International Journal of Data Science and Analytics (2020) 9:389–400
https://doi.org/10.1007/s41060-019-00174-4

REGULAR PAPER

Accelerating adaptive online learning by matrix approximation

Yuanyu Wan[1] · **Lijun Zhang**[1]

Received: 27 November 2018 / Accepted: 14 January 2019 / Published online: 24 January 2019

Abstract
Adaptive subgradient methods are able to leverage the second-order information of functions to improve the regret and have become popular for online learning and optimization. According to the amount of information used, these methods can be divided into diagonal-matrix version (ADA-DIAG) and full-matrix version (ADA-FULL). In practice, ADA-DIAG is the most commonly adopted instead of ADA-FULL, because ADA-FULL is computationally intractable in high dimensions though it has smaller regret when gradients are correlated. In this paper, we propose to employ techniques of matrix approximation to accelerate ADA-FULL and develop two methods based on random projections. Compared with ADA-FULL, at each iteration, our methods reduce the space complexity from $O(d^2)$ to $O(\tau d)$ and the time complexity from $O(d^3)$ to $O(\tau^2 d)$ where d is the dimensionality of the data and $\tau \ll d$ is the number of random projections. Experimental results about online convex optimization and training convolutional neural networks show that our methods are comparable to ADA-FULL and outperform other state-of-the-art algorithms including ADA-DIAG.

Keywords Online learning · Adaptive methods · Matrix approximation · Random projection

1 Introduction

Online learning is a general framework for modeling sequential decision making [2,16,36,37] and has been widely used in many domains such as online routing [3], online pattern mining [15] and online detection [25,34]. Adaptive subgradient methods (ADAGRAD) are popular for online learning, which dynamically integrate knowledge of the geometry of data observed in earlier iterations to guide the direction of updating [7]. Different from the conventional online methods, ADAGRAD employ adaptive proximal functions to control the learning rate for each dimension, and the proximal functions are iteratively modified by the algorithm instead of tuning manually. There are two versions of adaptive subgradient methods: ADA-DIAG which uses a diagonal matrix to define the proximal function and ADA-FULL which uses a full matrix to define the proximal function. Because ADA-FULL is computationally intractable in high dimensions, ADA-DIAG is the most commonly studied and adopted version in practice.

However, compared with ADA-FULL, ADA-DIAG cannot capture the correlation in the gradients. As a result, the regret of ADA-DIAG may be worse than that of ADA-FULL when the high-dimensional data are dense and have a low-rank structure. This dilemma prompts a question as to whether we can design algorithms that possess the merits of two versions: i.e., the light computation of ADA-DIAG and the small regret of ADA-FULL. In a recent work [19], Krummenacher et al. presented two approximation algorithms to accelerate ADA-FULL, namely ADA-LR and RADAGRAD. Although ADA-LR is equipped with a regret bound, its space and time complexities are quadratic in the dimensionality d, which is unacceptable when d is large. In contrast, the space and time complexities of RADAGRAD are linear in d, but it lacks theoretical guarantees.

Along this line of research, this paper aims to attain theoretical guarantees and at the same time keeping the computations light. Note that ADA-FULL is computationally impractical mainly due to the fact it needs to maintain a matrix of gradient outer products and compute its square root

This paper is an extension version of the PAKDD'2018 Long Presentation paper "Accelerating Adaptive Online Learning by Matrix Approximation" [31].

✉ Lijun Zhang
zhanglj@lamda.nju.edu.cn

Yuanyu Wan
wanyy@lamda.nju.edu.cn

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

and inverse in each round. Actually, similar problems have been encountered in online Newton step (ONS) for exponentially concave functions [16]. Recently, Luo et al. proposed to accelerate ONS using matrix sketching methods including random projections [22]. Motivated by previous work, we first propose to employ random projections to construct a low-rank approximation of gradient outer products and manipulate this low-rank matrix in subsequent calculations. In this way, the new algorithm, named ADA-GP, reduces the space complexity from $O(d^2)$ to $O(\tau d)$ and the time complexity from $O(d^3)$ to $O(\tau^2 d)$, implying both the space and time complexities have a linear dependence on the dimensionality d.

ADA-GP achieves excellent empirical performance in our experiments. However, due to subtle independence issues, it is difficult to analyze ADA-GP theoretically. To circumvent this problem, we propose to replace the outer product matrix of gradients in ADA-FULL with the outer product matrix of data and then develop a similar method, named ADA-DP, that applies random projections to the outer product matrix of data. The space and time complexities of ADA-DP are similar to those of ADA-GP. Moreover, we present theoretical analysis for ADA-DP when the outer product matrix of data is low rank, and further extend to the full-rank case. In the experiments, we first examine the performance of our methods on online convex optimization, and the results demonstrate that they are highly comparable to ADA-FULL and are much more efficient. Furthermore, we conduct experiments on training convolutional neural networks (CNN) and show that ADA-GP outperforms ADA-DIAG and RADAGRAD.

2 Related work

ADAGRAD Adaptive subgradient methods use the second-order information to tune the step size of gradient descent adaptively [7]. For sparse data, the regret guarantee of ADAGRAD could be exponentially smaller in the dimension d than the non-adaptive regret bound. In the following, we provide a brief introduction of ADAGRAD. Note that the idea of ADAGRAD can be incorporated into either primal–dual subgradient method [33] or composite mirror descent [8]. For brevity, we take the composite mirror descent as an example.

In the t-round, the learner needs to determine an action $\boldsymbol{\beta}_t \in \mathbb{R}^d$ and then observes 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 $F_t(\boldsymbol{\beta}_t)$, and the goal is to minimize the accumulated loss over T iterations. Let $\nabla f_t(\boldsymbol{\beta})$ denote the subdifferential set of function f_t evaluated at $\boldsymbol{\beta}$ and $\mathbf{g}_t \in \nabla f_t(\boldsymbol{\beta}_t)$ be a particular vector in the subdifferential set. Define the outer product matrix of gradients $G_t = \sum_{i=1}^t \mathbf{g}_i \mathbf{g}_i^\top$. Then, we use the square root of G_t to construct a positive definite matrix H_t and have the following two choices:

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

where $\sigma > 0$ is a parameter and I_d is the identity matrix of size $d \times d$. The proximal term is given by $\Psi_t(\boldsymbol{\beta}) = \frac{1}{2}\langle \boldsymbol{\beta}, H_t \boldsymbol{\beta} \rangle$, and the associated Bregman divergence is

$$B_{\Psi_t}(\mathbf{x}, \mathbf{y}) = \Psi_t(\mathbf{x}) - \Psi_t(\mathbf{y}) - \frac{1}{2}\langle \nabla \Psi_t(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle.$$

In each iteration, the composite mirror descent method updates by

$$\begin{aligned} \boldsymbol{\beta}_{t+1} &= \operatorname*{argmin}_{\boldsymbol{\beta}} \left\{ \eta \langle \mathbf{g}_t, \boldsymbol{\beta} \rangle + \eta \varphi(\boldsymbol{\beta}) + B_{\Psi_t}(\boldsymbol{\beta}, \boldsymbol{\beta}_t) \right\} \\ &= \boldsymbol{\beta}_t - \eta H_t^{-1} \mathbf{g}_t, \quad \text{if } \varphi = 0 \end{aligned}$$

where $\eta > 0$ is a fixed step size. When the dimensionality d is large, ADA-FULL is impractical because the storage cost of G_t and the running time of finding its square root and inverse of H_t are unacceptable.

To make ADA-FULL scalable, Krummenacher et al. proposed two methods that approximate the proximal term $\Psi_t(\boldsymbol{\beta})$ [19]. Based on the fast randomized singular value decomposition (SVD) [26], they presented an algorithm ADA-LR that performs the following updates:

$$\begin{aligned} &G_t = G_{t-1} + \mathbf{g}_t \mathbf{g}_t^\top \\ &\widetilde{G}_t = G_t \Pi \qquad \text{Random Projection} \\ &QR = \widetilde{G}_t \qquad \text{QR-decomposition} \\ &B = Q^\top G_t \\ &U \Sigma V^\top = B \qquad \text{SVD} \\ &\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \eta V (\Sigma^{1/2} + \sigma I_\tau)^{-1} V^\top \mathbf{g}_t \end{aligned} \tag{1}$$

where $\Pi \in \mathbb{R}^{d \times \tau}$ is the random matrix of the subsampled randomized Fourier transform. We note that random projections are utilized in the second step to generate a smaller matrix $\widetilde{G}_t \in \mathbb{R}^{d \times \tau}$. It is easy to verify that the space and time complexities of ADA-LR are, respectively, $O(d^2)$ and $O(\tau d^2)$, which are still unacceptable when the d is large.

To further improve the efficiency, they presented algorithm RADAGRAD by introducing more randomized approximations, the space and time complexities of which are, respectively, $O(\tau d)$ and $O(\tau^2 d)$. Unfortunately, RADAGRAD is a heuristic method and lacks theoretical guarantees. Note that the basic ideas of ADA-LR and RADAGRAD are using random projection to construct an orthogonal matrix Q

such that $QQ^\top G_t \approx G_t$. However, $QQ^\top G_t$ is not an unbiased estimation of G_t. In contrast, our methods directly use random projection to construct an unbiased estimation of G_t or the outer product matrix of data. Furthermore, our methods are very efficient in the sense that their computational complexities are linear in the dimensionality d, and ADA-DP is equipped with theoretical guarantees. In contrast, although RADAGRAD has a similar computational cost, it does not have theoretical justifications.

As previously mentioned, in [22], Luo et al. adopted matrix sketching methods to accelerate ONS that also encounters the similar problems as ADA-FULL. Specifically, their ONS updates by

$$A_t = \alpha I_d + \sum_{i=1}^{t} \eta_i \mathbf{g}_i \mathbf{g}_i^\top \text{ and } \boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - A_t^{-1}\mathbf{g}_t$$

where $\alpha > 0$ and $\eta_i = O(1/\sqrt{i})$ for general convex functions. We can reformulate this update rule as

$$H_t = \sigma I_d + \sum_{i=1}^{t} \frac{1}{\sqrt{i}} \mathbf{g}_i \mathbf{g}_i^\top \text{ and } \boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \eta H_t^{-1}\mathbf{g}_t.$$

To accelerate ONS, they use matrix sketching methods to calculate a low-rank approximation of $\sum_{i=1}^{t} \frac{1}{\sqrt{i}} \mathbf{g}_i \mathbf{g}_i^\top$. Motivated by Luo et al. [22], our work employs random projections to calculate a low-rank approximation of full matrix.

However, there are obviously differences between our work and this related work. First, our ADA-GP uses random projection to approximate $\sum_{i=1}^{t} \mathbf{g}_i \mathbf{g}_i^\top$ rather than $\sum_{i=1}^{t} \frac{1}{\sqrt{i}} \mathbf{g}_i \mathbf{g}_i^\top$ approximated by RP-SON [22]. Although these full matrices are similar, the latter could destroy the data-dependent property because of the corresponding $O(\sqrt{Td})$ regret bound. Moreover, we propose to use the outer product matrix of data to replace the outer product matrix of gradients $\sum_{i=1}^{t} \mathbf{g}_i \mathbf{g}_i^\top$ which leads to ADA-DP. Note that this simple change can avoid the dependence issue that the gradient $\mathbf{g}_t$ depends on the random vectors. Second, the theoretical analysis in our work is obviously different from Luo et al. [22]. The only common part is the property of the random projections for low-rank data. But we further exploit the property of the random projection for full-rank data. Third, our methods and this related work are designed for different tasks. Our paper aims to accelerate ADA-FULL, and this related work aims to accelerate ONS. Note that ADA-FULL is a data-dependent algorithm for general convex function and ONS is proposed to derive a logarithmic regret for exponentially concave functions.

Random projection Random projection [17,23,32] is a simple yet powerful method for dimensionality reduction. For a data point $\mathbf{x} \in \mathbb{R}^n$, random projection directly reduces its dimensionality to τ by $R^\top \mathbf{x}$, where $R \in \mathbb{R}^{n\times\tau}$ is a random matrix. For any two different points, random projection can approximately preserve their distance after reducing their dimensionality with high probability. It has been successfully applied to many machine learning tasks including classification [10,28], regression [24], clustering [4,9], manifold learning [6,11] and optimization [12,35]. Random projection can be implemented in various different ways with the corresponding random matrices [1,21], and the most classical one is the Gaussian random projection, where each entry of R is sampled from a Gaussian distribution. In this paper, we focus on Gaussian random projection due to its nice theoretical properties and easy implementations.

3 Adaptive methods with random projection

In this section, we introduce our proposed methods and theoretical results, the proofs of which have been deferred to "Appendix A".

3.1 Problem setting

To facilitate presentations, we consider the case $\varphi = 0$, and our methods can be directly extended to the general case $\varphi \neq 0$. The goal of the learner is to minimize the regret, 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 a fixed optimal predictor.

Besides that, let us introduce the necessary notations used in our paper. We use $\mathbb{E}[\cdot]$ to denote expectation and I_d to denote the identity matrix of size $d \times d$. For any matrix X, we use $\|X\|$ to denote its spectral norm, $\lambda_i(X)$ to denote its i-th largest eigenvalue and $\mathrm{tr}(X)$ to denote its trace. For any vector $\mathbf{x}$, we use $\|\mathbf{x}\|_2$ to denote its Euclidean ℓ_2 norm. More general, with proximal term $\Psi(\mathbf{x}) = \frac{1}{2}\langle \mathbf{x}, H\mathbf{x}\rangle$ defined by a positive definite matrix H, we use $\|\mathbf{x}\|_{\Psi^*} = \sqrt{\langle \mathbf{x}, H^{-1}\mathbf{x}\rangle}$ to denote the associated dual norm. These notations are summarized in Table 1.

3.2 The proposed ADA-GP method

From previous discussions, we know that if one can find a low-rank matrix to approximate G_t, then both space and time complexities of ADA-FULL can be reduced dramatically. Random projections provide an elegant way for low-rank matrix approximations, as explained below.

Define

$$A_t^\top = [\mathbf{g}_1, \ldots, \mathbf{g}_t] \in \mathbb{R}^{d\times t}, \quad R_t = [\mathbf{r}_1, \ldots, \mathbf{r}_t] \in \mathbb{R}^{\tau\times t}$$

Table 1 Summary of notations

Notation	Meaning
$\mathbb{E}[\cdot]$	Expectation
I_d	Identity matrix of size $d \times d$
$\|X\|$	Spectral norm of matrix X
$\lambda_i(X)$	i-th largest eigenvalue of matrix X
$\mathrm{tr}(X)$	Trace of matrix X
$\|\mathbf{x}\|_2$	Euclidean ℓ_2 norm of $\mathbf{x}$
$\|\mathbf{x}\|_{\Psi^*}$	Dual norm of $\mathbf{x}$ with respect to Ψ

where the i-th column of $A_t^\top$ is gradient $\mathbf{g}_i$, and each entry of R_t is a Gaussian random variable drawn from $\mathcal{N}(0, 1/\tau)$ independently. Then, we have

$$G_t = A_t^\top A_t, \ \mathbb{E}[R_t^\top R_t] = I_d.$$

To accelerate the computation, we define

$$S_t = R_t A_t = \sum_{i=1}^{t} \mathbf{r}_i \mathbf{g}_i^\top \in \mathbb{R}^{\tau \times d}.$$

Note that S_t can be calculated on the fly as $S_t = S_{t-1} + \mathbf{r}_t \mathbf{g}_t^\top$. When τ is large enough, we expect $R_t^\top R_t \approx I_d$, implying

$$S_t^\top S_t = A_t^\top R_t^\top R_t A_t \approx A_t^\top A_t = G_t.$$

Thus, $S_t^\top S_t$ could be used as a low-rank approximation of G_t. The matrix H_t in the proximal term can be redefined as

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

Let SVD of S_t be $S_t = U \Sigma V^\top$, then we have $S_t^\top S_t = V \Sigma^2 V^\top$ and $H_t = \sigma I_d + V \Sigma V^\top$. According to Woodbury formula [14], we have

$$\begin{aligned} H_t^{-1} &= (\sigma I_d + V \Sigma V^\top)^{-1} \\ &= \frac{1}{\sigma}\big(I_d - V(\sigma I_\tau + \Sigma)^{-1} \Sigma V^T\big). \end{aligned}$$

As a result, in the t-th round, our algorithm performs the following updates

$$\begin{aligned} &S_t = S_{t-1} + \mathbf{r}_t \mathbf{g}_t^\top \quad \text{Random Projection} \\ &U \Sigma V^\top = S_t \qquad \text{SVD} \\ &\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \frac{\eta}{\sigma}\left(\mathbf{g}_t - V(\sigma I_\tau + \Sigma)^{-1} \Sigma V^\top \mathbf{g}_t\right) \end{aligned} \tag{2}$$

The detailed procedure is summarized in Algorithm 1 and named as adaptive online learning with gradient projection (ADA-GP).

Algorithm 1 ADA-GP

Input: $\eta > 0, \sigma > 0, \tau, S_0 = \mathbf{0}_{\tau \times d}, \boldsymbol{\beta}_1 = \mathbf{0}$;
for $t = 1, \ldots, T$ **do**
  Receive $\mathbf{g}_t = \nabla f_t(\beta_t)$
  $S_t = S_{t-1} + \mathbf{r}_t \mathbf{g}_t^\top$ {Random Projections}
  $U \Sigma V^\top = S_t$ {SVD}
  $\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \frac{\eta}{\sigma}\left(\mathbf{g}_t - V(\sigma I_\tau + \Sigma)^{-1} \Sigma V^\top \mathbf{g}_t\right)$
end for

Remark First, it is easy to verify the time and space complexities of our ADA-GP are $O(\tau^2 d)$ and $O(\tau d)$, respectively. Thus, both of them are linear in the dimensionality d. Second, comparing (2) with (1), we observe that our updating rules are much more simple than those of ADA-LR. Note that the RADAGRAD algorithm of [19] is even more complicated than ADA-LR. Third, besides random projection, we also note that there exist other ways for low-rank matrix approximations, such as matrix sketching [32]. After the conference version of this work, we have further proposed to utilize a deterministic matrix sketching techniques named as frequent directions [13] to approximate ADA-FULL [30].

3.3 The proposed ADA-DP method

Although ADA-GP performs very well in our experiments, it is difficult to establish a regret bound due to dependence issues. To be specific, the gradient $\mathbf{g}_t$ depends on the random vectors $[\mathbf{r}_1, \ldots, \mathbf{r}_{t-1}]$, and as a result, standard concentration inequalities cannot be directly applied [29].

To avoid the aforementioned problem, we propose a strategy to get ride of the dependence issues and the new algorithm is equipped with theoretical guarantees. We consider the case $f_t(\boldsymbol{\beta}_t) = l(\boldsymbol{\beta}_t^\top \mathbf{x}_t)$ where $\mathbf{x}_t$ is a data vector. Then, we assume the data points $\mathbf{x}_1, \ldots, \mathbf{x}_t$ are independent from our algorithm. The key idea is to replace the outer product matrix of gradients G_t with the outer product matrix of data $X_t = \sum_{i=1}^{t} \mathbf{x}_i \mathbf{x}_i^\top$. Accordingly, H_t will be defined as $\sigma I_d + X_t^{1/2}$. To accelerate computations, our problem becomes finding a low-rank approximation of X_t.

Let $C_t^\top = [\mathbf{x}_1, \ldots, \mathbf{x}_t] \in \mathbb{R}^{d \times t}$, where each column is a data vector. Similar to ADA-GP, we define

$$S_t = R_t C_t = \sum_{i=1}^{t} \mathbf{r}_i \mathbf{x}_i^\top \in \mathbb{R}^{\tau \times d}$$

where $R_t \in \mathbb{R}^{\tau \times t}$ is the Gaussian random matrix. In this case, since R_t is independent of C_t, we have

$$\mathbb{E}[S_t^\top S_t] = C_t^\top \mathbb{E}[R_t^\top R_t] C_t = C_t^\top C_t = X_t$$

which means $S_t^\top S_t$ is an unbiased estimation of X_t.

The rest steps are similar to that of ADA-GP. The detailed procedure is summarized in Algorithm 2, named as adaptive

Algorithm 2 ADA-DP

Input: $\eta > 0, \sigma > 0, \tau$, $S_0 = \mathbf{0}_{\tau\times d}, \boldsymbol{\beta}_1 = \mathbf{0}$;
for $t = 1, \ldots, T$ **do**
  Receive $\mathbf{x}_t$ and $\mathbf{g}_t = \nabla f_t(\beta_t) = l'(\boldsymbol{\beta}_t^\top \mathbf{x}_t)\mathbf{x}_t$
  $S_t = S_{t-1} + \mathbf{r}_t \mathbf{x}_t^\top$ {Random Projections}
  $U\Sigma V^\top = S_t$ {SVD}
  $\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \frac{\eta}{\sigma}\left(\mathbf{g}_t - V(\sigma I_\tau + \Sigma)^{-1}\Sigma V^\top \mathbf{g}_t\right)$
end for

online learning with data projection (ADA-DP). It is obvious that the computation cost of ADA-DP is almost the same as that of ADA-GP. Thus, both the space and time complexities of ADA-DP are linear in d.

The main advantage of ADA-DP is that it has formal theoretical guarantees. We first consider the case that the data matrix C_T is low rank, and have the following theorem regarding the regret of Algorithm 2.

Theorem 1 *Let $r \ll d$ be the rank of C_T, and $0 < \delta < 1$ be the confidence parameter. Assume each entry of $\mathbf{r}_t \in \mathbb{R}^\tau$ is a Gaussian random variable drawn from $\mathcal{N}(0, 1/\tau)$ independently, $\tau = \Omega(\frac{r+\log(T/\delta)}{\epsilon^2})$ and $\sigma > 0$; then, ADA-DP ensures*

$$R(T) \le \frac{\sigma}{2\eta}\|\boldsymbol{\beta}_*\|_2^2 + \frac{1}{2\eta}\max_{t\le T}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_t\|_2^2 \operatorname{tr}(X_T^{1/2}) + \frac{2\eta}{\sqrt{1-\epsilon}}\max_{t\le T} l'(\boldsymbol{\beta}_t^\top \mathbf{x}_t)^2 \operatorname{tr}(X_T^{1/2}) + \frac{\epsilon}{2\eta}\max_{t\le T}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_t\|_2^2 \sum_{t=1}^{T}\|X_t^{1/2}\|$$

with probability at least $1-\delta$.

Remark Theorem 1 means that we can set the number of random projections as $\tau = \widehat{\Omega}(r)$ when the data matrix is low rank.

When the data matrix is full rank, Theorem 1 is inappropriate because it implies the number of random projections is on the order of the dimensionality. For the full-rank case, we further establish the following theorem to bound the regret of Algorithm 2.

Theorem 2 *Let $c \ge 1/32$, $\sigma \ge \sqrt{\alpha} > 0$, $\sigma_{ti}^2 = \lambda_i(X_t)$, $\tilde{r}_t = \sum_i \frac{\sigma_{ti}^2}{\alpha+\sigma_{ti}^2}$, $\tilde{r}_* = \max_{1\le t\le T}\tilde{r}_t$, $\sigma_{*1}^2 = \max_{1\le t\le T}\sigma_{t1}^2$, and $0 < \delta < 1$. Assume each entry of $\mathbf{r}_t \in \mathbb{R}^\tau$ is an independent random Gaussian variable drawn from $\mathcal{N}(0, 1/\tau)$, $\tau \ge \frac{\tilde{r}_*\sigma_{*1}^2}{c\epsilon^2(\alpha+\sigma_{*1}^2)}\log\frac{2dT}{\delta}$ and then ADA-DP ensures*

$$R(T) \le \frac{\sigma}{2\eta}\|\boldsymbol{\beta}_*\|_2^2 + \frac{1}{2\eta}\max_{t\le T}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_t\|_2^2 \operatorname{tr}(X_T^{1/2}) + \frac{2\eta}{\sqrt{1-\epsilon}}\max_{t\le T} l'(\boldsymbol{\beta}_t^\top \mathbf{x}_t)^2 \operatorname{tr}(X_T^{1/2}) + \frac{\epsilon}{2\eta}\max_{t\le T}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_t\|_2^2 \sum_{t=1}^{T}\|X_t^{1/2}\| + \frac{\sqrt{\epsilon\alpha}T}{\eta}\max_{t\le T}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_t\|_2^2.$$

with probability at least $1-\delta$.

Remark Following [35], we introduce the quantity $\tilde{r}_t$ to measure the effective rank of the data matrix C_t. When the eigenvalues of $C_t^\top C_t$ decrease rapidly, $\tilde{r}_t$ could be significantly smaller than d, even when C_t is full rank. Compared with Theorem 1, the upper bound in this theorem contains an additional term caused by the approximation error of full-rank matrices. Note that Theorem 2 means that we can set the number of random projections as $\tau = \widehat{\Omega}(\max_t \tilde{r}_t)$ when the data matrix has low effective rank.

Note that our methods and theories can be extended to the general case $\varphi \neq 0$. We just need to replace the updating rule as

$$\boldsymbol{\beta}_{t+1} = \operatorname*{argmin}_{\boldsymbol{\beta}}\left\{\eta\langle \mathbf{g}_t, \boldsymbol{\beta}\rangle + \eta\varphi(\boldsymbol{\beta}) + B_{\Psi_t}(\boldsymbol{\beta}, \boldsymbol{\beta}_t)\right\}.$$

Although the updating of $\boldsymbol{\beta}_{t+1}$ may not have closed-form solution, the computational cost of H_t^{-1} can still be reduced dramatically. The regret bound remains on the same order.

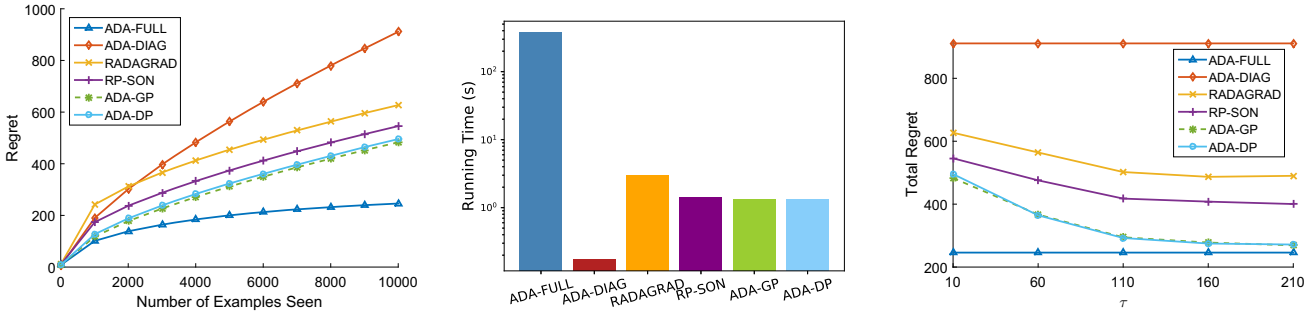
4 Experiments

In this section, we conduct numerical experiments to demonstrate the efficiency and effectiveness of our methods.

4.1 Online convex optimization

First, we compare our two methods against ADA-FULL, ADA-DIAG, RADAGRAD [19] and RP-SON [22] on a synthetic data, which is approximately low rank. Let $\boldsymbol{\beta}_* = \hat{\boldsymbol{\beta}}_*/\|\hat{\boldsymbol{\beta}}_*\|_2$ where each entry of $\hat{\boldsymbol{\beta}}_*$ is drawn independently from $\mathcal{N}(0, 1)$. We consider the problem of online regression where $f_t(\boldsymbol{\beta}) = |\boldsymbol{\beta}^\top \mathbf{x}_t - y_t|$ and $y_t = \boldsymbol{\beta}_*^\top \mathbf{x}_t$. We generate a regression dataset with $T = 10{,}000$ and $d = 500$. In order to meet the requirement of low rankness, 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 $\lambda_j(\Sigma) = \lambda_0 j^{-\alpha}$ with $\alpha = 2$ and $\lambda_0 = 100$.

For each algorithm, the parameters η and σ are searched in $\{1e{-}4, 1e{-}3, \ldots, 100\}$, and we choose the best values. We generate 5 random permutations for the synthetic data and report average results of all the algorithms. Figure 1a and b shows shows the regret and running time of different algorithms where we set $\tau = 10$ for methods using random



(a) Regret for $\tau = 10$ **(b)** Running time for $\tau = 10$ **(c)** Total regret for different τ

Fig. 1 Experimental results for online regression on the synthetic data

Table 2 Datasets used in experiments

Dataset	#Examples	#Features	#Classes
Gisette	6000/1000	5000	2
SensIT vehicle	78,823/19,705	100	3
Usps	7291/2007	256	10
MNIST	60,000/10,000	784	10
CIFAR10	50,000/10,000	3072	10
SVHN	73,257/26,032	3072	10

projections. The regret of our two methods is very close and better than ADA-DIAG, RADAGRAD and RP-SON, which indicates our methods approximate ADA-FULL very well. From the comparison of running time, we find that our two methods are obviously faster than ADA-FULL. Figure 1c further shows the total regret of all the algorithms when τ is set to different values. Note that ADA-FULL and ADA-DIAG are not affected by τ. We find that the total regret of methods using random projections is decreased with the increase of τ, and the regret of our two methods is much closer to that of ADA-FULL than other methods.

Second, following [7], we perform online classification to evaluate the performance of our methods. In each round, the learning algorithm receives a single example and ends with a single pass through the training data. There are two metrics to measure the performance: the online mistakes and the offline accuracy on the testing data. In the experiments, we use cross-entropy cost function as the loss.

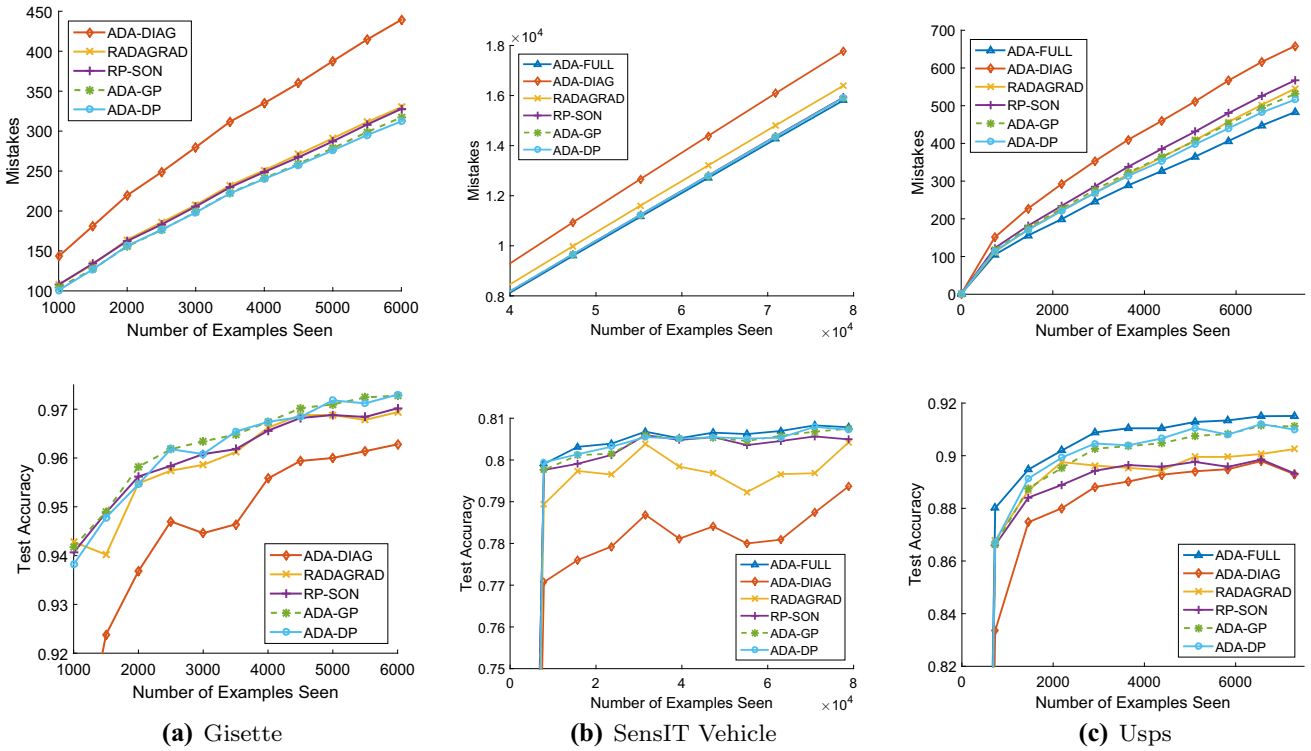
We conduct numerical experiments on three real-world datasets from LIBSVM repository [5]. Table 2 includes the description of these datasets. The parameters η and σ are searched in $\{1e{-}4, 1e{-}3, \ldots, 10\}$, and we choose the best values for each algorithm. To reduce the computational cost, we set the number of projections $\tau \leq \sqrt{d}$ for each dataset. Specifically, we set $\tau = 10$ for both SensIT Vehicle and Gisette datasets and $\tau = 15$ for Usps dataset. We omit the result of ADA-FULL on the Gisette dataset, because it is too slow.

We divide all the datasets into training part and testing part, and the numbers of training and testing examples are shown in Table 2. For training data, we generate 5 random permutations and report the average result. Figure 2 shows the comparison of test accuracy and mistakes among different algorithms. In addition, Table 3 presents the averaged test accuracy (%) with the standard deviation (%) after a single pass through training data for different methods. From Fig. 2 and Table 3, we have the following observations.

- Ignoring the computation issue, ADA-FULL achieves the highest test accuracy and the lowest mistakes after a single pass through training data.
- The performance of ADA-DIAG is much worse than all the other methods, which means only keeping a diagonal matrix is insufficient to capture the second-order information.
- Our ADA-GP and ADA-DP are close to ADA-FULL, which indicates that random projections cause little adverse affect on the performance.
- Our two methods are better than RADAGRAD in almost all the comparisons, which is due to the unbiased estimation used in our methods. Note that the unbiased estimation makes our methods to be better approximations of ADA-FULL than RADAGRAD.
- Our two methods are also better than RP-SON in almost all the comparisons, which verifies that the full matrix approximated by RP-SON could destroy the data-dependent property.

4.2 Non-convex optimization in CNN

Recently, ADA-DIAG becomes popular for non-convex optimization such as training neural networks. And in [19], Krummenacher et al. also show that RADAGRAD performs well for training neural networks. Therefore, taking training convolutional neural networks (CNN) as an example, we verify that our method outperforms ADA-DIAG and RADAGRAD. Because the convolutional layer does not meet



(a) Gisette **(b)** SensIT Vehicle **(c)** Usps

Fig. 2 The comparison of mistakes (top row) and test accuracy (bottom row) for online classification

Table 3 Test accuracy (%) after a single pass through training data

Dataset	ADA-FULL	ADA-DIAG	RADAGRAD	RP-SON	ADA-GP	ADA-DP
Gisette		96.28 ± 0.47	96.94 ± 0.08	97.02 ± 0.21	97.28 ± 0.17	97.30 ± 0.32
SensIT vehicle	80.78 ± 0.09	79.37 ± 0.21	80.42 ± 0.43	80.49 ± 0.12	80.75 ± 0.13	80.72 ± 0.06
Usps	91.51 ± 0.20	89.30 ± 0.92	90.25 ± 0.56	89.33 ± 0.55	91.11 ± 0.22	90.99 ± 0.18

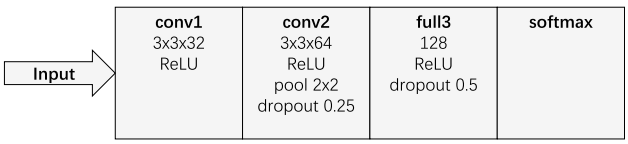


Fig. 3 The 4-layer CNN architecture used in our experiment

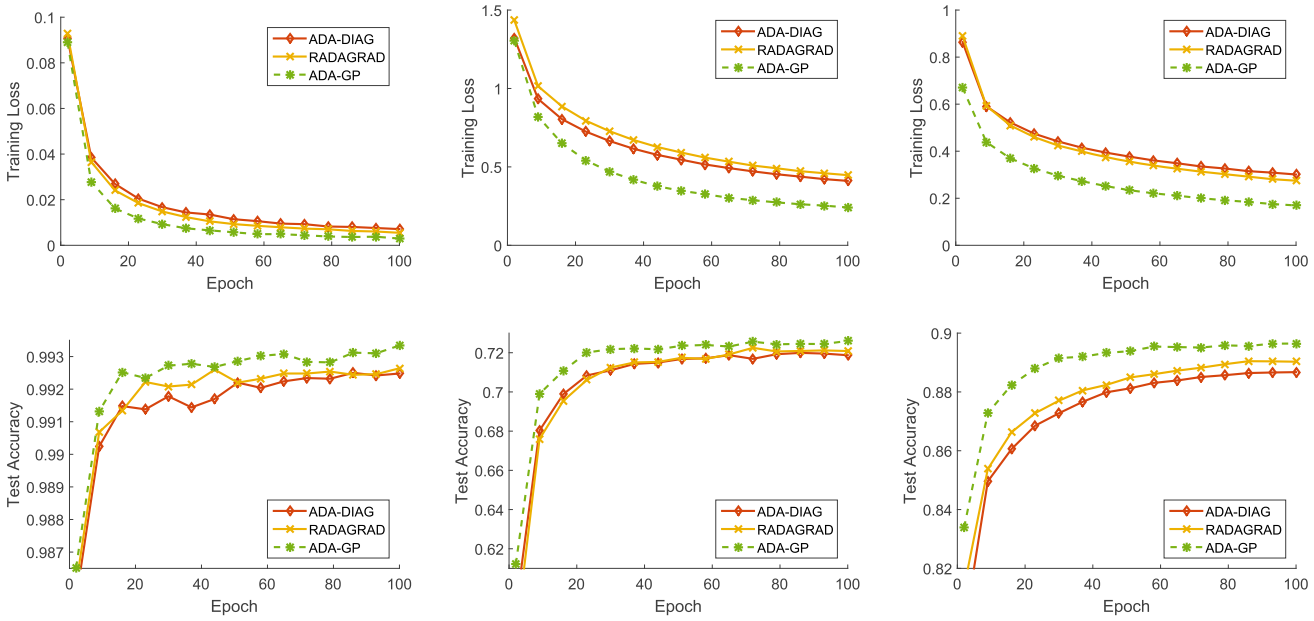
the case $f_t(\boldsymbol{\beta}_t) = l(\boldsymbol{\beta}_t^\top \mathbf{x}_t)$, we only perform ADA-GP on training CNN. We use the simple and standard architecture, which is chosen from Keras examples directory [1] and shown in Fig. 3, to perform classification on the MNIST [20], CIFAR10 [18] and SVHN [27] datasets.

Parameters η of all algorithms and σ of ADA-GP and RADAGRAD are searched in $\{1e{-}4, 1e{-}3, \ldots, 1\}$. For ADA-DIAG, σ is set to $1e{-}8$ as it is typically recommended. We choose the best values for each algorithm. Following as [19], we only consider applying ADA-GP and RADAGRAD to the convolutional layer, and other layers are still trained with ADA-DIAG 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 where we set $\tau = 20$. We find that ADA-GP can obviously improve the performance of ADA-DIAG on all datasets, and note that RADAGRAD is outperformed by ADA-DIAG in terms of training loss on CIFAR10. This result shows that ADA-GP is a better approximation of ADA-FULL than RADAGRAD.

5 Conclusions and future work

In this paper, we present ADA-GP and ADA-DP to approximate ADA-FULL using random projections. The time and space complexities of both algorithms are linear in the dimensionality d, and thus they are able to accelerate the

[1] https://github.com/keras-team/keras/blob/master/examples/mnist_cnn.py.



(a) MNIST **(b)** CIFAR10 **(c)** SVHN

Fig. 4 The comparison of training loss (top row) and test accuracy (bottom row) for training CNN

computation significantly. Furthermore, according to our theoretical analysis, the number of random projections in ADA-DP is on the order of the low rank or low effective rank. Numerical experiments on online convex optimization show that our methods outperform ADA-DIAG, RADAGRAD and RP-SON and are close to ADA-FULL. And experiments on training CNN show that ADA-GP outperforms ADA-DIAG and RADAGRAD.

One limitation of this work is that only the proposed ADA-DP has theoretical guarantees for the case $f_t(\boldsymbol{\beta}) = l(\boldsymbol{\beta}^\top \mathbf{x}_t)$. In the future, we will investigate how to extend our theoretical results for more general case.

Acknowledgements This work was partially supported by the National Key R&D Program of China (2018YFB1004300), NSFC-NRF Joint Research Project (61861146001) and YESS (2017QNRC001).

A Theoretical analysis

In this section, we provide proofs of Theorems 1 and 2.

A.1 Supporting results

The following results are used throughout our analysis.

Lemma 1 (Proposition 3 of [7]). *Let sequence* $\{\boldsymbol{\beta}_t\}$ *be generated by ADA-DP. We have*

$$R(T) \le \frac{1}{\eta}\sum_{t=1}^{T-1}\left[B_{\Psi_{t+1}}(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{t+1}) - B_{\Psi_t}(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{t+1})\right] + \frac{1}{\eta}B_{\Psi_1}(\boldsymbol{\beta}^*, \boldsymbol{\beta}_1) + \frac{\eta}{2}\sum_{t=1}^{T}\|f_t'(\boldsymbol{\beta}_t)\|^2_{\Psi_t^*}.$$

Lemma 2 *Let* $X_t = \sum_{i=1}^t \mathbf{x}_i\mathbf{x}_i^\top$ *and* $A^\dagger$ *denote the pseudo-inverse of* A*, then*

$$\sum_{t=1}^{T}\left\langle \mathbf{x}_t, (X_t^{1/2})^\dagger \mathbf{x}_t\right\rangle \le 2\sum_{t=1}^{T}\left\langle \mathbf{x}_t, (X_T^{1/2})^\dagger \mathbf{x}_t\right\rangle = 2\,\mathrm{tr}(X_T^{1/2}).$$

Lemma 2 can be proved in the same way as Lemma 10 of [7].

Theorem 3 (Theorem 2.3 of [32]). *Let* $0 < \epsilon, \delta < 1$ *and* $S = \frac{1}{\sqrt{k}}R \in \mathbb{R}^{k\times n}$ *where the entries of* R *are independent standard normal random variables. Then if* $k = \Theta(\frac{d+\log(1/\delta)}{\epsilon^2})$*, then for any fixed* $n \times d$ *matrix* A*, with probability* $1-\delta$*, simultaneously for all* $\mathbf{x} \in \mathbb{R}^d$*,*

$$(1-\epsilon)\|A\mathbf{x}\|_2^2 \le \|SA\mathbf{x}\|_2^2 \le (1+\epsilon)\|A\mathbf{x}\|_2^2.$$

Based on the above theorem, we derive the following corollary.

Corollary 1 *Let* $0 < \epsilon, \delta < 1$ *and each entry of* $\mathbf{r}_t \in \mathbb{R}^\tau$ *is a Gaussian random variable independently drawn from*

$\mathcal{N}(0, 1/\tau)$. *Then, if* $\tau = \Omega(\frac{r+\log(T/\delta)}{\epsilon^2})$, *with probability* $1-\delta$, *simultaneously for all* $t = 1, \ldots, T$,

$$(1-\epsilon)C_t^\top C_t \preceq S_t^\top S_t \preceq (1+\epsilon)C_t^\top C_t.$$

Theorem 4 (Theorem 10 of [35]). *Let* $C = \mathrm{diag}(c_1, \ldots, c_p)$ *and* $S = \mathrm{diag}(s_1, \ldots, s_p)$ *be* $p \times p$ *diagonal matrices, where* $c_i \neq 0$ *and* $c_i^2 + s_i^2 = 1$ *for all* i. *Let* $R \in \mathbb{R}^{p\times n}$ *be a Gaussian random matrix. Let* $M = C^2 + \frac{1}{n}SRR^\top S$ *and* $r = \sum_i s_i^2$.

$$\Pr(\lambda_1(M) \geq 1+t) \leq q \cdot \exp\left(-\frac{cnt^2}{\max_i(s_i^2)r}\right),$$
$$\Pr(\lambda_p(M) \leq 1-t) \leq q \cdot \exp\left(-\frac{cnt^2}{\max_i(s_i^2)r}\right),$$

where the constant c *is at least* $1/32$, *and* q *is the rank of* S.

Based on the above theorem, we derive the following corollary.

Corollary 2 *Let* $c \geq 1/32$, $\alpha > 0$, $\sigma_{ti}^2 = \lambda_i(C_t^\top C_t)$, $\tilde{r}_t = \sum_i \frac{\sigma_{ti}^2}{\alpha+\sigma_{ti}^2}$, $\tilde{r}_* = \max_{k\leq t\leq T} \tilde{r}_t$ *and* $\sigma_{*1}^2 = \max_{1\leq t\leq T} \sigma_{t1}^2$. *Let* $K_t = \alpha I_d + C_t^\top C_t$, $\tilde{K}_t = \alpha I_d + S_t^\top S_t$ *and* $\tilde{I}_t = K_t^{-1/2}\tilde{K}_t K_t^{-1/2}$. *If* $\tau \geq \frac{\tilde{r}_*\sigma_{*1}^2}{c\epsilon^2(\alpha+\sigma_{*1}^2)} \log\frac{2dT}{\delta}$, *with probability at least* $1-\delta$, *simultaneously for all* $t = 1, \ldots, T$,

$$(1-\epsilon)I_d \preceq \tilde{I}_t \preceq (1+\epsilon)I_d.$$

A.2 Proof of Theorem 1

Let $\widetilde{X}_t$ denote $S_t^\top S_t$. First, we consider bounding the first term in the upper bound of Lemma 1. With probability $1-\delta$, we have

$$\begin{aligned}
&B_{\Psi_{t+1}}(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{t+1}) - B_{\Psi_t}(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{t+1}) \\
&= \frac{1}{2}\left\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}, (\widetilde{X}_{t+1}^{1/2} - \widetilde{X}_t^{1/2})(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\right\rangle \\
&\leq \frac{1}{2}\left\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}, \sqrt{1+\epsilon}X_{t+1}^{1/2}(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\right\rangle \\
&\quad - \frac{1}{2}\left\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}, \sqrt{1-\epsilon}X_t^{1/2}(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\right\rangle \\
&\leq \frac{1}{2}\left\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}, (X_{t+1}^{1/2} - X_t^{1/2})(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\right\rangle \\
&\quad + \frac{\epsilon}{4}\left\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}, (X_{t+1}^{1/2} + X_t^{1/2})(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\right\rangle \\
&\leq \frac{1}{2}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}\|_2^2 \|(X_{t+1}^{1/2} - X_t^{1/2})\| \\
&\quad + \frac{\epsilon}{4}\left\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}, (X_{t+1}^{1/2} + X_t^{1/2})(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\right\rangle \\
&\leq \frac{1}{2}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}\|_2^2 \mathrm{tr}(X_{t+1}^{1/2} - X_t^{1/2}) \\
&\quad + \frac{\epsilon}{4}\left\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}, (X_{t+1}^{1/2} + X_t^{1/2})(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\right\rangle
\end{aligned}$$

where the first inequality is due to Corollary 1.

Thus, we can get

$$\begin{aligned}
&\sum_{t=1}^{T-1}\left[B_{\Psi_{t+1}}(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{t+1}) - B_{\Psi_t}(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{t+1})\right] \\
&\leq \frac{1}{2}\sum_{t=1}^{T-1}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}\|_2^2 \mathrm{tr}(X_{t+1}^{1/2} - X_t^{1/2}) \\
&\quad + \frac{\epsilon}{4}\sum_{t=1}^{T-1}\left\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}, (X_{t+1}^{1/2} + X_t^{1/2})(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\right\rangle \\
&\leq \frac{1}{2}\max_{t\leq T}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_t\|_2^2 \mathrm{tr}(X_T^{1/2}) - \frac{1}{2}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_1\|_2^2 \mathrm{tr}(X_1^{1/2}) \\
&\quad + \frac{\epsilon}{2}\max_{t\leq T}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_t\|_2^2 \sum_{t=1}^{T}\|X_t^{1/2}\| \\
&\quad - \frac{\epsilon}{4}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_1\|_2^2 \mathrm{tr}(X_1^{1/2}).
\end{aligned} \tag{3}$$

Note that $\boldsymbol{\beta}_1 = \mathbf{0}$, then

$$\begin{aligned}
B_{\Psi_1}(\boldsymbol{\beta}^*, \boldsymbol{\beta}_1) &= \frac{1}{2}\left\langle \boldsymbol{\beta}^*, (\sigma I_d + \widetilde{X}_1^{1/2})\boldsymbol{\beta}^*\right\rangle \\
&\leq \frac{1}{2}\sigma\|\boldsymbol{\beta}^*\|_2^2 + \frac{2+\epsilon}{4}\|\boldsymbol{\beta}^*\|_2^2 \mathrm{tr}(X_1^{1/2})
\end{aligned} \tag{4}$$

where the inequality is due to Corollary 1.

Then, we consider the bound of $\sum_{t=1}^T \|f_t'(\boldsymbol{\beta}_t)\|_{\Psi_t^*}^2$. With probability $1-\delta$, we have

$$\begin{aligned}
&\frac{1}{2}\|f_t'(\boldsymbol{\beta}_t)\|_{\Psi_t^*}^2 = \left\langle \mathbf{g}_t, (\sigma I_d + \widetilde{X}_t^{1/2})^{-1}\mathbf{g}_t\right\rangle \\
&\leq \frac{1}{\sqrt{1-\epsilon}}\left\langle \mathbf{g}_t, (X_t^\dagger)^{1/2}\mathbf{g}_t\right\rangle = \frac{l'(\boldsymbol{\beta}_t^\top\mathbf{x}_t)^2}{\sqrt{1-\epsilon}}\left\langle \mathbf{x}_t, (X_t^\dagger)^{1/2}\mathbf{x}_t\right\rangle
\end{aligned}$$

where the inequality is due to Corollary 1. According to Lemma 2, we have

$$\begin{aligned}
&\sum_{t=1}^{T}\|f_t'(\boldsymbol{\beta}_t)\|_{\Psi_t^*}^2 \\
&\leq \sum_{t=1}^{T}\frac{2l'(\boldsymbol{\beta}_t^\top\mathbf{x}_t)^2}{\sqrt{1-\epsilon}}\left\langle \mathbf{x}_t, (X_t^\dagger)^{1/2}\mathbf{x}_t\right\rangle \\
&\leq \max_{t\leq T} l'(\boldsymbol{\beta}_t^\top\mathbf{x}_t)^2 \frac{2}{\sqrt{1-\epsilon}}\sum_{t=1}^{T}\left\langle \mathbf{x}_t, (X_t^\dagger)^{1/2}\mathbf{x}_t\right\rangle \\
&\leq \frac{4}{\sqrt{1-\epsilon}}\max_{t\leq T} l'(\boldsymbol{\beta}_t^\top\mathbf{x}_t)^2 \mathrm{tr}(X_T^{1/2}).
\end{aligned} \tag{5}$$

We complete the proof by substituting (3), (4) and (5) into Lemma 1.

A.3 Proof of Theorem 2

Inspired by the proof of Theorem 1, we can derive Theorem 2 by, respectively, bounding each term in the upper bound of Lemma 1. Before that, we need to derive the lower and upper bounds of $(S_t^\top S_t)^{1/2}$ based on Corollary 2.

Let the SVD of $C_t^\top$ be $C_t^\top = U\Sigma V^\top$ where $U \in \mathbb{R}^{d\times d}$, $\Sigma \in \mathbb{R}^{d\times d}$, $V \in \mathbb{R}^{t\times d}$. Let $K_t = \alpha I_d + C_t^\top C_t$, $\tilde{K}_t = \alpha I_d + S_t^\top S_t$ and $\tilde{I}_t = K_t^{-1/2}\tilde{K}_t K_t^{-1/2}$. According to Corollary 2, with probability at least $1-\delta$, simultaneously for all $t = 1, \ldots, T$,

$$\begin{aligned} S_t^\top S_t &= \tilde{K}_t - \alpha I_d = K_t^{1/2}\tilde{I}_t K_t^{1/2} - \alpha I_d \\ &\preceq (1+\epsilon)K_t - \alpha I_d = (1+\epsilon)C_t^\top C_t + \epsilon\alpha I_d \\ &= U\big((1+\epsilon)\Sigma\Sigma + \epsilon\alpha I_d\big)U^\top \end{aligned}$$

and

$$\begin{aligned} S_t^\top S_t + \epsilon\alpha I_d &= \tilde{K}_t - \alpha I_d + \epsilon\alpha I_d \\ &= K_t^{1/2}\tilde{I}_t K_t^{1/2} - \alpha I_d + \epsilon\alpha I_d \\ &\succeq (1-\epsilon)K_t - \alpha I_d + \epsilon\alpha I_d \\ &= (1-\epsilon)C_t^\top C_t. \end{aligned}$$

Then simultaneously for all $t = 1, \ldots, T$, we have

$$\begin{aligned} (S_t^\top S_t)^{1/2} &\preceq \sqrt{1+\epsilon}U(\Sigma\Sigma)^{1/2}U^\top + \sqrt{\epsilon\alpha}UI_dU^\top \\ &= \sqrt{1+\epsilon}X_t^{1/2} + \sqrt{\epsilon\alpha}I_d \end{aligned} \tag{6}$$

and

$$\begin{aligned} (S_t^\top S_t)^{1/2} &= (S_t^\top S_t)^{1/2} + \sqrt{\epsilon\alpha}I_d - \sqrt{\epsilon\alpha}I_d \\ &\succeq \big((S_t^\top S_t) + \epsilon\alpha I_d\big)^{1/2} - \sqrt{\epsilon\alpha}I_d \\ &\succeq \sqrt{1-\epsilon}X_t^{1/2} - \sqrt{\epsilon\alpha}I_d. \end{aligned} \tag{7}$$

Then we consider bounding the first term in the upper bound of Lemma 1. Let $\widetilde{X}_t$ denote $S_t^\top S_t$. Simultaneously for all $t = 1, \ldots, T$, we have

$$\begin{aligned} &B_{\Psi_{t+1}}(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{t+1}) - B_{\Psi_t}(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{t+1}) \\ &= \frac{1}{2}\left\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}, (\widetilde{X}_{t+1}^{1/2} - \widetilde{X}_t^{1/2})(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\right\rangle \\ &\le \frac{1}{2}\left\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}, \sqrt{1+\epsilon}X_{t+1}^{1/2}(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\right\rangle \\ &\quad - \frac{1}{2}\left\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}, \sqrt{1-\epsilon}X_t^{1/2})(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\right\rangle \\ &\quad + \frac{1}{2}\left\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}, 2\sqrt{\epsilon\alpha}I_d(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\right\rangle \\ &= \frac{1}{2}\left\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}, \sqrt{1+\epsilon}X_{t+1}^{1/2}(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\right\rangle \\ &\quad - \frac{1}{2}\left\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}, \sqrt{1-\epsilon}X_t^{1/2})(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\right\rangle \\ &\quad + \sqrt{\epsilon\alpha}\|(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\|_2^2 \\ &\le \frac{1}{2}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}\|_2^2 \operatorname{tr}(X_{t+1}^{1/2} - X_t^{1/2}) \\ &\quad + \frac{\epsilon}{4}\left\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}, (X_{t+1}^{1/2} + X_t^{1/2})(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\right\rangle \\ &\quad + \sqrt{\epsilon\alpha}\|(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1})\|_2^2 \end{aligned}$$

where the first inequality is due to (6), (7) and the last inequality has been proved in the proof of Theorem 1.

Thus, we can get

$$\begin{aligned} &\sum_{t=1}^{T-1}\left[B_{\Psi_{t+1}}(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{t+1}) - B_{\Psi_t}(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{t+1})\right] \\ &\le \frac{1}{2}\max_{t\le T}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_t\|_2^2 \operatorname{tr}(X_T^{1/2}) - \frac{1}{2}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_1\|_2^2 \operatorname{tr}(X_1^{1/2}) \\ &\quad + \frac{\epsilon}{2}\max_{t\le T}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_t\|_2^2 \sum_{t=1}^{T}\|X_t^{1/2}\| \\ &\quad - \frac{\epsilon}{4}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_1\|_2^2 \operatorname{tr}(X_1^{1/2}) \\ &\quad + \sqrt{\epsilon\alpha}(T-1)\max_{t\le T}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}_t\|_2^2. \end{aligned} \tag{8}$$

Note that $\boldsymbol{\beta}_1 = \mathbf{0}$, then

$$\begin{aligned} B_{\Psi_1}(\boldsymbol{\beta}^*, \boldsymbol{\beta}_1) &= \frac{1}{2}\left\langle \boldsymbol{\beta}^*, (\sigma I_d + \widetilde{X}_1^{1/2})\boldsymbol{\beta}^*\right\rangle \\ &\le \frac{1}{2}\sigma\|\boldsymbol{\beta}^*\|_2^2 + \frac{2+\epsilon}{4}\|\boldsymbol{\beta}^*\|_2^2 \operatorname{tr}(X_1^{1/2}) \\ &\quad + \frac{1}{2}\sqrt{\epsilon\alpha}\|\boldsymbol{\beta}^*\|_2^2. \end{aligned} \tag{9}$$

Before considering the upper bound of $\sum_{t=1}^{T}\|f_t'(\boldsymbol{\beta}_t)\|_{\Psi_t^*}^2$, we need to derive the upper bound of H_t^{-1}.

Let the SVD of $S_t^\top$ be $S_t^\top = U\Sigma V^\top$ where $U \in \mathbb{R}^{d\times d}$, $\Sigma \in \mathbb{R}^{d\times d}$, $V \in \mathbb{R}^{t\times d}$. We also have, for all $t = 1, \ldots, T$,

$$\begin{aligned} H_t &= \sigma I_d + (S_t^\top S_t)^{1/2} = U\big(\sigma I_d + (\Sigma\Sigma)^{1/2}\big)U^\top \\ &\succeq U\big(\alpha I_d + (\Sigma\Sigma)\big)^{1/2}U^\top = (\alpha I_d + S_t^\top S_t)^{1/2} \end{aligned}$$

due to $\sigma \ge \sqrt{\alpha} \ge \sqrt{\lambda_i(S_t^\top S_t) + \alpha} - \sqrt{\lambda_i(S_t^\top S_t)}$ for all $i = 1, \ldots, d$.

Then according to Corollary 2, with probability at least $1-\delta$, simultaneously for all $t = 1, \ldots, T$,

$$H_t^{-1} \preceq \left((\alpha I_d + S_t^\top S_t)^{1/2}\right)^{-1} = \left((K_t^{1/2}\tilde{I}_t K_t^{1/2})^{-1}\right)^{1/2}$$
$$\preceq \frac{1}{\sqrt{1-\epsilon}}(K_t^{-1})^{1/2} = \frac{1}{\sqrt{1-\epsilon}}\left((\alpha I_d + X_t)^{-1}\right)^{1/2}.$$

Thus, we can get

$$\|f_t'(\boldsymbol{\beta}_t)\|_{\Psi_t^*}^2 = 2\left\langle \mathbf{g}_t, H_t^{-1}\mathbf{g}_t\right\rangle$$
$$\leq \frac{2}{\sqrt{1-\epsilon}}\left\langle \mathbf{g}_t, \left((\alpha I_d + X_t)^{-1}\right)^{1/2}\mathbf{g}_t\right\rangle$$
$$= \frac{2l'(\boldsymbol{\beta}_t^\top \mathbf{x}_t)^2}{\sqrt{1-\epsilon}}\left\langle \mathbf{x}_t, (X_t^\dagger)^{1/2}\mathbf{x}_t\right\rangle.$$

According to Lemma 2, we have

$$\sum_{t=1}^{T}\|f_t'(\boldsymbol{\beta}_t)\|_{\Psi_t^*}^2$$
$$\leq \frac{2}{\sqrt{1-\epsilon}}\max_{t\leq T} l'(\boldsymbol{\beta}_t^\top \mathbf{x}_t)^2 \sum_{t=1}^{T}\left\langle \mathbf{x}_t, (X_t^\dagger)^{1/2}\mathbf{x}_t\right\rangle \tag{10}$$
$$\leq \frac{4}{\sqrt{1-\epsilon}}\max_{t\leq T} l'(\boldsymbol{\beta}_t^\top \mathbf{x}_t)^2 \operatorname{tr}(X_T^{1/2}).$$

We complete the proof by substituting (8), (9) and (10) into Lemma 1.

A.4 Proof of Corollary 1

Let $C_t = U\Sigma V^\top$ be the singular value decomposition of C_t. Notice that $U \in \mathbb{R}^{t\times r}$, $\Sigma V^\top \in \mathbb{R}^{r\times d}$. According to Theorem 3, we have if $\tau = \Theta(\frac{r+\log(1/\delta)}{\epsilon^2})$, then simultaneously $\forall \mathbf{x} \in \mathbb{R}^r$, with probability $1-\delta$,

$$(1-\epsilon)\|U\mathbf{x}\|_2^2 \leq \|R_t U\mathbf{x}\|_2^2 \leq (1+\epsilon)\|U\mathbf{x}\|_2^2$$

Let $\mathbf{y} \in \mathbb{R}^d$ be arbitrary vector, then $C_t\mathbf{y} = U\Sigma V^\top \mathbf{y} = U\mathbf{x}$ where $\mathbf{x} = \Sigma V^\top \mathbf{y} \in \mathbb{R}^r$.
Then we have

$$\mathbf{y}^\top S_t^\top S_t \mathbf{y} = \mathbf{y}^\top C_t^\top R_t^\top R_t C_t \mathbf{y} = \|R_t U\mathbf{x}\|_2^2$$
$$\leq (1+\epsilon)\|U\mathbf{x}\|_2^2 = (1+\epsilon)\mathbf{y}^\top C_t^\top C_t \mathbf{y}$$

and

$$\mathbf{y}^\top S_t^\top S_t \mathbf{y} = \mathbf{y}^\top C_t^\top R_t^\top R_t C_t \mathbf{y} = \|R_t U\mathbf{x}\|_2^2$$
$$\geq (1-\epsilon)\|U\mathbf{x}\|_2^2 = (1-\epsilon)\mathbf{y}^\top C_t^\top C_t \mathbf{y}.$$

Then, we have $(1-\epsilon)C_t^\top C_t \preceq S_t^\top S_t \preceq (1+\epsilon)C_t^\top C_t$ with probability $1-\delta$, provided $\tau = \Omega(\frac{r+\log(1/\delta)}{\epsilon^2})$. Using the union bound, we have if $\tau = \Omega(\frac{r+\log(T/\delta)}{\epsilon^2})$, with probability $1-\delta$, simultaneously for all $t = 1, \ldots, T$,

$$(1-\epsilon)C_t^\top C_t \preceq S_t^\top S_t \preceq (1+\epsilon)C_t^\top C_t.$$

A.5 Proof of Corollary 2

Define the SVD of $C_t^\top$ as $C_t^\top = U\Sigma V^\top$ where $U \in \mathbb{R}^{d\times d}$, $\Sigma \in \mathbb{R}^{d\times d}$, $V \in \mathbb{R}^{t\times d}$. Then we have $K_t = U(\alpha I_d + \Sigma\Sigma^\top)U^\top$ and

$$\tilde{I}_t = K_t^{-1/2}\tilde{K}_t K_t^{-1/2} = K_t^{-1/2}(\alpha I_d + C_t^\top R_t^\top R_t C_t)K_t^{-1/2}$$
$$= U\Big((\alpha I_p + \Sigma\Sigma)^{-1/2}\Sigma V^\top R_t^\top R_t V \Sigma(\alpha I_d + \Sigma\Sigma^\top)^{-1/2}$$
$$+ \alpha I_d(\alpha I_d + \Sigma\Sigma)^{-1}\Big)U^\top$$
$$= U\Big((\alpha I_p + \Sigma\Sigma)^{-1/2}\Sigma R R^\top \Sigma(\alpha I_d + \Sigma\Sigma^\top)^{-1/2}$$
$$+ \alpha I_d(\alpha I_d + \Sigma\Sigma)^{-1}\Big)U^\top$$

where $R = V^\top R_t^\top \in \mathbb{R}^{d\times\tau}$ is a Gaussian random matrix due to that V is an orthogonal matrix and $R_t^\top$ is a Gaussian random matrix. Let $c_i^2 = \frac{\alpha}{\alpha+\sigma_{ti}^2}$ and $s_i^2 = \frac{\sigma_{ti}^2}{\alpha+\sigma_{ti}^2}$. Then according to Theorem 4, with probability at least $1-\delta$,

$$(1-\epsilon)I_d \preceq \tilde{I}_t \preceq (1+\epsilon)I_d$$

provided $\tau \geq \frac{\tilde{r}_t\sigma_{t1}^2}{c\epsilon^2(\alpha+\sigma_{t1}^2)}\log\frac{2d}{\delta}$ where the constant c is at least $1/32$. Using the union bound, we complete the proof.

References

1. Achlioptas, D.: Database-friendly random projections: Johnson–Lindenstrauss with binary coins. J. Comput. Syst. Sci. **66**(4), 671–687 (2003)
2. Allesiardo, R., Fraud, R., Maillard, O.A.: The non-stationary stochastic multi-armed bandit problem. Int. J. Data Sci. Anal. **3**(4), 267–283 (2017)
3. Awerbuch, B., Kleinberg, R.: Online linear optimization and adaptive routing. J. Comput. Syst. Sci. **74**(1), 97–114 (2008)
4. Boutsidis, C., Zouzias, A., Drineas, P.: Random projections for k-means clustering. In: Advances in Neural Information Processing Systems, vol. 23, pp. 298–306 (2010)
5. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 1–27 (2011)
6. Dasgupta, S., Freund, Y.: Random projection trees and low dimensional manifolds. In: Proceedings of the 40th Annual ACM Symposium on Theory of computing, pp. 537–546 (2008)
7. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
8. Duchi, J., Shalev-Shwartz, S., Singer, Y., Tewari, A.: Composite objective mirror descent. In: Proceedings of the 23rd Annual Conference on Learning Theory, pp. 14–26 (2010)

9. Fern, X.Z., Brodley, C.E.: Random projection for high dimensional data clustering: a cluster ensemble approach. In: Proceedings of the 20th International Conference on Machine Learning, pp. 186–93 (2003)
10. Fradkin, D., Madigan, D.: Experiments with random projections for machine learning. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 517–522 (2003)
11. Freund, Y., Dasgupta, S., Kabra, M., Verma, N.: Learning the structure of manifolds using random projections. In: Advances in Neural Information Processing Systems, vol. 21, pp. 473–480 (2008)
12. Gao, W., Jin, R., Zhu, S., Zhou, Z.H.: One-pass AUC optimization. In: Proceedings of the 30th International Conference on Machine Learning, pp. 906–914 (2013)
13. Ghashami, M., Liberty, E., Phillips, J.M., Woodruff, D.P.: Frequent directions: simple and deterministic matrix sketching. SIAM J. Comput. **45**(5), 1762–1792 (2016)
14. Hager, W.W.: Updating the inverse of a matrix. SIAM Rev. **31**(2), 221–239 (1989)
15. Hassani, M., Töws, D., Cuzzocrea, A., Seidl, T.: BFSPMiner: an effective and efficient batch-free algorithm for mining sequential patterns over data streams. Int. J. Data Sci. Anal. 1–17 (2017). https://doi.org/10.1007/S41060-017-0084-8
16. Hazan, E., Agarwal, A., Kale, S.: Logarithmic regret algorithms for online convex optimization. Mach. Learn. **69**(2), 169–192 (2007)
17. Kaski, S.: Dimensionality reduction by random mapping: fast similarity computation for clustering. In: Proceedings of the 1998 IEEE International Joint Conference on Neural Networks, vol. 1, pp. 413–418 (1998)
18. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report, University of Toronto (2009)
19. Krummenacher, G., McWilliams, B., Kilcher, Y., Buhmann, J.M., Meinshausen, N.: Scalable adaptive stochastic optimization using random projections. In: Advances in Neural Information Processing Systems, vol. 29, pp. 1750–1758 (2016)
20. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, vol. 86, pp. 2278–2324 (1998)
21. Liberty, E., Ailon, N., Singer, A.: Dense fast random projections and lean walsh transforms. Discrete Comput. Geom. **45**(1), 34–44 (2011)
22. Luo, H., Agarwal, A., Cesa-Bianchi, N., Langford, J.: Efficient second order online learning by sketching. In: Advances in Neural Information Processing Systems, vol. 29, pp. 902–910 (2016)
23. Magen, A., Zouzias, A.: Low rank matrix-valued Chernoff bounds and approximate matrix multiplication. In: Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1422–1436 (2011)
24. Maillard, O.A., Munos, R.: Linear regression with random projections. J. Mach. Learn. Res. **13**, 2735–2772 (2012)
25. Miyaguchi, K., Yamanishi, K.: Online detection of continuous changes in stochastic processes. Int. J. Data Sci. Anal. **3**(3), 213–229 (2017)
26. Nalko, N., Martinsson, P.G., Tropp, J.A.: Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. **53**(2), 217–288 (2011)
27. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011 (2011)
28. Rahimi, A., Recht, B.: Random features for large-scale kernel machines. In: Advances in Neural Information Processing Systems, vol. 21, pp. 1177–1184 (2008)
29. Tropp, J.A.: An introduction to matrix concentration inequalities. Found. Trends Mach. Learn. **8**(1–2), 1–230 (2015)
30. Wan, Y., Wei, N., Zhang, L.: Efficient adaptive online learning via frequent directions. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, pp. 2748–2754 (2018)
31. Wan, Y., Zhang, L.: Accelerating adaptive online learning by matrix approximation. In: Proceedings of the 22nd Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 405–417 (2018)
32. Woodruff, D.P.: Sketching as a tool for numerical linear algebra. Found. Trends Mach. Learn. **10**(1–2), 1–157 (2014)
33. Xiao, L.: Dual averaging method for regularized stochastic learning and online optimization. In: Advances in Neural Information Processing Systems, vol. 22, pp. 2116–2124 (2009)
34. Yenala, H., Jhanwar, A., Chinnakotla, M.K., Goyal, J.: Deep learning for detecting inappropriate content in text. Int. J. Data Sci. Anal. **6**(4), 273–286 (2018)
35. Zhang, L., Mahdavi, M., Jin, R., Yang, T., Zhu, S.: Recovering the optimal solution by dual random projection. In: Proceedings of the 26th Annual Conference on Learning Theory, pp. 135–157 (2013)
36. Zhang, L., Yang, T., Jin, R., Xiao, Y., Zhou, Z.H.: Online stochastic linear optimization under one-bit feedback. In: Proceedings of the 33rd International Conference on Machine Learning, pp. 392–401 (2016)
37. Zinkevich, M.: Online convex programming and generalized infinitesimal gradient ascent. In: Proceedings of the 20th International Conference on Machine Learning, pp. 928–936 (2003)