Efficient Adaptive Online Learning via Frequent Directions

Yuanyu Wan, Nan Wei and Lijun Zhang
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
{wanyy, zhanglj}@lamda.nju.edu.cn, nwei@smai1.nju.edu.cn

Abstract
By employing time-varying proximal functions, adaptive subgradient methods (ADA-GRAD) have improved the regret bound and been widely used in online learning and optimization. However, ADA-GRAD with full matrix proximal functions (ADA-FULL) cannot deal with large-scale problems due to the impractical time and space complexities, though it has better performance when gradients are correlated. In this paper, we propose ADA-FD, an efficient variant of ADA-FULL based on a deterministic matrix sketching technique called frequent directions. Following ADA-FULL, we incorporate our ADA-FD into both primal-dual subgradient method and composite mirror descent method to develop two efficient methods. By maintaining and manipulating low-rank matrices, at each iteration, the space complexity is reduced from $O(d^2)$ to $O(\tau d)$ and the time complexity is reduced from $O(d^3)$ to $O(\tau^2 d)$, where $d$ is the dimensionality of the data and $\tau \ll d$ is the sketching size. Theoretical analysis reveals that the regret of our methods is close to that of ADA-FULL as long as the outer product matrix of gradients is approximately low-rank. Experimental results show that our ADA-FD is comparable to ADA-FULL and outperforms other state-of-the-art algorithms in online convex optimization as well as in training convolutional neural networks (CNN).

1 Introduction
Online learning refers to the process of answering a sequence of questions given answers to previous questions, which enjoys an attractive combination of computational efficiency and theoretical guarantees [Shalev-Shwartz, 2011]. Recently, it has received ever-increasing attention due to the emergence of large-scale applications such as online classification [Preud and Schapire, 1999; Kakade et al., 2008], online metric learning [Jain et al., 2008; Chechik et al., 2010], online matrix factorization [Mairal et al., 2010] and online anomaly detection [Yuan et al., 2015]. Adaptive subgradient methods (ADA-GRAD), which employ proximal functions that adapt to the geometry of the data observed in earlier iterations, are popular for online learning and optimization [Duchi et al., 2011]. According to the type of proximal functions, ADA-GRAD can be divided into learning with full matrix proximal functions (ADA-FULL) and learning with diagonal matrix proximal functions (ADA-DIAG). In contrast to ADA-FULL, ADA-DIAG has been successfully applied to many large-scale applications because of its light computations and storages.

However, the diagonal matrix maintained in ADA-DIAG only contains limited information of gradient outer products. This shortcoming causes the regret of ADA-DIAG worse than that of ADA-FULL when the high-dimensional data is dense and has an approximately low-rank structure. In consideration of this dilemma, Duchi et al. [2011] proposed an open question concerning whether we can efficiently use full matrices in the proximal functions. To solve this problem, Krummenacher et al. [2016] utilized random projections to approximate ADA-FULL, and developed two methods, namely ADA-LR and RADAGRAD. Compared with ADA-FULL, ADA-LR has the same space complexity $O(d^2)$ and a slightly reduced time complexity $O(\tau d^2)$ where $d$ is the dimensionality of the data and $\tau \ll d$ is the number of random projections. Due to the quadratic dependence on $d$, ADA-LR is impractical for high-dimensional data, though it is equipped with a formal regret bound. In contrast, the time and space complexities of RADAGRAD are linear in $d$, but it lacks theoretical guarantees owing to further approximations. In our recent work [Wan and Zhang, 2018], we also utilized random projections to accelerate ADA-FULL, and proposed ADA-DP that has complexities linear in $d$. However, its theoretical guarantees are non-deterministic, and the regret bound contains additional terms that never vanish.

To tackle the above limitations, we develop an efficient variant of ADA-FULL that is also principled. The computational bottleneck of ADA-FULL is to store the outer product matrix of gradients and compute its square root in each round. We note that a similar bottleneck also exists in other second order methods such as online Newton step (ONS) [Hazan et al., 2007], and recently matrix sketching techniques have been used to reduce the computational cost of ONS [Luo et al., 2016]. Motivated by previous works, we propose to employ frequent directions [Ghashami et al., 2016] to approximate the outer product matrix of gradients in ADA-FULL. By utilizing the low-rank structure, the efficient variant of ADA-FULL, named ADA-FD, reduces the space complexity...
from $O(d^2)$ to $O(\tau d)$ and the time complexity from $O(d^3)$ to $O(\tau^2 d)$, which implies both the space and time complexities of our method are linear in the dimensionality $d$.

Moreover, because the idea of AdaGrad can be incorporated into either primal-dual subgradient method [Xiao, 2009] or composite mirror descent method [Duchi et al., 2010], we also develop two efficient methods according to the type of subgradient methods, and prove that our methods enjoy the regret bound close to that of Ada-Full. The slight differences in the regret bound are caused by the approximation error of frequent directions and vanish when the sketching size $\tau$ is larger than the rank of gradient outer products. More generally, when the outer product matrix of gradients is approximately low-rank, the differences do not affect the order of the regret bound. To verify the efficiency and effectiveness of our Ada-FD, we conduct several numerical experiments on online convex optimization and training CNN. The results turn out that our Ada-FD performs comparably with Ada-Full but is much more efficient.

2 Related Work

AdaGrad. Adaptive subgradient methods describe and analyze an apparatus for learning the proximal functions which are modified according to data observed in earlier iterations. This property significantly simplifies choosing the step size while ensures regret guarantees as good as the proximal functions tuned manually. In the following, we provide a brief introduction of AdaGrad for primal-dual subgradient method [Xiao, 2009] and composite mirror descent method [Duchi et al., 2010].

At each round $t = 1, 2, \ldots, T$, the online learner predicts a point $\beta_t \in \mathbb{R}^d$ and then the adversary reveals a composite function $F_t(\beta) = f_t(\beta) + \varphi(\beta)$ where $f_t$ and $\varphi$ are convex. The learner suffers loss $L_t(\beta_t)$ for this round, and the goal of learner is to minimize the regret that is defined as $R(T) = \sum_{t=1}^T F_t(\beta_t) - \sum_{t=1}^T F_t(\beta^*)$ where $\beta^*$ is the fixed optimal predictor. Let $\{g_t, \beta_t\}$ be a particular vector in the subdifferential set $\partial f_t(\beta_t)$ of function $f_t$. The outer product matrix of gradients is defined as $G_t = \sum_{i=1}^t g_t g_t^T$, and $H_t$ has the following two choices:

$$H_t = \begin{cases} \delta I + \text{diag}(G_t)^{1/2} & \text{ADA-DIAG} \\ \delta I + G_t^{1/2} & \text{ADA-FULL} \end{cases}$$

where $\delta > 0$ is a parameter making $H_t$ invertible. The proximal term is given by $\Psi_t(\beta) = \frac{1}{2} \langle \beta, H_t \beta \rangle$ and let $B_{\varphi}(x, y) = \Psi_t(x) - \Psi_t(y) - \frac{1}{2} \langle \nabla \psi_t(y), x - y \rangle$ denote the Bregman divergence associated with $\Psi_t$. Let $\eta > 0$ be a fixed step size and $\tilde{g}_t = \sum_{i=1}^t g_t$. In each iteration, the primal-dual subgradient method updates by

$$\beta_{t+1} = \arg\min_{\beta} \left\{ \eta \left( \langle g_t, \beta \rangle + \varphi(\beta) \right) + \frac{1}{t} \Psi_t(\beta) \right\}$$

(1)

$$= -\eta H_t^{-1} \tilde{g}_t, \quad \text{if } \varphi = 0.$$ 

And the composite mirror descent method updates by

$$\beta_{t+1} = \arg\min_{\beta} \left\{ \eta \left( \langle g_t, \beta \rangle + \varphi(\beta) \right) + B_{\varphi_t}(\beta, \beta_t) \right\}$$

(2)

$$= \beta_t - \eta H_t^{-1} \tilde{g}_t, \quad \text{if } \varphi = 0.$$ 

Because the storage complexity of $G_t$ is $O(d^2)$ and the time complexity of finding its square root is $O(d^3)$, Ada-Full is impractical for large-scale problems.

To reduce the complexities of Ada-Full, Krummenacher et al. [2016] proposed two methods based on the fast randomized singular value decomposition (SVD) [Nalko et al., 2011] to approximate the proximal term $\Psi_t(\beta)$. Let $\Pi \in \mathbb{R}^d \times d$ be the random matrix of the subsampled randomized Fourier transform. They first used the following steps to update $\beta_t$:

$$G_t = G_{t-1} + g_t g_t^T$$

$$\tilde{G}_t = G_t \Pi^T \quad \text{Random Projection}$$

$$QR = \tilde{G}_t \quad \text{QR-decomposition}$$

$$B = Q^T G_t, U \Sigma V^T = B \quad \text{SVD}$$

$$\beta_{t+1} = \beta_t - \eta \left( \Sigma^{1/2} + \delta I \right)^{-1} V^T \tilde{g}_t$$

and the resulting algorithm is referred to as Ada-LR. Note that the space and time complexities of Ada-LR are respectively $O(d^2)$ and $O(\tau^2 d)$, which prevent Ada-LR from being practical for high-dimensional data. By introducing more randomized approximations, they presented a more scalable algorithm RADAGrad that performs the following updates:

$$\tilde{g}_t = \Pi \tilde{g}_t \quad \text{Random Projection}$$

$$\tilde{G}_t = \tilde{G}_{t-1} + g_t g_t^T$$

$$[Q_t, R_t] = \text{qr-update}(Q_{t-1}, R_{t-1}, g_t, \tilde{g}_t)$$

$$B = G_t^T Q_t, U \Sigma V^T = B \quad \text{SVD}$$

$$V = Q_t W, \gamma_t = \eta (\tilde{g}_t - VV^T g_t)$$

$$\beta_{t+1} = \beta_t - \eta \left( \Sigma^{1/2} + \delta I \right)^{-1} V^T \tilde{g}_t - \gamma_t$$

where qr-update means QR decomposition with rank-one updates.

In this case, AdaGrad reduces the space and time complexities to $O(\tau d)$ and $O(\tau^2 d)$ respectively. Unfortunately, it is a heuristic method and lacks theoretical guarantees. When $f_t(\beta_t) = \ell(\beta_t, x_t)$ where $x_t$ is a data vector independent from algorithm, we recently proposed Ada-DP [Wan and Zhang, 2018] based on random projections to attain theoretical guarantees and keep complexities linear in $d$. However, due to the randomness of random projections, the regret bound of Ada-DP is non-deterministic and is worse than Ada-Full even when $\tau = d$.

Matrix Sketching and Frequent Directions. For any given matrix $A \in \mathbb{R}^{n \times d}$, the purpose of sketching is to generate a matrix $B \in \mathbb{R}^{r \times d}$ where $r \ll t$ is the sketching size such that $A \approx B$ or $AA^T \approx B B^T$. There are many matrix sketching techniques via frequent directions [Liberty, 2013; Ghashami and Phillips, 2014; Woodruff, 2014; Ghashami et al., 2016], random projection [Kaski, 1998; Charikar et al., 2004; Woodruff, 2014] and column selection [Drineas et al., 2006]. Frequent directions [Ghashami et al., 2016] is a deterministic matrix sketching technique by extending the well-known algorithm for approximating item frequencies in streams [Misra and Gries, 1982]. Let $B = [b_1, b_2, \ldots, b_r]^T \in \mathbb{R}^{r \times d}$ where $r \ll \min\{t, d\}$. For
**Algorithm 1** Adaptive Dual Averaging via Frequent Directions

1: **Input:** $\eta > 0, \delta > 0, \tau, S_0 = 0_{r \times d}, g_0 = 0, \beta_1 = 0$; 
2: for $t = 1, \ldots, T$ do 
3: Receive gradient $g_t = \nabla f_t(\beta_t)$ 
4: Insert $g_t$ into the last row of $S_t$ and $g_t = g_{t-1} + g_t$ 
5: Compute $U\Sigma V^T = S_{t-1}, \sigma_t = \Sigma_{tt} \geq 0$ 
6: Compute $S_t = \Sigma' V^T$ where $\Sigma' = \sqrt{\Sigma^2 - \sigma_t I_r}$ 
7: $\beta_{t+1} = \beta_t - \eta \left( g_t - V(\delta \text{I}_r + \Sigma')^{-1} \Sigma' V^T g_t \right)$ 
8: end for

any given matrix $A = [a_1, a_2, \ldots, a_t]^T$, frequent directions processes each row of $A$ as follows

$$b_t = \alpha U d_t + \sum_{i=1}^{t} \eta_i g_i g_i^T$$

and products a sketch matrix $B$.

Recently, the techniques of matrix sketching have been used by Luo et al. [2016] to accelerate online Newton step [Hazan et al., 2007]. They studied ONS that updates by

$$A_t = \alpha I_d + \sum_{i=1}^{t} \eta_i g_i g_i^T$$

where $\alpha > 0$ and $\eta_t > 0$ are some parameters for general convex functions, and used matrix sketching techniques to construct a low-rank approximation of the second order information. Motivated by Luo et al. [2016], our work employs frequent directions to calculate a low-rank approximation of the full matrix, which obviously reduces the storage and time complexities of ADA-FULL. Compared to the approximation algorithm used by Krummenacher et al. [2016], frequent directions has two advantages which are deterministic theoretical properties and easy implementations.

**3 An Efficient Variant of ADA-FULL**

In this section, we first describe how to make ADA-FULL more efficient by frequent directions. Then we introduce our proposed methods and the corresponding theoretical results. Finally, we compare our work with Krummenacher et al. [2016] and Luo et al. [2016]. To facilitate presentations, we consider the case $\varphi = 0$, and our methods can be extended to the general case $\varphi \neq 0$.

**3.1 Algorithms**

Define $C_t = [g_1, g_2, \ldots, g_t]^T \in \mathbb{R}^{t \times d}$, where each row is a gradient vector. The outer product matrix of gradients can be calculated as

$$G_t = \sum_{i=1}^{t} g_i g_i^T = C_t^T C_t.$$ 

To accelerate the computation of $H_t^{-1}$, we take advantage of frequent directions to produce a low-rank approximation of $C_t$ denoted by $S_t = [s_1^T, s_2^T, \ldots, s_t^T]^T \in \mathbb{R}^{r \times d}$. Then we can redefine the $H_t$ in the proximal term as

$$H_t = \delta I_d + (S_t^T S_t)^{1/2}.$$ 

**Algorithm 2** Adaptive Mirror Descent via Frequent Directions

1: **Input:** $\eta > 0, \delta > 0, \tau, S_0 = 0_{r \times d}, \beta_1 = 0$; 
2: for $t = 1, \ldots, T$ do 
3: Receive gradient $g_t = \nabla f_t(\beta_t)$ 
4: Insert $g_t$ into the last row of $S_t$ 
5: Compute $U\Sigma V^T = S_{t-1}, \sigma_t = \Sigma_{tt}$ 
6: Compute $S_t = \Sigma' V^T$ where $\Sigma' = \sqrt{\Sigma^2 - \sigma_t I_r}$ 
7: $\beta_{t+1} = \beta_t - \eta \left( g_t - V(\delta I_r + \Sigma')^{-1} \Sigma' V^T g_t \right)$ 
8: end for

Let SVD of $S_t$ be $S_t = U\Sigma V^T$, then $\Sigma = \Sigma_{tt} \geq 0$, and $V \in \mathbb{R}^{r \times d}$. By Woodbury Formula [Hager, 1989], we have

$$H_t^{-1} = (\delta I_d + \Sigma V^T)^{-1} = \frac{1}{\delta} (I_d - V(\delta I_r + \Sigma)^{-1} \Sigma V^T).$$

With the above procedures, we propose an efficient variant of ADA-FULL, named as adaptive online learning via frequent directions (ADA-FD). Then, we first consider to incorporate ADA-FD into primal-dual subgradient method. According to the update rules in (1), in the $t$-th round our algorithm performs the following updates

$$s_t^{t-1} = g_t, \quad \hat{g}_t = g_{t-1} + g_t$$

$$U \Sigma V^T = S_{t-1}, \sigma_t = \Sigma_{tt} \geq 0$$

$$\Sigma' = \sqrt{\Sigma^2 - \sigma_t I_r}, S_t = \Sigma' V^T$$

$$\beta_{t+1} = \beta_t - \eta \left( \hat{g}_t - V(\delta I_r + \Sigma')^{-1} \Sigma' V^T \hat{g}_t \right)$$

The detailed procedures of ADA-FD for primal-dual subgradient method are summarized in Algorithm 1 and this method is named as adaptive dual averaging via frequent directions.

Moreover, we incorporate ADA-FD into composite mirror descent method. According to the update rules in (2), in the $t$-th round our algorithm performs the following updates

$$s_t^{t-1} = g_t$$

$$U \Sigma V^T = S_{t-1}, \sigma_t = \Sigma_{tt} \geq 0$$

$$\Sigma' = \sqrt{\Sigma^2 - \sigma_t I_r}, S_t = \Sigma' V^T$$

$$\beta_{t+1} = \beta_t - \eta \left( g_t - V(\delta I_r + \Sigma')^{-1} \Sigma' V^T g_t \right)$$

The detailed procedures of ADA-FD for composite mirror descent method are summarized in Algorithm 2 and this method is named as adaptive mirror descent via frequent directions.

**Remark** First, the space complexity of our two methods is $O(\tau d)$ caused by the maintenance of $S_t, U, V$. And the time complexity of our two methods is $O(\tau^2 d)$ caused by step 5 in each algorithm which contains SVD. Thus, both of them are linear in the dimensionality $d$. Second, compared with the update rules of ADA-LR (3) and RADAGrad (4), our update rules (6) and (7) do not need random projection and are more simple. Third, when $\varphi \neq 0$, the computational cost of $H_t^{-1}$ can still be reduced dramatically, though the updating of $\beta_t$ may not have close-form solution.
3.2 Theoretical Guarantees

Compared with RADAGRAD, a significant advantage of our methods is that they are equipped with formal theoretical guarantees similar to ADA-FULL. We present regret bounds for Algorithms 1 and 2 in the following two theorems respectively and postpone the detailed analysis to the supplementary material due to the limitation of space.

Theorem 1. Adaptive dual averaging via frequent directions ensures

\[ R(T) \leq \frac{\delta}{2\eta} \|\beta^*\|^2 + \frac{1}{2\eta} \max_{t \leq T} \left( \frac{\sum_{i=1}^{t-1} \sigma_i}{t} \right) \text{tr}(G^{1/2}) + 2\eta \max_{t \leq T} \left( \frac{\sum_{i=1}^{t-1} \sigma_i}{t} \right) \text{tr}(G^{1/2}) \]

where \( \Delta_T = \sum_{t=1}^{T} \sigma_t \).

Theorem 2. Adaptive mirror descent via frequent directions ensures

\[ R(T) \leq \frac{\delta}{2\eta} \|\beta^*\|^2 + \frac{1}{2\eta} \max_{t \leq T} \left( \frac{\sum_{i=1}^{t-1} \sigma_i}{t} \right) \text{tr}(G^{1/2}) + 2\eta \max_{t \leq T} \left( \frac{\sum_{i=1}^{t-1} \sigma_i}{t} \right) \text{tr}(G^{1/2}) \]

where \( \Delta_T = \sum_{t=1}^{T} \sigma_t \).

For comparisons, let us introduce the theoretical guarantees of ADA-FULL below.

Theorem 3. (Theorem 7 of Duchi et al. [2011]) Provided with \( \delta \geq \max_{t \leq T} \|g_t\|_2 \), ADA-FULL incorporated into primal-dual subgradient method ensures

\[ R(T) \leq \frac{\delta}{2\eta} \|\beta^*\|^2 + \frac{1}{2\eta} \max_{t \leq T} \left( \frac{\sum_{i=1}^{t-1} \sigma_i}{t} \right) \text{tr}(G^{1/2}) + 2\eta \text{tr}(G^{1/2}). \]

ADA-FULL incorporated into composite mirror descent method ensures

\[ R(T) \leq \frac{\sigma}{2\eta} \|\beta^*\|^2 + \frac{1}{2\eta} \max_{t \leq T} \left( \frac{\sum_{i=1}^{t-1} \sigma_i}{t} \right) \text{tr}(G^{1/2}) + 2\eta \text{tr}(G^{1/2}). \]

Remark. Compared with the Theorem 3, we confirm that our methods enjoy the regret bound close to that of ADA-FULL, and we point out the slight differences as following. First, the regret bounds of our two methods contain an additional term, respectively, \( \frac{\sum_{i=1}^{t-1} \sigma_i}{t} \max_{t \leq T} \|\beta_{t+1}\|^2 \) and \( \frac{\sigma}{2\eta} \max_{t \leq T} \|\beta^* - \beta_t\|^2 \). According to Cauchy-Schwarz inequality and \( \Delta_T = \sum_{t=1}^{T} \sigma_t \), we have

\[ \sum_{t=1}^{T} \sqrt{\sigma_t} \leq \sqrt{T} \sqrt{\sum_{t=1}^{T} \sigma_t} = \sqrt{T} \Delta_T \]

which means the additional term does not affect the order of the regret bounds when \( \Delta_T \) is small. Second, our two bounds magnify the third term by \( \max \left( \frac{\max_{t \leq T} \|g_t\|_2 + \sqrt{\Delta_T}}{\delta}, \frac{1}{2\eta} \text{tr}(G^{1/2}), \frac{\sigma}{2\eta} \text{tr}(G^{1/2}) \right) \) and \( \max \left( \frac{\sqrt{\Delta_T}}{\delta}, \frac{1}{2\eta} \text{tr}(G^{1/2}), \frac{\sigma}{2\eta} \text{tr}(G^{1/2}) \right) \) respectively, which does not change the order of the regret bounds when \( \Delta_T \) is small. Let \( C^\tau_p \) denote the minimizer of \( \|C_t - C^\tau_p\|_F \) over all rank \( k \) matrices. According to the property of frequent directions, we have

\[ \Delta_T \leq \|C_T - C^\tau_p\|_F / (\tau - k) \]

for any \( k < \tau \), which means \( \Delta_T \) is small when the outer product matrix of gradients is approximately low-rank. Moreover, when \( G_T \) is low-rank and \( \text{rank}(G_T) = r \), we have \( \Delta_T = 0 \) by choosing \( \tau = r + 1 \).

3.3 Discussions

We would like to emphasize the differences between our work and Krummenacher et al. [2016]. First, both ADA-LR and RADAGRAD only consider composite mirror descent method, ignoring primal-dual subgradient method. Second, unlike random projections adopted in Krummenacher et al. [2016], our work makes use of frequent directions that belongs to deterministic matrix sketching techniques. As a result, our methods are much more simple than ADA-LR and RADAGRAD while the theoretical guarantees of our methods are deterministic, contrary to the probabilistic regret bound of ADA-LR. Third, our methods are very efficient in the sense that the time and storage complexities are linear in the dimensionality \( d \), without losing regret bound. By contrast, RADAGRAD reaches similar complexities while does not have theoretical Justifications.

Next, we discuss the differences between our work and Luo et al. [2016]. First, our methods and Luo et al. [2016] are designed for different tasks. Our goal is to develop an efficient variant of ADA-FULL, and the goal of Luo et al. [2016] is to accelerate ONS. Note that ADA-FULL is a data dependent algorithm for general convex functions and ONS is proposed to derive a logarithmic regret for exp-concave functions. Second, the theoretical analysis in our work is obviously different from Luo et al. [2016]. They provide a regret bound of ONS with order \( O(\sqrt{Td}) \) for general convex functions by setting \( \eta_t = O(1/\sqrt{t}) \). Although its update rules (5) can be reformulated as

\[ H_t = \delta I_d + \sum_{t=1}^{T} \frac{1}{\sqrt{t}} g_t g_t^\top \text{ and } \beta_{t+1} = \beta_t - \eta H_t^{-1} g_t \]

which is similar to RADAGRAD, its \( O(\sqrt{Td}) \) bound destroys the data dependent property. Third, the main algorithm of Luo et al. [2016] can be regarded as a mirror descent method. In contrast, our work proposes two efficient methods for primal-dual subgradient method and composite mirror descent method respectively.

4 Experiments

In this section, we perform numerical experiments to verify the efficiency and effectiveness of our ADA-FD. First, we
show that ADA-FD approximates ADA-FULL well and outperforms ADA-DIAG. Second, we compare ADA-FD with RADA\textsc{Grad} that is also an approximation of ADA-FULL and FD-SON [Luo et al., 2016] that is an approximation of ONS by frequent directions.

### 4.1 Online Regression

First, we consider the problem of online regression where 
\[ f_t(\beta) = |\beta^T x_t - y_t| \]
and \[ y_t = \beta^T x_t \] with ideal synthetic data. Specifically, we set \( T = 10,000, d = 500 \) and \( \beta_* = \beta_*/\|\beta_*\|_2 \) where each entry of \( \beta_* \) is drawn independently from \( N(0, 1) \). In order to meet the requirement of approximately low-rank structure, each data point \( x_t \) is sampled independently from a Gaussian distribution \( N(\mu, \Sigma) \) where \( \mu = 1 \) and \( \Sigma \) has rapidly decaying eigenvalues \( \lambda_j(\Sigma) = \lambda_0^{-j} \) with \( \alpha = 2 \) and \( \lambda_0 = 100 \).

For primal-dual subgradient method, we compare ADA-FD against ADA-FULL and ADA-DIAG. And for composite mirror descent method, we compare ADA-FD against ADA-FULL, ADA-DIAG, RADA\textsc{Grad} and FD-SON. Parameters \( \eta \) and \( \delta \) are searched in \( \{1e-4, 1e-3, \ldots, 100\} \), and we choose the best values for each algorithm. For fairness, all algorithms are run with the same permutation of the function sequence. Because of the randomness of RADA\textsc{Grad}, its results are averaged over 5 runs. Figure 1 shows the comparison of regret among different algorithms and the comparison of running time among different algorithms for composite mirror descent (CMD) method among different algorithms on Gisette and Epsilon. We find that ADA-FD outperforms ADA-DIAG, because it is significantly faster than ADA-FULL. For composite mirror descent method, our ADA-FD outperforms ADA-FULL and is significantly faster than ADA-FD-FULL. For composite mirror descent method, our ADA-FD outperforms RADA\textsc{Grad} and FD-SON. From the comparison of running time, we find that ADA-FD is faster than RADA\textsc{Grad} and as fast as FD-SON. The comparison of running time for following experiments is similar and can be found in the supplementary.

### 4.2 Online Classification

Then, we perform online classification to evaluate the performance of our ADA-FD with two real world datasets from LIB\textsc{SVM} repository [Chang and Lin, 2011]: Gisette and Epsilon which are high-dimensional and dense. At each round, the learning algorithm receives a single example \((x_t, y_t)\) and suffers the squared hinge loss \( f_t(\beta) = \frac{1}{2} (\max \{0, 1 - y_t \beta^T x_t\})^2 \). Each learning algorithm ends with a single pass through the training data. Following Duchi et al. [2011], we adopt two metrics to measure the performance: the online mistakes and the offline accuracy on the testing data. For primal-dual subgradient method, we compare ADA-FD against ADA-DIAG. And for composite mirror descent method, we compare ADA-FD against ADA-DIAG, RADA\textsc{Grad} and FD-SON. We omit the result of ADA-FULL, because it is too slow. Similar as before, parameters \( \eta \) and \( \delta \) are searched in \( \{2e-4, 2e-3, \ldots, 20\} \), and we choose the best values for each algorithm. To reduce the computational cost, we set the sketching size \( \tau = 10 \) for Gisette and \( \tau = 40 \) for Epsilon.

Both datasets are divided into training part and testing part, and the numbers of training and testing examples are shown in Table 1. For training data, we generate 5 random permutations, and report the average results. Figures 2 and 3 show the comparison of mistakes and test accuracy during training among different algorithms on Gisette and Epsilon. We find that our ADA-FD outperforms ADA-DIAG, RADA\textsc{Grad} and FD-SON on both datasets. Note that RADA\textsc{Grad} is outperformed by ADA-DIAG in term of test accuracy on Epsilon as shown in Figure 3(d). It means that our ADA-FD has better ability to approximate ADA-FULL than RADA\textsc{Grad}.

![Figure 1: The comparison of regret among different algorithms on synthetic data for primal-dual subgradient (PDS) method and composite mirror descent (CMD) method.](image)

![Figure 2: The comparison of mistakes and test accuracy among different algorithms on Gisette for primal-dual subgradient (PDS) method and composite mirror descent (CMD) method.](image)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Examples</th>
<th>#Features</th>
<th>#Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gisette</td>
<td>6,000/1,000</td>
<td>5,000</td>
<td>2</td>
</tr>
<tr>
<td>Epsilon</td>
<td>400,000/100,000</td>
<td>2,000</td>
<td>2</td>
</tr>
<tr>
<td>MNIST</td>
<td>60,000/10,000</td>
<td>784</td>
<td>10</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>50,000/10,000</td>
<td>3,072</td>
<td>10</td>
</tr>
<tr>
<td>SVHN</td>
<td>73,257/26,032</td>
<td>3,072</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 1: Datasets used in experiments
4.3 Non-convex Optimization in CNN

Besides convex optimization, ADA-DIAG incorporated into composite mirror descent method has also been applied to non-convex optimization such as training neural networks and has performed well. Therefore, we take convolutional neural networks (CNN) as an example and compare ADA-FD against ADA-DIAG, RADAGrad and FD-SON on training the classical neural networks. We consider a 4-layer CNN from Keras examples directory, which contains two $3 \times 3$ convolutional layers and one fully connected layer (128 hidden units followed by 0.5 dropout). The activation function is ReLU. The first convolutional layer generates 32 channels, and the second convolutional layer generates 64 channels followed by $2 \times 2$ max-pooling and 0.25 dropout. The final layer is a softmax layer and the objective is cross-entropy loss. We use this simple and standard architecture to perform classification on the MNIST [LeCun et al., 1998], CIFAR10 [Krizhevsky, 2009] and SVHN datasets [Netzer et al., 2011].

Parameters $\eta$ and $\delta$ of all algorithms are searched in $\{1e^{-8}, 1e^{-7}, \cdots, 1\}$, and we choose the best values for each algorithm. Following Krummenacher et al. [2016], we only consider applying ADA-FD, RADAGrad and FD-SON to the convolutional layers, and other layers are still trained with ADA-DIAG. To reduce the computational cost, we set the sketching size $\tau = 20$ for all datasets. For all algorithms, we run 5 times with batch size 128 and report the average results. Figure 4 shows the comparison of training loss and test accuracy during training among different algorithms. We find that our ADA-FD outperforms ADA-DIAG, RADAGrad and FD-SON on all datasets when applied to training CNN. We note that RADAGrad is outperformed by ADA-DIAG in terms of training loss on CIFAR10 as shown in Figure 4(c). This result once again validates that our ADA-FD has better ability to approximate ADA-FULL than RADAGrad.

5 Conclusions

In this paper, we present ADA-FD to approximate ADA-FULL using frequent directions. According to the type of subgradient methods, we develop two efficient methods. The time and space complexities of both algorithms are linear in the dimensionality $d$, which means our methods have similar efficiency compared to ADA-DIAG. Furthermore, according to our theoretical analysis, our methods enjoy the regret bound close to that of ADA-FULL when the outer product matrix of gradients is approximately low-rank. Numerical experiments show that our ADA-FD outperforms ADA-DIAG, RADAGrad and FD-SON and is close to ADA-FULL.

Acknowledgments

This work was partially supported by the National Key R&D Program of China (SQ2018YFB100002), NSFC (61603177), JiangsuSF (BK20160658), YESS (2017QNRC001), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.
References


