Dual Averaging and Proximal Gradient Descent for Online Alternating Direction Multiplier Method

Taiji Suzuki s-taiji@stat.t.u-tokyo.ac.jp
Department of Mathematical Informatics, The University of Tokyo, Tokyo 113-8656, Japan

Abstract

We develop new stochastic optimization methods that are applicable to a wide range of structured regularizations. Basically our methods are combinations of basic stochastic optimization techniques and Alternating Direction Multiplier Method (ADMM). ADMM is a general framework for optimizing a composite function, and has a wide range of applications. We propose two types of online variants of ADMM, which correspond to online proximal gradient descent and regularized dual averaging respectively. The proposed algorithms are computationally efficient and easy to implement. Our methods yield $O(1/\sqrt{T})$ convergence of the expected risk. Moreover, the online proximal gradient descent type method yields $O(\log(T)/T)$ convergence for a strongly convex loss. Numerical experiments show effectiveness of our methods in learning tasks with structured sparsity such as overlapped group lasso.

1. Introduction

Stochastic and online optimization approach is one of the most promising approaches to efficiently process learning tasks on big data. These days, the size of data is rapidly increasing in various domains such as natural language processing, image recognition, signal processing, and bio-informatics. We often encounter such huge data that can not fit in memory. In that situation, sequential learning procedures such as stochastic and online optimizations are quite powerful tools. Moreover high dimensionality is also a common feature of recent data. To cope with high dimensionality, recent developments of online learning algorithms have successfully involved sparse regularizations into online methods, e.g., Forward-Backward-Splitting (FOBOS) and Regularized Dual Averaging method (RDA) (Duchi and Singer, 2009; Xiao, 2009).

The efficiency of online algorithms with sparse regularizations relies on the efficiency of computing the *proximal operation*: $\min_x\{\|x-q\|^2/2+\psi(x)\}$ where ψ is a regularization function. However, for structured sparsity regularizations, it is often difficult to compute the proximal operation. Structured sparsity is an important notion to capture complex structures of data. Examples of that include overlapped group lasso, low rank tensor estimation, and graph lasso (Jacob et al., 2009; Signoretto et al., 2010; Tomioka et al., 2011). Alternating Direction Multiplier Method (ADMM) is a promising optimization method to deal with these kinds of structured regularizations (Gabay and Mercier, 1976; Boyd et al., 2010; Qin and Goldfarb, 2012). The most favorable property of ADMM is its generality. These days, ADMM has been gathering much attentions because it has a broad class of applications in machine learning, image processing, and statistics (see the survey of Boyd et al. (2010)). However, ADMM is basically a batch method which needs to store the whole data in memory. To resolve this issue, Wang and Banerjee (2012) proposed a new method that combines the two notions; stochastic optimization and ADMM. Theoretically, the method posses a favorable convergence property. However, the method needs to solve an exact non-linear optimization in each iteration.

In this paper, we propose a couple of new online variants of ADMM: Online Proximal Gradient descent type method (OPG-ADMM) and Regularized Dual Averaging type method (RDA-ADMM). The core of our proposal is linearization of the loss function. In the existing analysis of online ADMM (Wang and Banerjee, 2012), the optimization of the loss function is assumed to be easily carried out or is given as oracle. However, in practice, that is not necessarily easy and should be carefully addressed. Our

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

proposal gives a solution to this issue by utilizing linearization of the loss function. The main features of our algorithms are as follows:

- (Efficiency) The computations for every update step are efficiently carried out.
- (Generality) They are applicable to a wide class of structured regularizations.
- (Easiness of implement) The algorithms are quite simple and easy to implement.

Moreover we show that the convergence rates of both algorithms achieve $O(1/\sqrt{T})$ which is minimax optimal (Nemirovskii and Yudin, 1983). We also show that OPG-ADMM achieves $O(\log(T)/T)$ convergence for a strongly convex loss. Finally, we present numerical experiments to show the effectiveness of the proposed methods.

Independently of our study, Ouyang et al. (2013) developed the same algorithm as our OPG-ADMM. They gave the convergence rate of the expected risk as in our study. Moreover their analysis includes the tail probability of the risk which is not addressed in this paper.

2. Stochastic Optimization for Regularized Risk Minimization

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

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

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

$$\min_{x\in\mathcal{X}} \ \frac{1}{T}\sum_{t=1}^{T} f(x,w_t) + \tilde{\psi}(x), \tag{1}$$

where $\{w_t\}_{t=1}^T$ are i.i.d. samples drawn from the (unknown) distribution. This formulation includes several machine learning tasks such as SVM, logistic regression, ridge regression and Lasso.

To solve such a regularized risk minimization problem, a lot of "batch" type algorithms have been proposed (Beck and Teboulle, 2009; Figueiredo and Nowak, 2003; Combettes and Wajs, 2005; Tomioka et al., 2012). Since batch type methods maintain all observed samples during the optimization, such methods do not work when data are so large that they cannot fit in memory. Therefore we need an alternative approach to tackle large size problems.

Stochastic optimization is a promising approach to deal with such large scale data. This approach sequentially draws samples, one at a time, and adequately update the weight vector based on the single sample observed at the latest iteration. Here we introduce two representative stochastic optimization methods on the basis of which we develop new 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:

$$\text{(OPG)}\ x_{t+1} = \operatorname*{argmin}_{x\in\mathcal{X}} \left\{ g_t^\top x + \tilde{\psi}(x) + \frac{1}{2\eta_t}\|x - x_t\|^2 \right\},$$

where η_t is a step size parameter. OPG achieves the minimax optimal regret bound. Typically η_t is set to be decreasing, thus the step size shrinks as the iteration proceeds. The second method, Regularized Dual Averaging (RDA), is developed on 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:

$$\text{(RDA)}\quad x_{t+1} = \operatorname*{argmin}_{x\in\mathcal{X}} \left\{ \bar{g}_t^\top x + \tilde{\psi}(x) + \frac{1}{2\eta_t}\|x\|^2 \right\}.$$

In this approach η_t is typically increasing, and the regularization for the new step is vanishing. Thus RDA does not down-grade the importance of newly observed samples. 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).

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

$$q \mapsto \operatorname*{argmin}_x \{\|x-q\|^2/2 + \tilde{\psi}(x)\} =: \mathrm{prox}(q|\tilde{\psi}).$$

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

However, the proximal operation can not be efficiently computed for structured regularizations as presented in Section 5 unless we develop a specifically tailored optimization method for each regularization function. In this article, we overcome this problem utilizing the idea of Alternating Direction Multiplier Method (ADMM).

3. Alternating Direction Multiplier Method (ADMM)

Here we describe the concept of ADMM. We once turn back to the batch situation. Instead of considering the naive optimization problem Eq. (1), we transform the problem into the following linear constraint optimization problem*:

$$\min_{x\in\mathcal{X}, y\in\mathcal{Y}} \ \frac{1}{T}\sum_{t=1}^T f(x, w_t) + \psi(y), \quad \text{s.t.} \ \ Ax = y, \tag{2}$$

where $\tilde{\psi}(x) = \psi(Ax)$ with a matrix $A \in \mathbb{R}^{l\times m}$ and $\mathcal{Y} \subset \mathbb{R}^l$ is a convex set such that $Ax \in \mathcal{Y}$ for all $x \in \mathcal{X}$. Here we assume it is easy to compute the proximal operation corresponding to ψ. 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:

$$x_{t+1} = \operatorname*{argmin}_{x\in\mathcal{X}} \left\{ \frac{1}{T}\sum_{\tau=1}^T f(x, w_\tau) - \lambda_t^\top (Ax - y_t) + \frac{\rho}{2}\|Ax - y_t\|^2 \right\}, \tag{3a}$$

$$y_{t+1} = \operatorname*{argmin}_{y\in\mathcal{Y}} \left\{ \psi(y) - \lambda_t^\top (Ax_{t+1} - y) + \frac{\rho}{2}\|Ax_{t+1} - y\|^2 \right\}, \tag{3b}$$

$$\lambda_{t+1} = \lambda_t - \rho(Ax_{t+1} - y_{t+1}), \tag{3c}$$

where λ_t is the dual variable and ρ 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):

$$\min_{x,y} \ \frac{1}{T}\sum_{\tau=1}^T f(x, w_\tau) + \psi(y) - \lambda_t^\top (Ax - y) + \frac{\rho}{2}\|Ax - y\|^2,$$

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, $O(1/n)$ convergence was proven by He and Yuan (2012) and the linear convergence for strongly convex functions was shown by Deng and Yin (2012).

These days, ADMM has been gathering much attentions because it has a broad class of applications in machine learning, image processing, and statistics (see the survey of Boyd et al. (2010)). However, ADMM is basically a batch algorithm. In the next section, we develop two online versions of ADMM.

4. Our Proposal: RDA-ADMM and OPG-ADMM

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

4.1. 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 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 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 $\bar{g}_t^\top x$. This can be seen as a linear approximation of the loss function, which makes the computation of the update much easier. y_t and λ_t are replaced with their averaged versions $\bar{y}_t$ and $\bar{\lambda}_t$. The averaging technique producing $\bar{g}_t, \bar{y}_t, \bar{\lambda}_t$ works like smoothing and allows us to access the past information at the current update. There is a regularization term $\frac{1}{2\eta_t}\|x\|_{G_t}^2$ that controls the step size. The term $\frac{\rho}{2t}\|Ax\|^2$ is a bit tricky. Unlike the original ADMM, there is $1/t$ factor in front of $\|Ax\|^2$. This is needed for a technical reason to show

*The linear constraint $Ax = y$ can be generalized to $Ax + By = b$. We can show similar convergence rates under this generalization. See the supplementary material for the general results.

Algorithm 1 RDA-ADMM

Input: $\rho > 0$, $\{\eta_t\}_{t=1}^{T-1}$
Initialize $x_1 = \mathbf{0}$, $y_1 = \mathbf{0}$, $\lambda_1 = \mathbf{0}$.
for $t = 1$ **to** $T-1$ **do**
Observe w_t, and Compute sub-gradient $g_t \in \nabla_x f(x, w_t)|_{x=x_t}$.

$$x_{t+1} = \operatorname*{argmin}_{x\in\mathcal{X}} \Big\{ \bar{g}_t^\top x - \bar{\lambda}_t^\top Ax + \frac{\rho}{2t}\|Ax\|^2 + \rho(A\bar{x}_t - \bar{y}_t)^\top Ax + \frac{1}{2\eta_t}\|x\|_{G_t}^2 \Big\} \quad (4)$$

Update y_{t+1} and λ_{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 $\bar{y}_T := \frac{1}{T}\sum_{t=1}^T y_t$

$O(1/\sqrt{T})$ convergence. The update rules of y_t and λ_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 $G_t = \gamma I - \frac{\rho\eta_t}{t}A^\top A$ with a sufficiently large γ such that G_t is positive definite, the update rule becomes drastically simple:

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

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 $\|Ax\|^2$ is called *linearization* of ADMM and has been used also for the batch-type 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 operation corresponding to ψ. Indeed that can be rewritten as

$$\begin{aligned} y_{t+1} &= \operatorname*{argmin}_{y\in\mathcal{Y}} \left\{\psi(y) + \frac{\rho}{2}\|y - Ax_{t+1} + \lambda_t/\rho\|^2\right\} \\ &= \mathrm{prox}\big(Ax_{t+1} - \lambda_t/\rho|\psi/\rho\big). \end{aligned}$$

Therefore, under our assumption that the proximal operation corresponding to ψ is easily computed, the update of y_t is also efficiently carried out.

Convergence Analysis Here, we give convergence analysis of 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., $\forall x, x' \in \mathcal{X}$, $\|x - x'\| \le R$ and $\forall y, y' \in \mathcal{Y}$, $\|y - y'\| \le R$. Moreover we assume that $Ax \in \mathcal{Y}$ for all $x \in \mathcal{X}$.*

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

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

In this section, we suppose that $G_t = \gamma I - \rho\eta_t A^\top A/t$, and γ, ρ, η_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 η_t/t is non-increasing. Then, we have the following regret bound.

Theorem 1. *Let $\lambda^* \in \mathbb{R}^l$ be arbitrary. Then there exists a constant K depending on $R, G, L_\psi, \rho, A, \eta_1, \lambda^*$ such that, for all $x^* \in \mathcal{X}, y^* \in \mathcal{Y}$ satisfying $Ax^* = y^*$,*

$$\begin{aligned} &\frac{1}{T}\sum_{t=1}^T (f(x_t, w_t) + \psi(y_t)) - \frac{1}{T}\sum_{t=1}^T (f(x^*, w_t) + \psi(y^*)) \\ &- \langle\lambda^*, A\bar{x}_T - \bar{y}_T\rangle + \frac{\rho}{2T}\sum_{t=1}^T \|Ax_{t+1} - y_{t+1}\|^2 \\ &\le \frac{1}{T}\sum_{t=2}^T \frac{\eta_{t-1}}{2(t-1)}G^2 + \frac{\gamma}{\eta_T}R^2 + \frac{K}{T}. \end{aligned}$$

The proof can be found in the supplementary material. Basically the proof is a combination of those of the original RDA (Xiao, 2009) and ADMM (He and Yuan, 2012).

Here, we should notice the fact that (x_t, y_t) produced by RDA-ADMM does not necessarily satisfies the linear constraint $Ax_t = y_t$ (see Eq. (2)). Therefore we need to modify x_t or y_t so that the constraint holds. A naive approach is to use $y'_t := Ax_t$ instead of y_t. On the other hand, if A is invertible, it is also a natural strategy to use $x'_t := A^{-1}y_t$ instead of x_t. We can show that both strategies achieve the minimax optimal rate for the expected risk. For notational simplicity, we define

$$F(x, y) := \mathrm{E}_w[f(x, w)] + \psi(y).$$

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

Theorem 2 (Convergence rate of RDA-ADMM)**.** *There exists a constant K depending on $R, L_\psi, \rho, \eta_1, A$*

such that, for all $x^* \in \mathcal{X}, y^* \in \mathcal{Y}$ such that $Ax^* = y^*$, the expected risk of RDA-ADMM is bounded as

$$\begin{aligned}&\mathrm{E}_{w_{1:T-1}}[F(\bar{x}_T, \bar{y}'_T) - F(x^*, y^*)] \\ &+ \tfrac{\rho}{2}\mathrm{E}_{w_{1:T}}[\|A\bar{x}_{T+1} - \bar{y}_{T+1}\|^2] \\ \leq &\frac{1}{T}\sum_{t=2}^{T}\frac{\eta_{t-1}}{2(t-1)}G^2 + \frac{\gamma}{\eta_T}R^2 + \frac{K}{T}.\end{aligned}$$

The proof can be found in the supplementary material. It is shown by using Theorem 1 with a specifically chosen λ^* and Jensen's inequality. We again would like to remark that $\bar{x}_T$ and $\bar{y}'_T$ always satisfies the linear constraint $A\bar{x}_T = \bar{y}'_T$.

The convergence rate is basically same as that of the standard RDA. The first term in the upper bound expresses how aggressively the estimator adapt to the newly observed data, and, on the other hand, the second term expresses a regularization on the estimator's fluctuation. Roughly speaking, these two terms express explore and exploit trade-off. The third term is $O(1/T)$ and is asymptotically negligible. Here $\eta_t = \eta_0\sqrt{t}$ balances the first and the second term. Under this setting, one can show that RDA-ADMM yields $O(1/\sqrt{T})$ convergence: there exists a constant C_1 such that

$$\mathrm{E}_{w_{1:T-1}}[F(\bar{x}_T, \bar{y}'_T) - F(x^*, y^*)] \leq \frac{C_1}{\sqrt{T}}.$$

The convergence rate $O(1/\sqrt{T})$ is known as the minimax optimal rate (Nemirovskii and Yudin, 1983). Therefore our proposed algorithm achieves the optimal rate. Moreover, we observe that $(\bar{x}_t, \bar{y}_t)$ approximately satisfies the constraint $A\bar{x}_t = \bar{y}_t$ for large t. Indeed, Theorem 2 gives $\|A\bar{x}_T - \bar{y}_T\|^2 = O_p(1/\sqrt{T})$ for $\eta_t = \eta_0\sqrt{t}$.

(ii) Convergence analysis for the pair (x'_t, y_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). Obviously the pair (x'_t, y_t) satisfies the linear constraint $Ax'_t = y_t$. Letting $\bar{x}'_t := \frac{1}{t}\sum_{\tau=1}^{t} x'_\tau$, we have the following convergence bound (the proof can be found in the supplementary material).

Theorem 3. *Suppose that A is invertible. For all $x^* \in \mathcal{X}, y^* \in \mathcal{Y}$ such that $Ax^* = y^*$, there exists a constant K' depending on R, G, A, ρ, η_1 such that*

$$\begin{aligned}&\mathrm{E}_{w_{1:T-1}}[F(\bar{x}'_T, \bar{y}_T) - F(x^*, y^*)] \\ &\leq \frac{1}{T}\sum_{t=2}^{T}\frac{\eta_{t-1}}{2(t-1)}G^2 + \frac{\gamma}{\eta_T}R^2 + \frac{K'}{T}.\end{aligned}$$

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 $R, G, A, B, L_\psi, \rho, \eta_0, \gamma$.

Comparing Theorems 2 and 3, the difference of the upper bounds is only the $O(1/T)$ term, which is negligible. Therefore, we have the (almost) same convergence rate for both strategies, $(\bar{x}_t, \bar{y}'_t)$ and $(\bar{x}'_t, \bar{y}_t)$.

4.2. OPG-ADMM

Here we describe our second proposal, OPG-ADMM, that is a combination of OPG and ADMM. The procedure is summarized in Algorithm 2.

Algorithm 2 OPG-ADMM

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

$$\begin{aligned}x_{t+1} = &\operatorname*{argmin}_{x\in\mathcal{X}}\Big\{g_t^\top x - \lambda_t^\top(Ax - y_t) \\ &+ \frac{\rho}{2}\|Ax - y_t\|^2 + \frac{1}{2\eta_t}\|x - x_t\|_{G_t}^2\Big\} \quad (5)\end{aligned}$$

Update y_{t+1} and λ_{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 $\bar{g}_t^\top x$, the gradient at the current state $g_t^\top x$ is used for the linearized loss function, and there is a proximal term $\frac{1}{2\eta_t}\|x - x_t\|_{G_t}^2$ that works as smoothing penalty to let x_{t+1} close to the previous state x_t. The update rule of OPG-ADMM is more similar to the original batch ADMM than that of RDA-ADMM.

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

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

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, σ, that is a non-negative real such that

$$f(x', w) \geq f(x, w) + (x' - x)^\top\nabla_x f(x, w) + \frac{\sigma}{2}\|x - x'\|^2,$$

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

Theorem 4 (Convergence rate of OPG-ADMM)**.** *Suppose $G_t = \gamma I - \rho\eta_t A^\top A$, and γ, ρ, η_t are chosen so that $G_t \succeq I$. Under Assumption 1, there exists a constant K depending on $R, G, L_\psi, \rho, \eta_1, A$ such that, for all $x^* \in \mathcal{X}, y^* \in \mathcal{Y}$ such that $Ax^* = y^*$, the expected risk of OPG-ADMM is bounded as*

$$\begin{aligned}&\mathrm{E}_{w_{1:T-1}}[F(\bar{x}_T, \bar{y}'_T) - F(x^*, y^*)]\\ \le &\frac{1}{2T}\sum_{t=2}^{T} \max\left\{\frac{\gamma}{\eta_t} - \frac{\gamma}{\eta_{t-1}} - \sigma, 0\right\} R^2 + \frac{1}{T}\sum_{t=1}^{T}\frac{\eta_t}{2}G^2 + \frac{K}{T}.\end{aligned}$$

The proof can be found in the supplementary material. Now if we set $\eta_t = \eta_0/\sqrt{t}$, then we also observe that OPG-ADMM shows $O(1/\sqrt{T})$ convergence: there exists a constant C'_1 such that

$$\mathrm{E}_{w_{1:T}}[f(\bar{x}_T, w_T) + \psi(\bar{y}'_T) - (f(x^*, w_T) + \psi(y^*))] \le \frac{C'_1}{\sqrt{T}}.$$

Moreover, if $\sigma > 0$, by letting $\eta_t = \frac{\gamma}{\sigma t}$, we have that there exists a constant C''_1 such that

$$\mathrm{E}_{w_{1:T}}[F(\bar{x}_T, \bar{y}'_T) - F(x^*, y^*)] \le C''_1 \frac{\log(T)}{T}.$$

As for the pair $(\bar{x}'_T, \bar{y}_T)$, we also have analogous convergence results as in the case of RDA-ADMM. See the supplementary material for the detailed proofs.

4.3. Related Work

Recently, Wang and Banerjee (2012) proposed a similar algorithm that is also an online version of ADMM. The different point from our algorithm is that, in the update of x, they don't utilize the linear approximation of loss function, but minimizes the loss function directly as follows:

$$\begin{aligned}x_{t+1} = \operatorname*{argmin}_{x\in\mathcal{X}} \Big\{ &f(x, w_t) - \lambda_t^\top (Ax - y_t)\\ &+ \frac{\rho}{2}\|Ax - y_t\|^2 + \frac{1}{2\eta_t}\|x - x_t\|^2_{G_t}\Big\}. \quad (6)\end{aligned}$$

This update also achieves $O(1/\sqrt{T})$ convergence for a general loss and $O(\log(T)/T)$ for a strongly convex loss. However we need to go through an exact non-linear optimization. This sometimes requires much additional computational cost. In particular, the optimization is not efficient when each draw is *sub-batch*, i.e., w_t at each iteration is a bunch of subsamples $w_t = \{z_{t,1}, \ldots, z_{t,M}\}$ and the loss function is a concatenation of the loss for each sub-sample, $f(x, w_t) = \frac{1}{M}\sum_{i=1}^{M} \ell(x, z_{t,i})$, where $\ell(\cdot,\cdot)$ is a a loss function for sub-samples. The sub-batch technique is often used to stabilize the solution. In this situation, the optimization requires more computational cost than one sample optimization. On the other hand, our method is hardly affected by the increasing size of sub-batch.

Independently of our study, Ouyang et al. (2013) developed the same algorithm as our OPG-ADMM. They also gave the convergence rate of the expected risk such as $O(1/\sqrt{T})$ for a general loss function and $O(\log(T)/T)$ for a strongly convex loss function. Moreover their analysis includes the tail probability of the risk which is not addressed in this paper.

5. Examples of Structured Regularizations

There are several applications of ADMM. In this section, we present some examples of structured sparse regularizations for which our online type ADMM method is effective.

Overlapped group lasso The group lasso imposes a group sparsity as

$$\tilde{\psi}(x) = C\textstyle\sum_{\mathfrak{g}\in\mathfrak{G}} \|x_{\mathfrak{g}}\| =: C\|x\|_{\mathfrak{G}},$$

where $\mathfrak{G}$ is a set of subsets (groups) of indexes, and $x_{\mathfrak{g}}$ is a restriction of x onto the index set $\mathfrak{g}$ ($x_{\mathfrak{g}} = (x_i)_{i\in\mathfrak{g}}$). If groups $\{\mathfrak{g}\}_{\mathfrak{g}\in\mathfrak{G}}$ have no overlap, then the proximal operation corresponding to the group lasso regularization is analogous to the soft-thresholding operation. However, if there are overlaps, the proximal operation can not be straightforwardly computed (Jacob et al., 2009; Yuan et al., 2011). This difficulty can be simply avoided by setting A and ψ as follows. Divide $\mathfrak{G}$ into sets $\mathfrak{G}_1, \ldots, \mathfrak{G}_m$ each of which consists of non-overlapped groups, let Ax be concatenation of m-repetitions of x, that is, $Ax = [x; \ldots; x]$, and let $\psi([x_1; \ldots; x_m]) = C\sum_{i=1}^{m} \|x_i\|_{\mathfrak{G}_i}$. Then one can check that $\psi(Ax) = C\sum_{i=1}^{m} \|x\|_{\mathfrak{G}_i} = C\|x\|_{\mathfrak{G}} = \tilde{\psi}(x)$. Here the proximal operation corresponding to ψ can be efficiently computed because, for $q = [x_1; \ldots; x_m]$,

$$\begin{aligned}&\operatorname*{argmin}_y \left\{\frac{\|y - q\|^2}{2} + \psi(y)\right\}\\ &= \operatorname*{argmin}_{y=[y_1;\ldots;y_m]} \left\{\sum_{i=1}^{m}\left(\frac{\|y_i - q_i\|^2}{2} + C\|y_i\|_{\mathfrak{G}_i}\right)\right\}\\ &= \begin{pmatrix}\mathrm{prox}(q_1|C\|\cdot\|_{\mathfrak{G}_1})\\ \vdots\\ \mathrm{prox}(q_m|C\|\cdot\|_{\mathfrak{G}_m})\end{pmatrix}.\end{aligned}$$

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

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 impose variables on adjacent vertexes are 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 A as the adjacent matrix which is a $|\mathcal{E}|\times\dim(x)$ matrix where each row of A 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., $Ax = (x_i - x_j)_{(i,j)\in\mathcal{E}}$. Then, for $\psi(y) = \sum_{e\in\mathcal{E}} h(y_e)$ $(y \in \mathbb{R}^{|\mathcal{E}|})$, we have $\tilde{\psi}(x) = \sum_{(i,j)\in\mathcal{E}} h(x_i - x_j) = \psi(Ax)$. One can also observe that the proximal operation corresponding to ψ can be carried out by a concatenation of proximal operation for h on each edge $e \in \mathcal{E}$ as in the previous examples.

6. Numerical Experiments

In this section, we demonstrate the performance of the proposed methods through synthetic data and real data. We compare our stochastic ADMMs (RDA-ADMM and OPG-ADMM) with the conventional stochastic optimization methods such as OPG, RDA and the Non-Linearized online ADMM (NL-ADMM) given by Eq. (6)[†]. Through this section, we fix $\eta_0 = 0.01$, $\gamma = 1$ and $\rho = 1$. All problems are classification problems, and we employed logistic loss.

In the experiments, we utilize overlapped group regularizations. To run OPG and RDA, we need to directly compute the proximal operation for the overlapped group lasso penalty. To compute that, we employed the-state-of-the-art dual formulation proposed by Yuan et al. (2011). As for stochastic ADMMs, we used the decomposition technique explained in Section 5, and the pair $(\bar{x}_t, \bar{y}'_t) = (\bar{x}_t, A\bar{x}_t)$ is employed for the t-th step estimator.

6.1. Simulated Sparse Classification Tasks

Here we compare the performances in synthetic data where an overlapped group lasso regularization is imposed. We generated $N = 512$ input feature vectors $\{a_n\}_{n=1}^N$ with dimension $d = 32 \times 32 = 1024$ (each a_n is of 1024 dimension). Each feature is generated from an i.i.d. standard normal. The training output is generated as $c_n = \mathrm{sign}(a_n^\top x^* + \epsilon_n)$ where ϵ_n is a normal distribution with the standard deviation 3.Here the true weight vector x^* is constructed as follows: we generated a 32×32 sparse matrix such that only the first column is non-zero (generated from i.i.d. standard normal) and other columns are zero, and set x^* as the vectorization of that matrix.

As the sample draw w_t in the t-th step, we drew subbatch of size 10, that is, we randomly picked up 10 samples $\{(a_{n_{t,j}}, c_{n_{t,j}})\}_{j=1}^{10} = w_t$ and set $f(x, w_t) = \frac{1}{10}\sum_{j=1}^{10} \ell(c_{n_{t,j}}, a_{n_{t,j}}^\top x)$ where ℓ is the logistic loss.

Then we imposed an overlapped group lasso regularization defined as follows. We converted a weight vector $x \in \mathbb{R}^{1024}$ into a 32×32 matrix, denoted by X, and imposed column-wise and row-wise group regularizations: $\tilde{\psi}(x) = C(\sum_{i=1}^{32} \|X_{i,\cdot}\| + \sum_{j=1}^{32} \|X_{\cdot,j}\|) =: C\|x\|_{\mathrm{block}}$ where $C = 0.025$. This is an overlapped group lasso regularization. Here the expected risk of a weight vector x is $\theta(x) := \frac{1}{N}\sum_{n=1}^{N} \ell(c_n, a_n^\top x) + \tilde{\psi}(x)$.

We independently repeated the experiments 10 times and averaged the excess expected risk: $\theta(\hat{x}) - \min_x \theta(x)$. In Figure 1, the excess expected risk is depicted against the CPU time. In this dataset, OPG-ADMM and RDA-ADMM show almost the same performances while NL-ADMM, RDA and OPG show slower convergence. The main reason for the slow convergence of NL-ADMM is because the nonlinear optimization required in each iteration (6) takes much longer time than the iteration of our methods. We observe that RDA and OPG take even longer time to achieve a certain precision. This is because the proximal operation solved by the dual formulation (Yuan et al., 2011) consumes much time since it requires executing a constrained optimization.

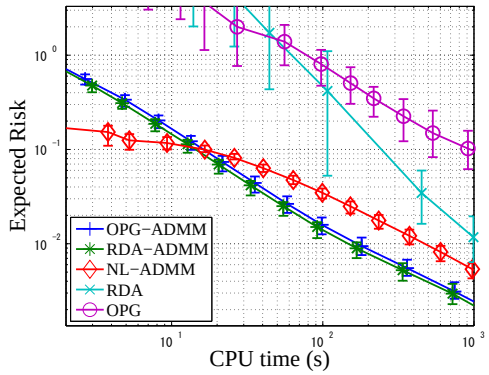

Figure 1. Expected risk averaged over 10 independent repetitions as a function of CPU time (s) on synthetic data. The figure is depicted in log-log scale. The error bars depict one standard deviation of the measurements.

[†] To optimize (6), we applied the Newton method on its dual.

6.2. Real Data Set

Finally, we show the experimental results on a real dataset, 'Adult'[‡]. Adult dataset consists of $N = 32,561$ training samples and $16,281$ test samples with $d = 123$ dimensional feature vector with 0/1 values. In addition to the original 123 features, we took products of features to give additional $15,129$ $(= 123^2)$ features. Then we concatenated the original features and the newly produced features to obtain total $15,252$ features. The weight vector x is also $15,252$ dimensional and we divided the weight vector into two parts $x = [x^{(1)}; x^{(2)}]$ corresponding to the original features and the product features respectively ($x^{(1)}$ is 123 dimensional and $x^{(2)}$ is $15,129$ dimensional). We imposed L_1-regularization on $x^{(1)}$ and the block wise overlapped regularization introduced in the synthetic data on $x^{(2)}$: $\tilde{\psi}(x) = C(\|x^{(1)}\|_1 + \|x^{(2)}\|_{\text{block}}/\sqrt{123})$ with $C = 0.01$.

We again drew sub-batch of size 10 for each step t. We repeated the experiments 10 times and averaged the classification error on the test set. Figure 2 shows the averaged classification error as a function of CPU time. Here, OPG was excluded from the figure because OPG showed much worse performance than the listed methods. We can see that RDA-ADMM shows the best performance followed by NL-ADMM, OPG-ADMM and RDA. Here again we observe that a heavier computation of each iteration of NL-ADMM causes a slower convergence than RDA-ADMM. On the other hand, RDA-ADMM requires a quite light computation for each iteration that leads to a fast convergence. RDA-ADMM also outperforms OPG-ADMM. This is because RDA-ADMM induces a sparser solution that leads to a better generalization error. RDA showed quite slow convergence since the proximal operation required heavy computation because of the high dimensionality. It did not go through sufficient number of iterations in 10^3 seconds, and as a result, did not reach a stable convergence phase.

7. Conclusion

In this paper, we proposed two online variants of ADMM: OPG-ADMM and RDA-ADMM. The proposed methods are applicable to a wide range of structured regularizations, efficiently computed and easy to implement. We have shown that both methods achieve $O(1/\sqrt{T})$ convergence rate for a general loss function, and OPG-ADMM achieves $O(\log(T)/T)$ for a strongly convex loss. In numerical experiments, our methods showed nice convergence behaviors. Overall RDA-ADMM showed favorable performances compared with OPG-ADMM.

An interesting future work is to justify whether RDA-ADMM can also achieve $O(\log(T)/T)$ convergence rate for a strongly convex loss or regularization function.

[‡]We used the preprocessed 'Adult' (a9a) dataset available at 'LIBSVM data sets'.

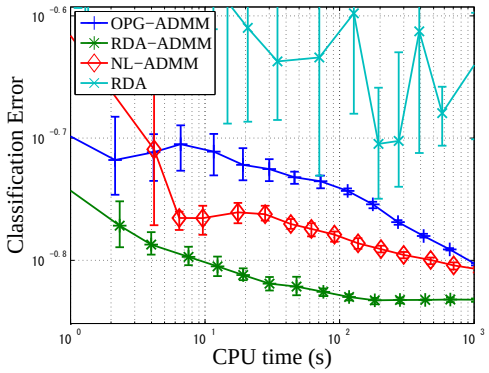

Figure 2. Test classification error averaged over 10 independent repetitions as a function of CPU time (s) on 'Adult' dataset (log-log scale). The error bars depict one standard deviation of the measurements.

Acknowledgement

We would like to thank Hua Ouyang, Niao He, Long Q. Tran, and Alexander Gray for the communication. TS was partially supported by MEXT Kakenhi 22700289, Global COE Program "The Research and Training Center for New Development in Mathematics," and the Aihara Project, the FIRST program from JSPS, initiated by CSTP.

References

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. 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.

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. 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 Trans. 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.

B. He and X. Yuan. On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. *SIAM J. Numerical Analisis*, 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.

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.

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. 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.

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. Superlinear 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. 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.