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@smail.nju.edu.cn

Abstract

By employing time-varying proximal functions, adaptive subgradient methods (ADAGRAD) 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 [Freund 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 (ADAGRAD), 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

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

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

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

$$\boldsymbol{\beta}_{t+1} = \min_{\boldsymbol{\beta}} \left\{ \eta \left\langle \frac{1}{t}\bar{\mathbf{g}}_t, \boldsymbol{\beta} \right\rangle + \eta \varphi(\boldsymbol{\beta}) + \frac{1}{t}\Psi_t(\boldsymbol{\beta}) \right\} \quad (1)$$
$$= -\eta \psi_t^{-1} \bar{\mathbf{g}}_t, \quad \text{if } \varphi = 0.$$

And the composite mirror descent method updates by

$$\boldsymbol{\beta}_{t+1} = \min_{\boldsymbol{\beta}} \{ \eta \langle \mathbf{g}_t, \boldsymbol{\beta} \rangle + \eta \varphi(\boldsymbol{\beta}) + B_{\Psi_t}(\boldsymbol{\beta}, \boldsymbol{\beta}_t) \} \quad (2)$$
$$= \boldsymbol{\beta}_t - \eta \psi_t^{-1} \mathbf{g}_t, \quad \text{if } \varphi = 0.$$

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 $\Psi_t(\boldsymbol{\beta})$. Let $\Pi \in \mathbb{R}^{\tau \times d}$ be the random matrix of the subsampled randomized Fourier transform. They first used the following steps to update $\boldsymbol{\beta}_t$:

$$\begin{aligned} G_t &= G_{t-1} + \mathbf{g}_t \mathbf{g}_t^\top \\ \widetilde{B}_t &= B_t \Pi^\top && \text{Random Projection} \\ B_t &= \widetilde{B}_t && \text{QR-decomposition} \\ U_t, \Sigma_t, V_t &= \text{SVD}(B_t) \\ \boldsymbol{\beta}_{t+1} &= \boldsymbol{\beta}_t - \eta \left(\Sigma^{1/2} + \delta I_\tau \right)^{-1} V_t^\top \mathbf{g}_t \end{aligned} \quad (3)$$

and the resulting algorithm is refereed to as ADA-LR. Note that the space and time complexities of ADA-LR are respectively $O(d^2)$ and $O(\tau d^2)$, 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:

$$\begin{aligned} \widetilde{\mathbf{g}}_t &= \Pi \mathbf{g}_t && \text{Random Projection} \\ \widetilde{B}_t &= B_{t-1} + \mathbf{g}_t \widetilde{\mathbf{g}}_t^\top \\ B_t, Q_t &= \text{qr\_update}(B_{t-1}, Q_{t-1}, \mathbf{g}_t, \widetilde{\mathbf{g}}_t) \\ U_t, \Sigma_t, V_t &= \text{SVD}(\widetilde{B}_t) \\ \gamma_t &= \eta (\mathbf{g}_t - V_t V_t^\top \mathbf{g}_t) \\ \boldsymbol{\beta}_{t+1} &= \boldsymbol{\beta}_t - \eta \left(\Sigma^{1/2} + \delta I_\tau \right)^{-1} V_t^\top \mathbf{g}_t - \gamma_t \end{aligned} \quad (4)$$

where qr_update means QR decomposition with rank-one updates. In this case, RADAGRAD 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 $f_t(\boldsymbol{\beta}_t) = \ell(\boldsymbol{\beta}_t^\top \mathbf{x}_t)$ where $\mathbf{x}_t$ 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 $A \in \mathbb{R}^{t \times d}$, the purpose of sketching is to generate a matrix $B \in \mathbb{R}^{\tau \times d}$ where $\tau \ll t$ is the sketching size such that $A \approx B$ or $A^\top \approx B B^\top$. 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 $B = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_\tau]^\top = \mathbf{0}_{\tau \times d}$ where $\tau \ll \min\{t, d\}$. For

Algorithm 1 Adaptive Dual Averaging via Frequent Directions

1: **Input:** $\eta > 0, \delta > 0, \tau,$ $_0$ $\mathbf{0}_{\tau\times d}, \bar{\mathbf{g}}_0$ $\mathbf{0}, \boldsymbol{\beta}_1$ $\mathbf{0}$;
2: **for** $1, ..., T$ **do**
3: Receive gradient $\mathbf{g}_t$ $\nabla f_t(\boldsymbol{\beta}_t)$
4: Insert $\mathbf{g}_t$ into the last row of $_{t-1}$ and $\bar{\mathbf{g}}_t$ $\bar{\mathbf{g}}_{t-1} + \mathbf{g}_t$
5: Compute $^\top$ $_{t-1}, \sigma_t$ Σ_{tt}^2
6: Compute $_t$ Σ' $^\top$ where Σ' $\sqrt{\Sigma^2 - \sigma_t}$
7: $\boldsymbol{\beta}_{t+1}$ $-\frac{\eta}{\delta}\big(\bar{\mathbf{g}}_t -$ $(\delta$ $+\Sigma')^{-1}\Sigma'$ $^\top\bar{\mathbf{g}}_t\big)$
8: **end for**

Algorithm 2 Adaptive Mirror Descent via Frequent Directions

1: **Input:** $\eta > 0, \delta > 0, \tau,$ $_0$ $\mathbf{0}_{\tau\times d}, \boldsymbol{\beta}_1$ $\mathbf{0}$;
2: **for** $1, ..., T$ **do**
3: Receive gradient $\mathbf{g}_t$ $\nabla f_t(\boldsymbol{\beta}_t)$
4: Insert $\mathbf{g}_t$ into the last row of $_{t-1}$
5: Compute $^\top$ $_{t-1}, \sigma_t$ Σ_{tt}^2
6: Compute $_t$ Σ' $^\top$ where Σ' $\sqrt{\Sigma^2 - \sigma_t}$
7: $\boldsymbol{\beta}_{t+1}$ $\boldsymbol{\beta}_t - \frac{\eta}{\delta}\big(\mathbf{g}_t -$ $(\delta$ $+\Sigma')^{-1}\Sigma'$ $^\top\mathbf{g}_t\big)$
8: **end for**

any given matrix $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_t$ $^\top$, frequent directions processes each row of as follows

$\mathbf{b}_\tau$ $\mathbf{a}_i,$ $^\top$ SVD

$\sqrt{\Sigma^2 - \sigma}$ $^\top$ where σ $\Sigma_{\tau\tau}^2$

and products a sketch matrix.

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

$$_t \quad \alpha + \sum_{i=1}^{t} \eta_i \mathbf{g}_i \mathbf{g}_i^\top \text{ and } \boldsymbol{\beta}_{t+1} \quad \boldsymbol{\beta}_t - \quad ^{-1}\mathbf{g}_t \qquad (5)$$

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 φ 0, and our methods can be extended to the general case φ / 0.

3.1 Algorithms

Define $_t$ $\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_t$ $^\top \in \mathbb{R}^{t\times d}$, where each row is a gradient vector. The outer product matrix of gradients can be calculated as

$$_t \quad \sum_{i=1}^{t} \mathbf{g}_i \mathbf{g}_i^\top \qquad ^\top_t \quad _t.$$

To accelerate the computation of $_t^{-1}$, we take advantage of frequent directions to produce a low-rank approximation of $_t$ denoted by $_t$ $\mathbf{s}_1^t, \mathbf{s}_2^t, ..., \mathbf{s}_\tau^t$ $^\top \in \mathbb{R}^{\tau\times d}$. Then we can redefine the $_t$ in the proximal term as

$_t$ δ $+ (\quad ^\top_t \quad _t)^{1/2}$.

Let SVD of $_t$ be $_t$ $^\top$ where $\in \mathbb{R}^{\tau\times\tau}, \Sigma \in \mathbb{R}^{\tau\times\tau}$ and $\in \mathbb{R}^{\tau\times d}$, then we have $(\quad ^\top_t \quad _t)^{1/2}$ Σ $^\top$. By Woodbury Formula [Hager, 1989], we have

$$^{-1}_t \quad (\delta + \quad \Sigma \quad ^\top)^{-1} \quad \frac{1}{\delta}\big(- \quad (\delta + \Sigma)^{-1}\Sigma \quad ^T\big).$$

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 -th round our algorithm performs the following updates

$$\begin{aligned} &\mathbf{s}_\tau^{t-1} \quad \mathbf{g}_t, \bar{\mathbf{g}}_t \quad \bar{\mathbf{g}}_{t-1} + \mathbf{g}_t \\ &^\top \quad _{t-1}, \sigma_t \quad \Sigma_{tt}^2 \qquad \text{SVD} \\ &\Sigma' \quad \sqrt{\Sigma^2 - \sigma_t}, \quad _t \quad \Sigma' \quad ^\top \\ &\boldsymbol{\beta}_{t+1} \quad -\frac{\eta}{\delta}\big(\bar{\mathbf{g}}_t - \quad (\delta + \Sigma')^{-1}\Sigma' \quad ^\top\bar{\mathbf{g}}_t\big) \end{aligned} \qquad (6)$$

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 -th round our algorithm performs the following updates

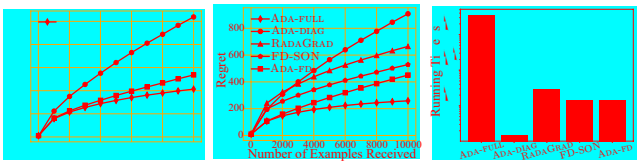
$$\begin{aligned} &\mathbf{s}_\tau^{t-1} \quad \mathbf{g}_t \\ &^\top \quad _{t-1}, \sigma_t \quad \Sigma_{tt}^2 \qquad \text{SVD} \\ &\Sigma' \quad \sqrt{\Sigma^2 - \sigma_t}, \quad _t \quad \Sigma' \quad ^\top \\ &\boldsymbol{\beta}_{t+1} \quad \boldsymbol{\beta}_t - \frac{\eta}{\delta}\big(\mathbf{g}_t - \quad (\delta + \Sigma')^{-1}\Sigma' \quad ^\top\mathbf{g}_t\big) \end{aligned} \qquad (7)$$

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 $_t,$. 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 φ / 0, the computational cost of $^{-1}_t$ can still be reduced dramatically, though the updating of $\boldsymbol{\beta}_t$ may not have closed-form solution.

3.2 Theoretical Guarantees

Compared with RADAGRAD, a significant advantage of our methods is they are equipped with formal theoretical guarantees similar to ADA



(a) Regret for PDS (b) Regret for CMD (c) Running Time

Figure 1: The comparison of regret among different algorithms on synthetic data for primal-dual subgradient (PDS) method and composite mirror descent (CMD) method, and the comparison of running time among different algorithms for composite mirror descent (CMD) method

show that ADA-FD approximates ADA-FULL well and outperforms ADA-DIAG. Second, we compare ADA-FD with RADAGRAD that is also an approximation of ADA-FULL and FD-SON [Luo *et al.*, 2016] that is an approximation of ONS by frequent directions.

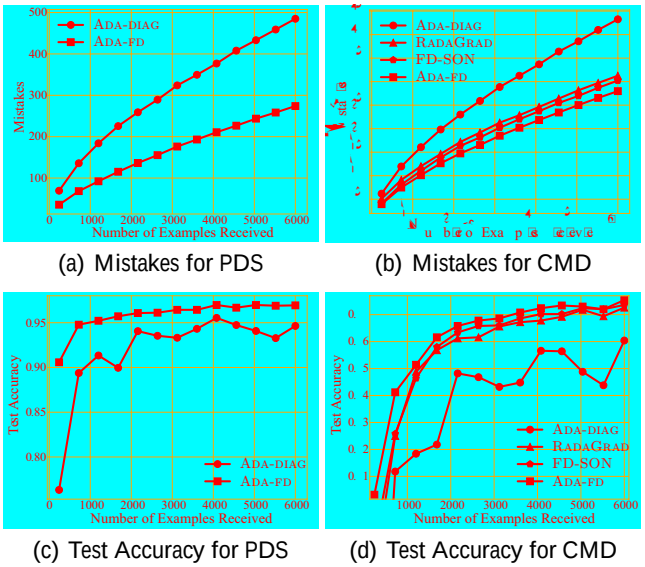
4.1 Online Regression

First, we consider the problem of online regression where $f_t(\boldsymbol{\beta})$ $|\boldsymbol{\beta}^\top \mathbf{x}_t -$ $_t|$ and $_t$ $\boldsymbol{\beta}_*^\top \mathbf{x}_t$ with ideal synthetic data. Specifically, we set T $10,000$, d 500 and $\boldsymbol{\beta}_*$ $\boldsymbol{\beta}_*/\|\boldsymbol{\beta}_*\|_2$ where each entry of $\boldsymbol{\beta}_*$ is drawn independently from $\mathcal{N}(0,1)$. In order to meet the requirement of approximately low-rank structure, each data point $\mathbf{x}_t$ is sampled independently from a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ where $\boldsymbol{\mu}$ $\mathbf{1}$ and Σ has rapidly decaying eigenvalues $\lambda_j(\Sigma)$ λ_0 $^{-\alpha}$ with α 2 and λ_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, RADAGRAD and FD-SON. Parameters η and δ are searched in $\{1e{-}4, 1e{-}3, \cdots, 100\}$, 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 RADAGRAD, 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 method where we set τ 10 for methods using matrix approximation. For both primal-dual subgradient method and composite mirror descent method, our ADA-FD outperforms ADA-DIAG and is significantly faster than ADA-FULL. For composite mirror descent method, our ADA-FD outperforms RADAGRAD and FD-SON. From the comparison of running time, we find that ADA-FD is faster than RADAGRAD 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 LIBSVM repository [Chang and Lin, 2011]: Gisette and Epsilon which are high-dimensional and dense. At each round, the learning algorithm receives a single example $(\mathbf{x}_t,$ $_t)$ and suffers the squared hinge loss $f_t(\boldsymbol{\beta})$ $\frac{1}{2}\big($ $\big(0, 1 -$ $_t\boldsymbol{\beta}^\top \mathbf{x}_t\big)\big)^2$. 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, RADAGRAD and FD-SON. We omit the result of ADA-FULL, because it is too slow. Similar as before, parameters η and δ are searched in $\{2e{-}4, 2e{-}3, \cdots, 20\}$, and we choose the best values for each algorithm. To reduce the computational cost, we set the sketching size τ 10 for Gisette and τ 40 for Epsilon.



(a) Mistakes for PDS (b) Mistakes for CMD

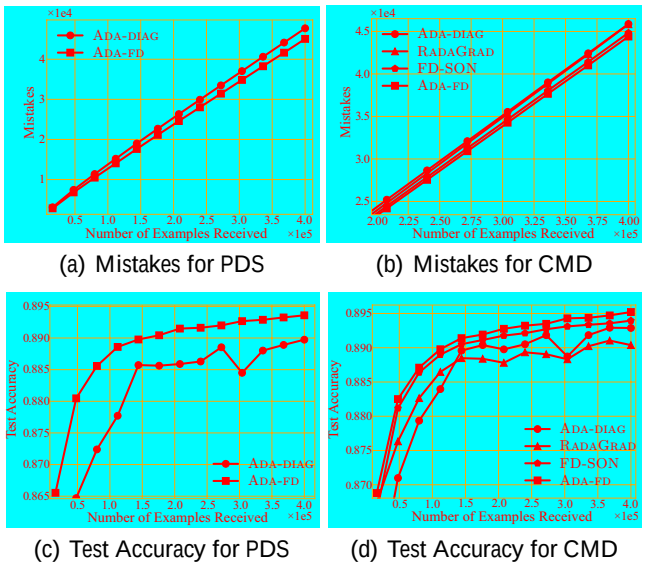
(c) Test Accuracy for PDS (d) Test Accuracy for CMD

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

Dataset	#Examples	#Features	#Classes
Gisette	6,000/1,000	5,000	2
Epsilon	400,000/100,000	2,000	2
MNIST	60,000/10,000	784	10
CIFAR10	50,000/10,000	3,072	10
SVHN	73,257/26,032	3,072	10

Table 1: Datasets used in experiments

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, RADAGRAD and FD-SON on both datasets. Note that RADAGRAD 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 RADAGRAD.



(a) Mistakes for PDS (b) Mistakes for CMD

(c) Test Accuracy for PDS (d) Test Accuracy for CMD

Figure 3: The comparison of mistakes and test accuracy among different algorithms on Epsilon for primal-dual subgradient (PDS) method and composite mirror descent (CMD) method

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 [1], which contains two 3×3 convolutional layers and one fully connected layer (12 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 2×2 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 η and δ of all algorithms are searched in $\{1e-3, 1e-2, \cdots, 1\}$, 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

References

[Chang and Lin, 2011] Chih-Chung Chang and Chih-Jen Lin. Libsvm : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011.

[Charikar *et al.*, 2004] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theoretical Computer Science*, 312(1):3–15, 2004.

[Chechik *et al.*, 2010] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11:1109–1135, 2010.

[Drineas *et al.*, 2006] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast monte carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006.

[Duchi *et al.*, 2010] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. In *Proceedings of the 23rd Annual Conference on Learning Theory*, pages 14–26, 2010.

[Duchi *et al.*, 2011] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[Freund and Schapire, 1999] Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.

[Ghashami and Phillips, 2014] Mina Ghashami and Jeff M. Phillips. Relative errors for deterministic low-rank matrix approximations. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 707–717, 2014.

[Ghashami *et al.*, 2016] Mina Ghashami, Jeff M. Phillips Edo Liberty, and David P. Woodruff. Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing*, 45(5):1762–1792, 2016.

[Hager, 1989] William W. Hager. Updating the inverse of a matrix. *SIAM review*, 31(2):221–239, 1989.

[Hazan *et al.*, 2007] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192, 2007.

[Jain *et al.*, 2008] Prateek Jain, Brian Kulis, Inderjit S. Dhillon, and Kristen Grauman. Online metric learning and fast similarity search. In *Advances in Neural Information Processing Systems 21*, pages 761–768, 2008.

[Kakade *et al.*, 2008] Sham M. Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Efficient bandit algorithms for online multiclass prediction. In *Proceedings of the 25th International Conference on Machine Learning*, pages 440–447, 2008.

[Kaski, 1998] Samuel Kaski. Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *Proceedings of the 1998 IEEE International Joint Conference on Neural Networks*, pages 413–418, 1998.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[Krummenacher *et al.*, 2016] Gabriel Krummenacher, Brian McWilliams, Yannic Kilcher, Joachim M. Buhmann, and Nicolai Meinshausen. Scalable adaptive stochastic optimization using random projections. In *Advances in Neural Information Processing Systems 29*, pages 1750–1758, 2016.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998.

[Liberty, 2013] Edo Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 581–588, 2013.

[Luo *et al.*, 2016] Haipeng Luo, Alekh Agarwal, Nicolo Cesa-Bianchi, and John Langford. Efficient second order online learning by sketching. In *Advances in Neural Information Processing Systems 29*, pages 902–910, 2016.

[Mairal *et al.*, 2010] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

[Misra and Gries, 1982] J. Misra and David Gries. Finding repeated elements. *Science of Computer Programming*, 2(2):143–152, 1982.

[Nalko *et al.*, 2011] N. Nalko, P. G. Martinsson, and J. A.