2.93GHz with 8GB RAM. The codes were implemented and executed in Matlab.

We compare OPG-ADMM, RDA-ADMM, 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). We also compared those methods with the standard non-stochastic ADMM (Batch-ADMM) in the artificial data sets. We used sub-batch with size 50 for all the methods ($|I_k| = 50$ ($\forall k$), but $|I_K|$ could be less than 50). We employed the parameter settings $\gamma = 1/n$ and $\rho = 0.1$. As for $\eta_Z$, we used $\eta_{Z,I} = 1.1\sigma_{\text{max}}(Z_I^\top Z_I)$ and $\eta_B = \sigma_{\text{max}}(BB^\top) + 1$. All of the experiments are classification problems with structured sparsity. We employed the smoothed hinge loss.

---

3The codes are available at http://www.is.titech.ac.jp/~s-taiji/software/sADMM/sADMM_2014_6_3.zip

4As for the real data sets, we omitted the comparison with Batch-ADMM because it showed much worse performance.
loss:

\[ f_i(u) = \begin{cases} 
0, & (y_i u \geq 1), \\
\frac{1}{2} - y_i u, & (y_i u < 0), \\
\frac{1}{2}(1 - y_i u)^2, & \text{(otherwise)}. 
\end{cases} \]  \quad (14)

Then the proximal mapping with respect to the dual function of the smoothed hinge loss is analytically given by

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

7.1. 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; 2014). 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 \( w \) 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 \( y \) is given by \( y_i = \text{sign}(z_i^T w_0 + \epsilon_i) \) where \( \epsilon_i \) is distributed from normal distribution with mean 0 and standard deviation 0.1.

The group regularization is given as \( \psi(x) = C (\sum_{i=1}^n \|X_{i,i}\| + \sum_{j=1}^{32} \|X_{i,j}\| + 0.01 \times \sum_{i,j} X_{i,j}^2 / 2) \) where \( X \) is the \( 32 \times 32 \) matrix obtained by reshaping \( x \). The quadratic term is added to make the regularization function strongly convex. Even if there is no quadratic term, all methods converged with almost the same speed. Since there exist overlaps between groups, the proximal mapping can not be straightforwardly computed (Jacob et al., 2009). To deal with this regularization function in ADMM framework, we let \( B^T x = [x; x](= [x^T x^T]^T) \), that is \( B = [I_p, I_p] \), and \( \psi([x; x^']) = C (\sum_{i=1}^{32} \|X_{i,i}\| + \sum_{j=1}^{32} \|X_{j,j}\|) \). Then we can see that \( \tilde{\psi}(x) = \psi(B^T x) \) and the proximal mapping with respect to \( \psi \) is analytically obtained; indeed it is easily checked that \( \text{prox}([q; q^']/\psi) = [\text{ST}_{C'}(Q_{:1}/(1 + 0.01C)); \ldots; \text{ST}_{C'}(Q_{:32}/(1 + 0.01C)); \text{ST}_{C'}(Q'_{:1}/(1 + 0.01C)); \ldots; \text{ST}_{C'}(Q'_{:32}/(1 + 0.01C))] \) where \( \text{ST}_{C'}(q) = q \max(1 - C/\|q\|, 0) \) and \( C' = C/(1 + 0.01C) \).

The original RDA requires a direct computation of the proximal mapping 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 \( (E_{(z,y)}[f(y, z^T w(t))]) \) and the classification error \( (E_{(z,y)}[1\{y \neq \text{sign}(z^T w(t))\}]) \). Figure 1 shows these three values against CPU time with the standard deviation for \( n = 512 \) and \( n = 5120 \). We employed \( C = 0.1/\sqrt{n} \).

We observe that the excess empirical risk of 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 a small sample situation \( (n = 512) \), SDCA-ADMM is much faster than Batch-ADMM when the number of observations is large \( (n = 5120) \). As for the classification error, the stochastic ADMM methods (OPG-ADMM and RDA-ADMM) also show nice performances despite the poor convergence of the empirical risk. The non-ADMM online stochastic methods converge
very slowly compared with the ADMM type methods. This is because the computation of the proximal mapping at each iteration has a high cost.

7.2. Real Data

Here we execute numerical experiments on real data sets: ‘20 Newsgroups’ and ‘a9a’. ‘20 Newsgroups’ contains 100 dimensional training data of the size 12,995 (6,500 positive and 6,495 negative) and test data of the size 3,247 (1,624 positive and 1,623 negative). ‘a9a’ contains 123 dimensional training data of the size 32,561 (7,841 positive and 24,720 negative) and test data of the size 16,281 (3,846 positive and 12,435 negative). 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 data, and obtained 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:

---

5Available 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.

6Available at ‘LIBSVM data sets’ http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets.
\[ \tilde{\psi}(w) = C_1 \sum_{i=1}^{p} |w_i| + C_2 \sum_{(i,j) \in E} |w_i - w_j| + 0.01 \times (C_1 \sum_{i=1}^{p} |w_i|^2 + C_2 \sum_{(i,j) \in E} |w_i|^2). \]

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 \( B^\top = [I_p; F] \) and \( \psi(u) = C_1 \sum_{i=1}^{p} |u_i| + C_2 \sum_{i=p+1}^{p+|E|} |u_i| + 0.01(C_1 \sum_{i=1}^{p} |u_i|^2 + C_2 \sum_{i=p+1}^{p+|E|} |u_i|^2) \) for \( u \in \mathbb{R}^{p+|E|} \), we have \( \tilde{\psi}(w) = \psi(B^\top w) \). Note that the proximal mapping with respect to \( \psi \) 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 perform 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.
8. Conclusion

In this paper, we presented recently developed stochastic ADMM methods, namely OPG-ADMM, RDA-ADMM and SDCA-ADMM. OPG-ADMM is the online stochastic gradient descent type modification of ADMM and RDA-ADMM is the online regularized dual averaging type modification of ADMM. SDCA-ADMM is a combination of the stochastic dual coordinate descent method and ADMM. Stochastic optimization is a powerful tool to optimize the regularized risk with huge data. However, ordinary stochastic optimization is not well suited to complicated regularization such as the structured regularization. The presented three methods can be easily applied to various types of structured regularization by employing the same structure as ADMM. They are easily implemented and show nice convergence. In particular, OPG-ADMM and RDA-ADMM achieve $O(1/\sqrt{T})$ convergence in general and OPG-ADMM achieves $O(\log(T)/T)$ convergence for a strongly convex objective. On the other hand, SDCA-ADMM shows $O(\exp(-CT))$ convergence of the empirical risk (not the expected error).

We have shown the effectiveness of the presented methods through numerical experiments on artificial and real data sets. SDCA-ADMM rapidly converged in terms of the empirical risk while the convergence of the other online methods was slow. With regard to the convergence of the test error, there were not large differences among the three methods.

Future work includes utilizing second order information as performed in the online Newton method (Hazan et al., 2007). In this direction, Zhao et al. (2015) proposed a method that is a combination of OPG-ADMM and AdaGrad (Duchi et al., 2011). Another direction is combining our method with Nesterov’s acceleration technique (Nesterov, 1983; 2004; 2013). The study by Azadi and Sra (2014) is in this direction. Another future area of work is to establish a method that choses the best $K$ in SDCA-ADMM. If we employ a large sub-batch, the computational complexity per iteration will be increased but the overhead of memory loading will be decreased. Even if we may assume these computational costs are constant, the iteration number depends on the structure of the data such as correlation. It would be nice to develop a method that adaptively gives the best $K$ and construct an appropriate partition of data.

Acknowledgment

This work was partially supported by MEXT Kakenhi (25730013, 25120012, 26280009 and 15H01678), JST-PRESTO and JST-CREST.

REFERENCES

A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, 2012.

S. Azadi and S. Sra. Towards an optimal stochastic alternating direction method of multipliers. In *Proceedings of the 31st International Conference on Machine Learning*, pages 620–628, 2014.

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

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.
Stochastic ADMM for Structured Regularization

P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200, 2005.

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.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

J. C. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari. Composite objective mirror descent. In *Proceedings of the Annual Conference on Computational Learning Theory*, 2010.

M. A. T. Figueiredo and R. Nowak. An em algorithm for wavelet-based image restoration. *IEEE Transactions on Image Process*, 12:906–916, 2003.

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.

E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

B. He and X. Yuan. On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. *SIAM J. Numerical Analysis*, 50(2):700–709, 2012.

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.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 315–323. Curran Associates, Inc., 2013.

T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 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.

Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, 2004.

Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming, Series B*, 140:125–161, 2013.

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.

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.

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. Proximal stochastic dual coordinate ascent. Technical report, 2013b. 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.

T. Suzuki. Stochastic dual coordinate ascent with alternating direction method of multipliers. In *Proceedings of the 31th International Conference on Machine Learning*, 2014.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of Royal Statistical Society: B*, 67(1):91–108, 2005.

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

R. Tomioka, T. Suzuki, and M. Sugiyama. Super-linear convergence of dual augmented lagrangian algorithm for sparsity regularized estimation. *Journal of Machine Learning Research*, 12:1537–1586, 2012.

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. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24:2057–2075, 2014.

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

X. Q. Zhang, M. Burger, and S. Osher. A unified primal-dual algorithm framework based on Bregman iteration. *Journal of Scientific Computing*, 46(1):20–46, 2011.

P. Zhao, J. Yang, T. Zhang, and P. Li. Adaptive stochastic alternating direction method of multipliers. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 69–77, 2015.

(Received: February 27, 2015, Accepted: August 24, 2015)