Effective Data-Aware Covariance Estimator From Compressed Data

Xixian Chen, Haiqin Yang, *Member, IEEE*, Shenglin Zhao, *Member, IEEE*, Michael R. Lyu, *Fellow, IEEE*, and Irwin King, *Fellow, IEEE*

***Abstract*— Estimating covariance matrix from massive high-dimensional and distributed data is significant for various real-world applications. In this paper, we propose a data-aware weighted sampling-based covariance matrix estimator, namely DACE, which can provide an unbiased covariance matrix estimation and attain more accurate estimation under the same compression ratio. Moreover, we extend our proposed DACE to tackle multiclass classification problems with theoretical justification and conduct extensive experiments on both synthetic and real-world data sets to demonstrate the superior performance of our DACE.**

***Index Terms*— Covariance estimation, dimension reduction, randomized algorithms, unsupervised learning.**

I. Introduction

COVARIANCE matrix, absorbing the second-order information of data points, plays a significant role in many machine learning and statistics applications [23]. For example, the estimated covariance matrix plays the role of dimension reduction or denoising for the principal component analysis (PCA) [48], the linear discriminant analysis (LDA), and the quadratic discriminant analysis (QDA) [8]. Via an estimated noise covariance matrix, generalized least squares (GLS) regression can attain the best linear estimator [33]. The independent component analysis (ICA) relies on the covariance matrix for pre-whitening [31]. The generalized method of moments (GMM) [28] improves the effectiveness of the model by estimating a precise covariance matrix. Many real-world applications, such as gene relevance networks [12], [40], modern wireless communications [45], array signal processing [2], and policy learning [20], also rely on directly estimating the covariance matrix [10].

Manuscript received February 10, 2018; revised December 16, 2018, March 13, 2019, and July 3, 2019; accepted July 5, 2019. Date of publication August 14, 2019; date of current version July 7, 2020. This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (CUHK No. 14208815, CUHK No. 14234416, and Project No. UGC/IDS14/16). *(Corresponding author: Haiqin Yang.)*

X. Chen and S. Zhao are with the Youtu Lab, Tencent, Shenzhen 518057, China (e-mail: xixianchen@tencent.com; henryslzhao@tencent.com).

H. Yang is with Meitu, Hong Kong, and also with the Department of Computing, Hang Seng University of Hong Kong, Hong Kong (e-mail: haiqin.yang@gmail.com).

M. R. Lyu and I. King are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, and also with the Shenzhen Research Institute, The Chinese University of Hong Kong, Shenzhen 518057, China (e-mail: lyu@cse.cuhk.edu.hk; king@cse.cuhk.edu.hk).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2019.2929106

Nowadays, large and high-dimensional data are routinely generated in various distributed applications, such as sensor networks, surveillance systems, and distributed databases [26], [29], [41]. The communication cost becomes challenging because the distributed data need to be transmitted to a fusion center from remote sites, requiring tremendous bandwidth and power consumption [1], [42]. One effective solution is to utilize the compressed data, i.e., projecting the original data to a small-sized one via a Gaussian matrix, where the space cost, the computational cost, and the communication cost can be reduced significantly to linearly depend on the projected size. However, the above solution suffers from two critical drawbacks. First, projecting to a Gaussian matrix is inefficient compared with computing sparse projection matrices [34], structured matrices [14], or sampling matrices [22]. Second, applying the same projection matrix to all data points cannot recover the original covariance matrix precisely. Current theoretical investigation and empirical results show that even the size of the samples with a fixed dimension increases to infinity, and the estimator cannot recover the target covariance matrix [6], [7], [9], [24].

To tackle the above challenges, we propose a data-aware convariance matrix estimator, namely DACE, to leverage different projection matrices for each data point. It is known that without statistical assumptions or low-rank/sparsity structural assumptions on the distribution of the data, our DACE can achieve consistent covariance matrix estimation. By a crafty designed weighted sampling scheme, we can compress the data and recover the covariance matrix in the center efficiently and precisely. We summarize our contributions as follows.

1) First, we propose a data-aware covariance matrix estimator by a weighted sampling scheme. This is different from existing data-oblivious projection methods [5]–[7], [9]. By exploiting the most important entries, our strategy requires considerably fewer entries to achieve the same estimation precision.
2) Second, we rigorously prove that our DACE is an unbiased covariance estimator. Moreover, our DACE can achieve more accurate estimation precision and consume less time cost than existing methods under the same compression ratio. The theoretical justification is verified in both synthetic and real-world data sets.
3) Third, we extend our DACE to tackle the multiclass classification problem and provide both theoretical justification and empirical evaluation. The compact theoretical result and the superior empirical performance imply that

the covariance matrix estimated from compressed data indeed guarantees the intrinsic properties of data and can be applied in various down-stream applications.

II. Problem Definition and Related Work

A. Notations and Problem Definition

Following the notations defined in [15], given n distributed data in g remote sites, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $g \ll n$, the corresponding covariance matrix can be computed by $\mathbf{C} = (1/n)\mathbf{X}\mathbf{X}^T - \bar{\mathbf{x}}\bar{\mathbf{x}}^T$, where $\bar{\mathbf{x}} = (1/n)\sum_{i=1}^{n}\mathbf{x}_i$ can be exactly computed in the fusion center through $\bar{\mathbf{x}} = (1/n)\sum_{j=1}^{g}\mathbf{g}_j$, where $\mathbf{g}_j \in \mathbb{R}^d$ represents the summation of all data points in the jth remote site before being compressed. Hence, without loss of generality, we can assume zero-mean, i.e., $\bar{\mathbf{x}} = \mathbf{0}$.

Now, we define the procedure of covariance matrix recovery as follows: given data $\mathbf{X}$ and specific designed sampling matrices, $\{\mathbf{S}_i\}_{i=1}^{n} \in \mathbb{R}^{d\times m}$, where $m \ll \{d, n\}$, the original data is compressed via $\mathbf{S}_i^T\mathbf{x}_i$ and transmitted to the fusion center while the covariance matrix of the original data is recovered by a transformation only via $\mathbf{S}_i$. The question is how to design the sampling matrices to guarantee the estimated covariance matrix as precisely as possible.

B. Related Work

Various randomized algorithms have been proposed to tackle the above problem and can be divided into three main streams.

1) *Independent Projection:* The *Gaussian-inverse* method [39] applies a Gaussian matrix $\mathbf{S}_i$ to compress each data point and recovers the data via $\mathbf{S}_i(\mathbf{S}_i^T\mathbf{S}_i)^{-1}(\mathbf{S}_i^T\mathbf{x}_i)$. The information of all entries in each data vector is very likely to be acquired uniformly and substantively because $\mathbf{S}_i(\mathbf{S}_i^T\mathbf{S}_i)^{-1}\mathbf{S}_i^T$ is an m-dimensional orthogonal projection, whose projection spaces are uniformly and randomly drawn. Hence, $\frac{1}{n}\sum_{i=1}^{n}\mathbf{S}_i(\mathbf{S}_i^T\mathbf{S}_i)^{-1}\mathbf{S}_i^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{S}_i(\mathbf{S}_i^T\mathbf{S}_i)^{-1}\mathbf{S}_i^T$ is expected to constitute an accurate and consistent covariance matrix estimation up to a known scaling factor [9]. However, computing the Gaussian matrix is computational burden because the Gaussian matrix is dense and unstructured. Moreover, the matrix inverse operation requires much computational time and memory cost. A biased estimator $(1/n)\sum_{i=1}^{n}\mathbf{S}_i\mathbf{S}_i^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{S}_i\mathbf{S}_i^T$ is then presented in [7] by applying a sparse matrix $\mathbf{S}_i$ to avoid computing the matrix inversion. This strategy is less accurate because $\mathbf{S}_i\mathbf{S}_i^T$ approximates only an m-dimensional random orthogonal projection. Moreover, its performance is guaranteed only on data that satisfy a certain statistical assumption, e.g., Gaussian distribution. An unbiased estimator [5] is then proposed to adopt an unstructured sparse matrix to construct the projection. The method is computational inefficient and fails to afford error bounds to trade off the estimation error and the compression ratio. To improve computational efficiency, the strategy of sampling *without replacement* has been employed to $\mathbf{S}_i$. However, this method recovers the data via $\mathbf{S}_i^T\mathbf{x}_i$, which is poor because $\mathbf{S}_i\mathbf{S}_i^T$ is an m-dimensional orthogonal projection drawn only from d deterministic orthogonal spaces/coordinates and removes $(d - m)$ entries of each vector. To retain the accuracy, the Hadamard matrix [43] is applied in [6] before sampling, which flattens out all entries, particularly those with large magnitudes, to all coordinates. Even though the proposed uniform sampling scheme can capture sufficient information embedded in all entries, it fails to capture the information uniformly in all coordinates of each vector because the Hadamard matrix involves deterministic orthogonal projection. Hence, it requires numerous samples to obtain sufficient accuracy [6]. Overall, existing independent projection methods cannot capture the most valuable information sufficiently.
2) *Projection via a Low-Rank Matrix:* A representative work [16], [27] is to improve the approximation precision by projecting the original data via a low-dimensional data-aware matrix $\mathbf{X}\widehat{\mathbf{S}}$, where $\widehat{\mathbf{S}}$ is a random projection matrix and $\mathbf{X}$ must be a low-rank matrix. This method has to take one extra pass through all entries in $\mathbf{X}$ to compute $\mathbf{X}\widehat{\mathbf{S}}$. Theoretical and empirical investigation shows that a single projection matrix for all data points cannot consistently and accurately estimate the covariance matrix [9]. The problem of inconsistent covariance estimation and the restriction of low-rank matrix assumption also exist in [37] and [47] for fast approximating matrix products in a single pass.
3) *Sampling in a Whole:* Other existing methods [21], [30], [38], [46] leverage column-based sampling to apply the column norms or leverage scores in the sampling probabilities matrix, while in 3], [4], and [46], element-wise sampling is applied in the entire matrix. These methods adopt various sampling distributions to sample entries from a matrix. However, they require one or more extra passes over data because computing the sampling distributions requires to observe all data. Moreover, the sampling probabilities are created for matrix approximation and cannot be trivially extended to covariance matrix estimation because it is not allowed to obtain the exact covariance matrix in advance. Note that although the uniform sampling is a simple one-pass algorithm for matrix approximation, the structural non-uniformity in the data makes it perform poorly [6].

Other than randomized algorithms, researchers also establish theory to recover the covariance matrix from given data [11], [13], [17], [18]. However, these methods are only applicable when the covariance matrix is low-rank, sparse, or follows a certain statistical assumption, and restrict their application potentials.

III. Our Proposal

A. Method and Algorithm

Our proposed DACE utilizes data-aware weighted sampling matrices $\{\mathbf{S}_i\}_{i=1}^{n}$ to compress each data via $\mathbf{S}_i^T\mathbf{x}_i$ and then

Algorithm 1 Data-Aware Covariance Estimator (DACE)

Require:
Data $\mathbf{X} \in \mathbb{R}^{d\times n}$, sampling size m, and $0 < \alpha < 1$.

Ensure:
Estimated covariance matrix $\mathbf{C}_e \in \mathbb{R}^{d\times d}$.

1: Initialize $\mathbf{Y} \in \mathbb{R}^{m\times n}$, $\mathbf{T} \in \mathbb{R}^{m\times n}$, $\mathbf{v} \in \mathbb{R}^n$, and $\mathbf{w} \in \mathbb{R}^n$ with $\mathbf{0}$.
2: **for** all $i \in [n]$ **do**
3: Load $\mathbf{x}_i$ into memory, let $v_i = \|\mathbf{x}_i\|_1 = \sum_{k=1}^{d} |x_{ki}|$ and $w_i = \|\mathbf{x}_i\|_2^2 = \sum_{k=1}^{d} x_{ki}^2$
4: **for** all $j \in [m]$ **do**
5: Pick $t_{ji} \in [d]$ with $p_{ki} \equiv \mathbb{P}(t_{ji} = k) = \alpha\frac{|x_{ki}|}{v_i} + (1-\alpha)\frac{x_{ki}^2}{w_i}$, and let $y_{ji} = x_{t_{ji}i}$
6: **end for**
7: **end for**
8: Pass the compressed data $\mathbf{Y}$, sampling indices $\mathbf{T}$, $\mathbf{v}$, $\mathbf{w}$, and α to the fusion center.
9: **for** all $i \in [n]$ **do**
10: Initialize $\mathbf{S}_i \in \mathbb{R}^{d\times m}$ and $\mathbf{P} \in \mathbb{R}^{d\times n}$ with $\mathbf{0}$
11: **for** all $j \in [m]$ **do**
12: Let $p_{t_{ji}i} = \alpha\frac{|y_{ji}|}{v_i} + (1-\alpha)\frac{y_{ji}^2}{w_i}$, and $s_{t_{ji}j,i} = \frac{1}{\sqrt{mp_{t_{ji}i}}}$
13: **end for**
14: **end for**
15: Compute $\mathbf{C}_e$ as defined in Eq. (1).

back-project the compressed data into the original space via $\mathbf{S}_i\mathbf{S}_i^T\mathbf{x}_i$. The estimated covariance matrix is computed by

$$\mathbf{C}_e = \widehat{\mathbf{C}}_1 - \widehat{\mathbf{C}}_2, \quad \text{where} \quad b_{ki} = \frac{1}{1+(m-1)p_{ki}} \tag{1}$$

$$\widehat{\mathbf{C}}_1 = \frac{m}{nm-n}\sum_{i=1}^{n}\mathbf{S}_i\mathbf{S}_i^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{S}_i\mathbf{S}_i^T \tag{2}$$

$$\widehat{\mathbf{C}}_2 = \frac{m}{nm-n}\sum_{i=1}^{n}\mathbb{D}\big(\mathbf{S}_i\mathbf{S}_i^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{S}_i\mathbf{S}_i^T\big)\mathbb{D}(\mathbf{b}_i). \tag{3}$$

In (1), at most m entries in each $\mathbf{b}_i$ have to be calculated because each $\mathbf{S}_i\mathbf{S}_i^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{S}_i\mathbf{S}_i^T$ contains at most m non-zero entries in its diagonal.

Algorithm 1 outlines the flow of DACE. Steps 1–7 show the procedure of compressing the data $\mathbf{X}$ to $\mathbf{Y}$, where each entry is retained according to the probability proportional to the combination of its relative absolute value and the square value. Step 8 describes the communication procedure to transmit the compressed data from all the remote sites to the fusion center. Steps 9–14 reveal the construction of an unbiased covariance matrix estimator in the fusion center from the compressed data. It is shown that only one pass is required to load all data from the external space into the memory, which reveals the applicability of our DACE for streaming data.

B. Primary Provable Results

Theorem 1 shows that our proposed DACE can attain an unbiased estimator for the target covariance matrix.

Theorem 1: Assume $\mathbf{X} \in \mathbb{R}^{d\times n}$ and the sampling size $2 \le m < d$. m entries are sampled from each $\mathbf{x}_i \in \mathbb{R}^d$ with replacement by running Algorithm 1. Let $\{p_{ki}\}_{k=1}^{d}$ and $\mathbf{S}_i \in \mathbb{R}^{d\times m}$ denote the sampling probabilities and the sampling matrix, respectively. Then, the unbiased estimator for the target covariance matrix $\mathbf{C} = (1/n)\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T = (1/n)\mathbf{X}\mathbf{X}^T$ can be recovered by (1).

The detailed proof is provided in Appendix V-B. The estimation error can also be bounded by Theorem 2.

Theorem 2: Let $\mathbf{X}$, m, $\mathbf{C}$, and $\mathbf{C}_e$ be defined as in Theorem 1. If the sampling probabilities satisfy $p_{ki} = \alpha(|x_{ki}|/\|\mathbf{x}_i\|_1) + (1-\alpha)(x_{ki}^2/\|\mathbf{x}_i\|_2^2)$ with $0 < \alpha < 1$ for all $k \in [d]$ and $i \in [n]$, then with probability at least $1-\eta-\delta$

$$\|\mathbf{C}_e - \mathbf{C}\|_2 \le \log\left(\frac{2d}{\delta}\right)\frac{2R}{3} + \sqrt{2\sigma^2\log\left(\frac{2d}{\delta}\right)} \tag{4}$$

where $R = \max_{i\in[n]}[(7\|\mathbf{x}_i\|_2^2/n) + \log^2((2nd/\eta))(14\|\mathbf{x}_i\|_1^2/nm\alpha^2)]$ and $\sigma^2 = \sum_{i=1}^{n}[(8\|\mathbf{x}_i\|_2^4/n^2m^2(1-\alpha)^2) + (4\|\mathbf{x}_i\|_1^2\|\mathbf{x}_i\|_2^2/n^2m^3\alpha^2(1-\alpha)) + (9\|\mathbf{x}_i\|_2^4/n^2m(1-\alpha)) + (2\|\mathbf{x}_i\|_2^2\|\mathbf{x}_i\|_1^2/n^2m^2\alpha(1-\alpha))] + \|\sum_{i=1}^{n}(\|\mathbf{x}_i\|_1^2\mathbf{x}_i\mathbf{x}_i^2/n^2m\alpha)\|_2$.

The proof of Theorem 2 is in Appendix V-C.

Remark 1: The error bound is linear with R and σ. The selected p_{ki} makes R and σ smaller and tightens the bound.

Remark 2: The balance parameter α can adjust the impact of the normalized ℓ_1-norm sampling and the square of the normalized ℓ_2-norm sampling. ℓ_2 sampling owns more potential to select larger entries to decrease error than ℓ_1 sampling, but ℓ_2 sampling is unstable and sensitive to small entries, incurring incredibly high estimation error if extremely small entries are picked. Hence, if α varies from 1 to 0, the estimation error decreases first and then increases gradually.

The explicit bound is represented in terms of n, d, and m under the constraint of $2 \le m < d$.

Corollary 1: Let $\mathbf{X}$, m, $\mathbf{C}$, and $\mathbf{C}_e$ be defined as in Theorem 1. Define $(\|\mathbf{x}_i\|_1/\|\mathbf{x}_i\|_2) \le \varphi$ with $1 \le \varphi \le \sqrt{d}$ and $\|\mathbf{x}_i\|_2 \le \tau$ for all $i \in [n]$. Then, with probability at least $1-\eta-\delta$, we have

$$\|\mathbf{C}_e-\mathbf{C}\|_2 \le \min\left\{\widetilde{O}\left(f + \frac{\tau^2\varphi}{m}\sqrt{\frac{1}{n}} + \tau^2\sqrt{\frac{1}{nm}}\right), \widetilde{O}\left(f + \frac{\tau\varphi}{m}\sqrt{\frac{d\|\mathbf{C}\|_2}{n}} + \tau\sqrt{\frac{d\|\mathbf{C}\|_2}{nm}}\right)\right\} \tag{5}$$

where $f = (\tau^2/n) + (\tau^2\varphi^2/nm) + \tau\varphi((\|\mathbf{C}\|_2/nm))^{1/2}$, and $\widetilde{O}(\cdot)$ hides the logarithmic factors on η, δ, m, n, d, and α. The proof is given by Appendix V-D.

Remark 3: When $\varphi = \sqrt{d}$, the magnitudes of each entry in all the input data vectors are the same, i.e., highly uniformly distributed. The error bound in (5) yields the worst case and derives a bound with a leading term of order $\min\{\widetilde{O}((\tau^2d/nm) + \tau((d\|\mathbf{C}\|_2/nm))^{1/2} + (\tau^2/m)((d/n))^{1/2}), \widetilde{O}((\tau^2d/nm) + (\tau d/m)((\|\mathbf{C}\|_2/n))^{1/2})\}$, the same as Gauss-Inverse, which ignores logarithmic factors [39].

Accordingly, as the magnitudes of the entries in each data vector become uneven, φ gets smaller and yields a

tighter error bound than that in Gauss-Inverse. Furthermore, when most of the entries in each vector $\mathbf{x}_i$ have very low magnitudes, the summation of these magnitudes will be comparable to a particular constant. This situation is typical because, in practice, only a limited number of features in each input data dominate the learning performance. Hence, φ turns to $O(1)$ and (5) becomes $\min\{\widetilde{O}((\tau^2/n) + \tau^2((1/nm))^{1/2}), \widetilde{O}((\tau^2/n) + \tau((d\|\mathbf{C}\|_2/nm)))^{1/2}\}$, which is tighter than the leading term of Gauss-Inverse by a factor of at least $(d/m)^{1/2}$.

Remark 4: As practically, $m \ll d$, $O(d-m)$ approximates to $O(d)$. The error of UniSample-HD is $\widetilde{O}((\tau^2 d/nm) + \tau((d\|\mathbf{C}\|_2/nm))^{1/2} + (\tau^2 d/m)((1/nm))^{1/2})$, which is asymptotically worse than our bound. When n is sufficiently large, the leading term of its error becomes $\widetilde{O}(\tau((d\|\mathbf{C}\|_2/nm))^{1/2} + (\tau^2 d/m)((1/nm))^{1/2})$, which can be weaker than the leading term in our method by a factor of 1 to $(d/m)^{1/2}$ when $\varphi = \sqrt{d}$, and at least d/m when $\varphi = O(1)$.

However, if m is close to d, though not meaningful for practical applications, $O(d-m) = O(1)$ will hold and the error of UniSample-HD becomes $\widetilde{O}((\tau^2 d/nm) + \tau((d\|\mathbf{C}\|_2/nm))^{1/2} + (\tau^2/m)((d/nm))^{1/2})$. This bound may slightly outperform ours by a factor of $(d/m)^{1/2} = O(1)$ when $\varphi = \sqrt{d}$, but is still worse than ours when $\varphi = O(1)$.

Note: The derivation and proof of our DACE do not make statistical nor structural assumptions concerning the input data or the covariance matrix. Motivated by [9], it is straightforward to extend our results to the data by a certain statistical assumption and a low-rank covariance matrix estimation.

Corollary 2: Let $\mathbf{X} \in \mathbb{R}^{d\times n}$ $(2 \le d)$, an unknown population covariance matrix $\mathbf{C}_p \in \mathbb{R}^{d\times d}$ with each column vector $\mathbf{x}_i \in \mathbb{R}^d$ i.i.d. generated from the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{C}_p)$, and $\mathbf{C}_e$ be constructed by Algorithm 1 with the sampling size $2 \le m < d$. Then, with probability at least $1-\eta-\delta-\zeta$

$$\frac{\|\mathbf{C}_e - \mathbf{C}_p\|_2}{\|\mathbf{C}_p\|_2} \le \widetilde{O}\left(\frac{d^2}{nm} + \frac{d}{m}\sqrt{\frac{d}{n}}\right). \tag{6}$$

Additionally, assuming rank $(\mathbf{C}_p) \le r$, with probability at least $1-\eta-\delta-\zeta$, we have

$$\frac{\|[\mathbf{C}_e]_r - \mathbf{C}_p\|_2}{\|\mathbf{C}_p\|_2} \le \widetilde{O}\left(\frac{rd}{nm} + \frac{r}{m}\sqrt{\frac{d}{n}} + \sqrt{\frac{rd}{nm}}\right) \tag{7}$$

where $[\mathbf{C}_e]_r$ is the solution to $\min_{\text{rank}(A)\le r}\|\mathbf{A} - \mathbf{C}_e\|_2$ and $\widetilde{O}(\cdot)$ hides the logarithmic factors on η, δ, ζ, m, n, d, and α.

Corollary 3: Given $\mathbf{X}$, d, m, $\mathbf{C}_p$, and $\mathbf{C}_e$ as defined in Corollary 2. Let $\prod_k = \sum_{i=1}^k \mathbf{u}_i\mathbf{u}_i^T$ and $\widehat{\prod}_k = \sum_{i=1}^k \hat{\mathbf{u}}_i\hat{\mathbf{u}}_i^T$ with $\{\mathbf{u}_i\}_{i=1}^k$ and $\{\hat{\mathbf{u}}_i\}_{i=1}^k$ being the leading k eigenvectors of $\mathbf{C}_p$ and $\mathbf{C}_e$, respectively. Denote by λ_k the kth largest eigenvalue of $\mathbf{C}_p$. Then, with probability at least $1-\eta-\delta-\zeta$

$$\frac{\|\widehat{\prod}_k - \prod_k\|_2}{\|\mathbf{C}_p\|_2} \le \frac{1}{\lambda_k - \lambda_{k+1}}\widetilde{O}\left(\frac{d^2}{nm} + \frac{d}{m}\sqrt{\frac{d}{n}}\right) \tag{8}$$

where the eigengap $\lambda_k - \lambda_{k+1} > 0$ and $\widetilde{O}(\cdot)$ hides the logarithmic factors on η, δ, ζ, m, n, d, and α.

TABLE I

COMPUTATIONAL COSTS ON COMMUNICATION (COMM.) AND TIME, WHERE THE STORAGE OF THE STANDARD METHOD IS $O(nd + d^2)$ WHILE OTHER METHODS CONSUME $O(nm + d^2)$

Method	Comm.	Time
Standard	$O(nd)$	$O(nd^2)$
Gauss-Inverse	$O(nm)$	$O(nmd + nm^2d + nd^2) + T_G$
Sparse	$O(nm)$	$O(d + nm^2) + T_S$
UniSample-HD	$O(nm)$	$O(nd\log d + nm^2)$
Our method	$O(nm)$	$O(nd + nm\log d + nm^2)$

The sketch proof of the above two corollaries is provided in Appendix V-E. Corollary 2 shows the (low-rank) covariance matrix estimation on Gaussian data and Corollary 3 indicates that the derived covariance estimator also guarantees the accuracy of the principal components regarding the learned subspace. In particular, setting $n = \Theta(d)$ in (7) reveals that $m = \widetilde{\Omega}(r/\epsilon^2)$ entries can achieve an ϵ spectral norm error for the low-rank covariance matrix estimation, which is also polynomially equal to the literature that leverages low-rank structure to derive methods for the low-rank (covariance) matrix recovery [13], [17].

C. Computational Complexity

In Table I, T_G and T_S represent the time taken by fast pseudorandom number generators such as Mersenne Twister [36] to generate the Gaussian matrices and sparse matrices, which can be proportional to nmd and nd^2, respectively, up to a certain small constant. When d is large, our method exhibits the most efficient method. By applying the smallest m to achieve the same estimation accuracy as the other methods, our DACE incurs the least computational cost.

D. Analysis on Multiclass Classification

We turn to multiclass classification problem rather than image recovery [5] because multiclass classification is popular in many real-world applications whose performance heavily depends on the estimated covariance of each individual class.

1) Multiclass Classification Solution: Following the setup in [32], we are given data from T classes and compute the corresponding covariance matrix for each class, denoted by $\{\mathbf{C}_t\}_{t=1}^T$. Let $\prod_{k,t} = \sum_{j=1}^k \mathbf{u}_{j,t}\mathbf{u}_{j,t}^T$, where $\{\mathbf{u}_{j,t}\}_{j=1}^k$ are the leading k eigenvectors of $\mathbf{C}_t$ corresponding to the principal components. For a test data $\mathbf{x}$, we fix k and predict the class label by $\max_t \mathbf{x}^T\prod_{k,t}\mathbf{x}$. If the mean vector in each class is zero, then the class label will purely be determined by the class covariance matrices rather than the mean vectors.

2) Analysis: Let the target covariance matrix $\mathbf{C}_t$ in the tth class be calculated from $\{\mathbf{x}_{i,t}\}_{i=1}^{n_t}$; by Corollary 1, we derive the error of our estimator $\mathbf{C}_{e,t}$ as follows:

$$\frac{\|\mathbf{C}_{e,t} - \mathbf{C}_t\|_2}{\|\mathbf{C}_t\|_2} \le \frac{1}{\|\mathbf{C}_t\|_2}\min\{A, B\} \tag{9}$$

where $A = \widetilde{O}(f_t + (\tau^2\varphi/m)((1/n_t))^{1/2} + \tau^2((1/n_t m))^{1/2})$, $B = \widetilde{O}(f_t + (\tau\varphi/m)((d\|\mathbf{C}_t\|_2/n_t))^{1/2} + \tau((d\|\mathbf{C}_t\|_2/n_t m))^{1/2})$,

and $f_t = (\tau^2/n_t) + (\tau^2\varphi^2/n_t m) + \tau\varphi((\|\mathbf{C}_t\|_2/n_t m))^{1/2}$, $(\|\mathbf{x}_{i,t}\|_1/\|\mathbf{x}_{i,t}\|_2) \le \varphi$ with $1 \le \varphi \le \sqrt{d}$, and $\|\mathbf{x}_{i,t}\|_2 \le \tau$ for all $i \in [n_t]$.

Compared with the estimator $\mathbf{C}_e$ obtained from the entire data consisting of $n = \sum_{t=1}^{T} n_t$ data points, the error $\mathbf{C}_{e,t}$ is dominated by $1/\sqrt{n_t}$ and $1/(\|\mathbf{C}_t\|_2)^{1/2}$ (not $1/\|\mathbf{C}_t\|_2$). Suppose n_t is the same for all classes, the term $1/\sqrt{n_t}$ in (9) becomes $\sqrt{T/n}$, which is $\sqrt{T}$ times as large as $1/\sqrt{n}$ in (5) of Corollary 1. Meanwhile, if all $\mathbf{C}_t$ have very similar principal components as those of $\mathbf{C}$ (i.e., all $\{\mathbf{C}_t\}_{t=1}^{T}$ and $\mathbf{C}$ are calculated from the same data distributions, and thus, the data tend to be difficult to classify), then $1/(\|\mathbf{C}_t\|_2)^{1/2}$ nearly equals $1/(\|\mathbf{C}\|_2)^{1/2}$. If all $\mathbf{C}_t$ have totally different principal components from each other, $1/(\|\mathbf{C}_t\|_2)^{1/2}$ will become $1/\sqrt{T}$ times as large as $1/(\|\mathbf{C}\|_2)^{1/2}$.

Thus, compared with the estimation error derived in Corollary 1 over n data with T classes, the estimation error for each class covariance estimator roughly increases with $O(\sqrt{T})$ if all class covariance matrices yield nearly the same leading principal components while the error remaining nearly the same if all class covariance matrices have totally different leading principal components.

Remark 5: The above result does not contradict with our statement in Theorem 2 or Corollary 1 that the estimation error approaches zero (i.e., decreases with the number of data) when receiving more data, because the data follow only one category of distribution result in a stable $1/\|\mathbf{C}\|_2)^{1/2}$. When the data follow different distributions, they practically come randomly and yield a stable $1/\|\mathbf{C}\|_2)^{1/2}$.

Remark 6: In Section III-D1, the performance is determined only by $\mathbf{C}_{e,t}$ or the principal components $\prod_{k,t}$ of $\mathbf{C}_{e,t}$. Via Corollaries 1 and 3, the compressed data obtained by our method guarantee a superior approximation for $\mathbf{C}_{e,t}$ and $\prod_{k,t}$ and yield a comparable classification performance compared with $\mathbf{C}_t$. In other words, if each n_t or the target compressed dimension m is not too small, the distances between individual class estimators will approach to those among the original class covariance matrices with high probability and guarantee the classification performance.

IV. Experiments

In this section, we conduct empirical evaluation to address the following issues.

1) How the dimension, the data size, and the compression ratio affect the estimation precision of our DACE?
2) What performance of our DACE can be attained in real-world data sets?
3) What performance of our DACE can be attained in multiclass classification problems?

To provide fair comparisons, we compare our DACE with three representative algorithms: Gauss-Inverse [39], Sparse [7], and UniSample-HD [6]. In our DACE, the hyper-parameter α is empirically set to 0.9 due to good empirical performance. The hyper-parameter settings in Gauss-Inverse, Sparse, and UniSample-HD simply follow the original work in [6], [7], and [39].

All algorithms are implemented in C++ and run in a single thread mode on a standard workstation with Intel CPU at 2.90 GHz and 128-GB RAM to record the time consumption measured by FLOPS.

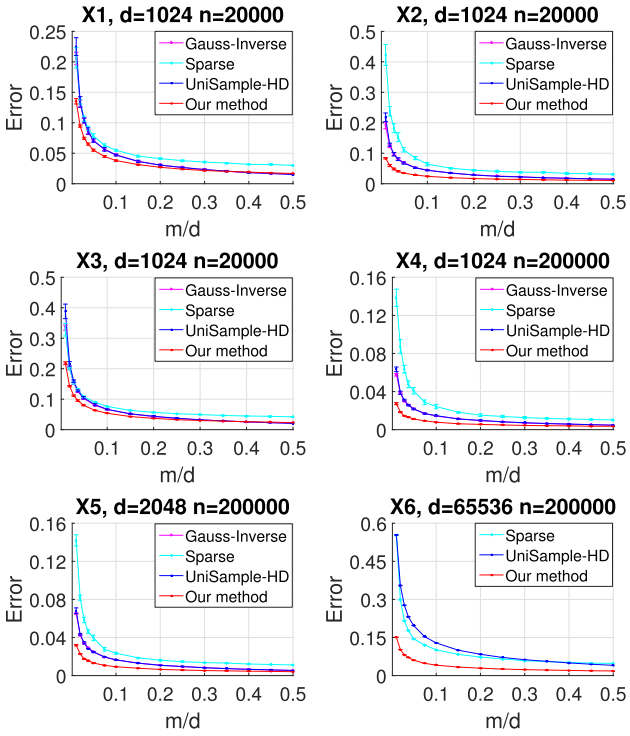

Fig. 1. Accuracy comparisons of covariance matrix estimation on synthetic data sets. The estimation error is measured by $\|\mathbf{C}_e - \mathbf{C}\|_2/\|\mathbf{C}\|_2$ with $\mathbf{C}_e$ calculated by all compared methods and cf $= m/d$ is the compression ratio.

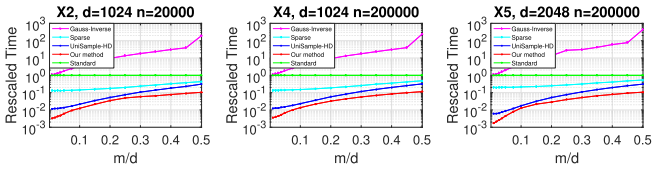

Fig. 2. Rescaled time cost (plotted in the log scale) of covariance matrix estimation on synthetic data sets. The time is normalized to the Standard way of calculating $\mathbf{C} = \mathbf{X}\mathbf{X}^T/n$ on the original data.

A. Covariance Estimation on Synthetic Data Sets

Following the generation procedure [35], we construct six synthetic data sets: 1) $\{\mathbf{X}_i\}_{i=1}^{3}$, $d = 1024$ and $n = 20\,000$; 2) $\mathbf{X}_4$, $d = 1024$ and $n = 200\,000$; and 3) $\mathbf{X}_5$, $d = 2048$, $200\,000$, and $\mathbf{X}_6$, $d = 65536$ and $n = 200\,000$. More specifically, $\mathbf{X} = \mathbf{UFG}$, where $\mathbf{U} \in \mathbb{R}^{d\times k}$ $(\mathbf{U}^T\mathbf{U} = \mathbf{I}_k,\ k \le d)$ defines the signal column space, the square diagonal matrix $\mathbf{F} \in \mathbb{R}^{k\times k}$ contains the diagonal entries $f_{ii} = 1-(i-1)/k$ with linearly diminishing signal singular values, and $\mathbf{G} \in \mathbb{R}^{k\times n}$ is the Gaussian signal, i.e., $g_{ij} \sim \mathcal{N}(0,1)$. In $\mathbf{X}_1$, $k \approx 0.005d$. $\mathbf{X}_2 = \mathbf{DX}$, where $\mathbf{D}$ is a square diagonal matrix with $d_{ii} = 1/\beta_i$ and integer β_i is uniformly sampled from range 1–15. $\mathbf{X}_3$ is constructed the same way as $\mathbf{X}_1$ except that $\mathbf{F}$ is set by an identity matrix. $\{\mathbf{X}_i\}_{i=4}^{6}$ follow the same generation scheme of $\mathbf{X}_2$ by only setting different n and d values.

Fig. 1 shows the average relative estimation error with its standard deviation on ten runs with respect to the compression

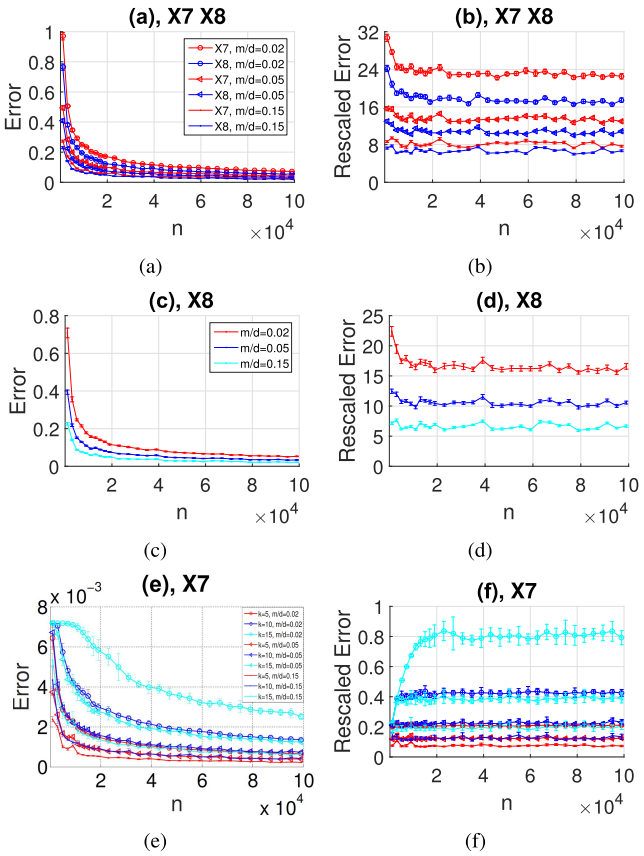

Fig. 3. Convergence analysis of CACE in terms of different n, cf, and k. The legend of (b) is the same as (a). The legend of (d) is the same as (c). The legend of (f) is the same of (e).

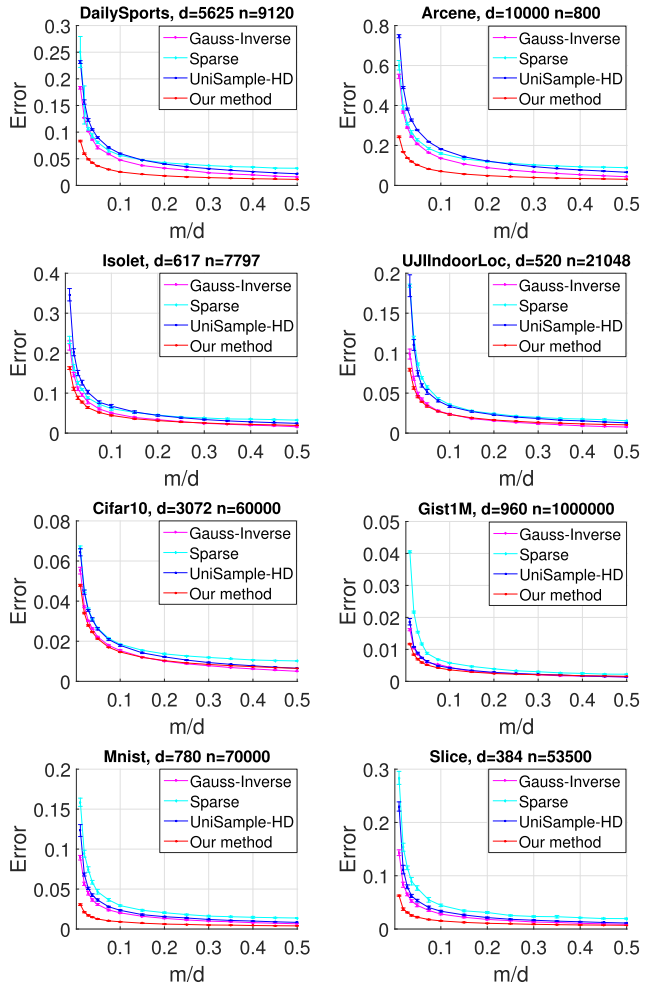

Fig. 4. Accuracy comparisons of covariance matrix estimation on real-world data sets.

ratio cf $= m/d$. Note that the performance of Gauss-Inverse has been revealed on $\mathbf{X}_1$–$\mathbf{X}_5$ and is not provided on $\mathbf{X}_6$ due to enormous computation time. Fig. 2 reports the rescaled time cost in both the compressing and recovering stages. The results show the following.

1) Our DACE exhibits the least estimation error and deviation for all data sets when the dimension d increases. When applying more data, the error decreases gradually. The error decreases dramatically with slightly increasing cf and becomes flat soon. It indicates that our DACE can achieve sufficient estimation accuracy by using substantially fewer data entries than other methods.
2) When the compression ratio cf increases, the estimation error decreases gradually while the time cost increases accordingly. Gauss-Inverse, though good for smaller storage and less communication, consumes significantly much more time than Standard due to the computation of non-sparse projection. Sparse, though without error analysis of the estimator, generally consumes less time than Standard, but performs worse than the other methods. UniSample-HD beats other two methods, but it performs slightly worse than our DACE and consumes more time than DACE. Especially, the inferiority is remarkable when cf is small.
3) In $\mathbf{X}_1$, φ is measured empirically as $0.81\sqrt{d}$, the magnitudes of the data entries are more uniformly distributed, and our DACE can be regarded as uniform sampling with replacement and may perform slightly worse than UniSample-HD and Gauss-Inverse. In $\mathbf{X}_2$, $\varphi = 0.55\sqrt{d}$, the magnitude varies in a moderately larger range, and our DACE outperforms the three other methods significantly. The improvement lies in that our DACE is only sensitive to φ and a smaller φ produces a tighter estimation, which confirms the elaboration in Remarks 3 and 4.
4) The error of each method in $\mathbf{X}_3$ $[\tau/(\|\mathbf{C}\|_2)^{1/2} = 5.5, \varphi = 0.81\sqrt{d}]$ is larger than that in $\mathbf{X}_1$ $[\tau/(\|\mathbf{C}\|_2)^{1/2} = 4.3, \varphi = 0.81\sqrt{d}]$. It is because of that almost all methods are sensitive to $\tau/(\|\mathbf{C}\|_2)^{1/2}$, and the error $\|\mathbf{C}_e - \mathbf{C}\|_2/\|\mathbf{C}\|_2$ increases when $\tau/(\|\mathbf{C}\|_2)^{1/2}$ increases. Such phenomenon is demonstrated via dividing numerous error bounds in Remarks 3 and 4 by $\|\mathbf{C}\|_2$. Our method also achieves the best performance in $\mathbf{X}_4$. Although φ and $\tau/(\|\mathbf{C}\|_2)^{1/2}$ in $\mathbf{X}_4$ are approximately equal to those in $\mathbf{X}_2$, yet the proved error bounds with Remarks 3 and 4 reveal that a larger n in $\mathbf{X}_4$ will lead to smaller estimation errors given the same cf.

To verify the theoretical results in Corollaries 2 and 3, we generate two new synthetic data sets from multivariate normal distribution $\{\mathbf{X}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_{pt})\}_{t=7}^{8} \in \mathbb{R}^{d\times d}$. In $\mathbf{X}_7$,

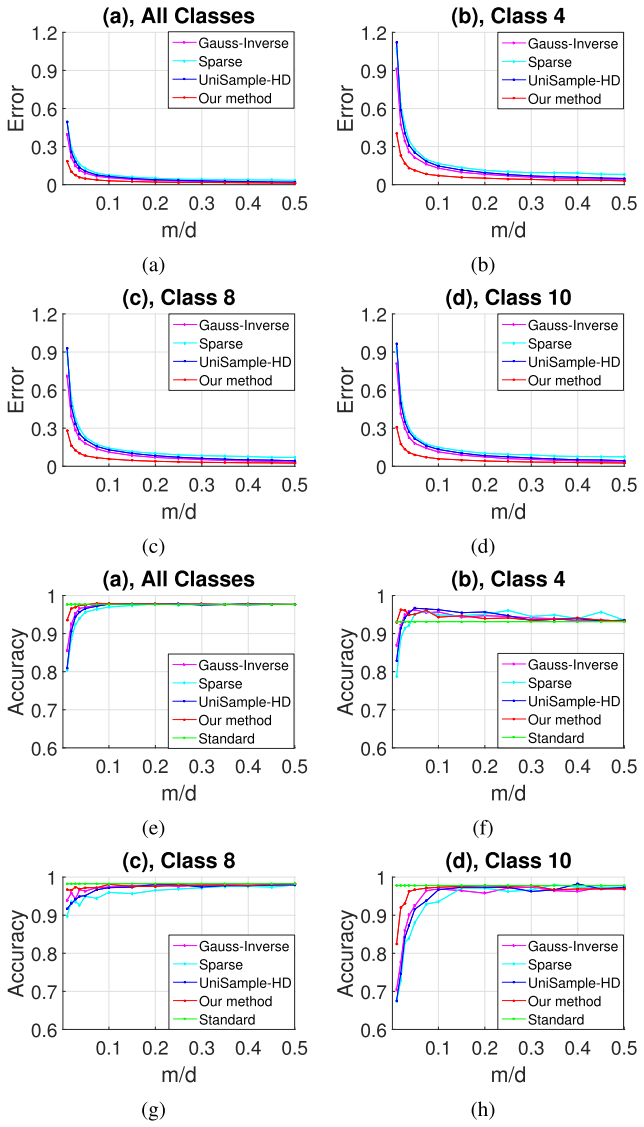

Fig. 5. (a) Covariance estimation error over all data. (b)–(d) Estimation error over the data of three different classes. (e) Classification accuracy averaged over all test data. (f)–(h) Classification accuracy over the test data of three different classes.

the (i, j)th element of $\mathbf{C}_{p7}$ is $0.5^{|i-j|/50}$, while $\mathbf{C}_{p8}$ being a low-rank matrix, which is the solution to $\min_{\text{rank}(\mathbf{A})\leq r} \|\mathbf{A} - \mathbf{C}_{p7}\|_2$. We take $d = 1000$, $r = 5$, $m/d = \{0.02, 0.05, 0.15\}$, and $k = \{5, 10, 15\}$ and vary n from 1000 to 100 000.

Fig. 3(a), (c), and (e) reports the errors defined in the LHS of (6)–(8) under different settings, while Fig. 3(b), (d) and (f) records the errors divided by $1/\sqrt{n}$. The results show that the errors decrease with the increase of n. Especially, the roughly flat curves in Fig. 3(b), (d), and (f) indicate that the error bounds induced by our DACE converge rapidly in the rate of $1/\sqrt{n}$, which coincides with the results in (6)–(8). Fig. 3(a) also exhibits that our DACE can attain more accurate estimation precision from a low-rank generated covariance matrix than that from a high-rank covariance matrix, and enlarging n can improve all the estimation precisions. Fig. 3(e) shows that the estimated errors increase with the increase of k, and these results cohere with (8) by considering the empirical findings that the eigengap $\lambda_k - \lambda_{k+1}$ in $\mathbf{C}_{p7}$ decreases with k.

B. Covariance Estimation on Real-World Data Sets

Fig. 4 reports the estimation errors on eight publicly available real-world data sets and shows that the errors decrease dramatically with the increase of cf. Our DACE consistently exhibits superior accuracy with the least deviation in all cases.

C. Evaluation on Multiclass Classification

To guarantee that the classification performance is purely determined by the class covariance matrices rather than the mean vectors, we generate a new data set, namely MNIST-ZM, which centers the MNIST data set in each class. The data set consists of ten classes of data with around 7000 data points in each class. 100 data points from each class are randomly picked for test while the remaining are applied to calculate $\{\mathbf{C}_t\}_{t=1}^{10}$ and $\{\mathbf{C}_{e,t}\}_{t=1}^{10}$. The parameter k is set to 30 because Standard can obtain good classification results. The ratio m/d is varied from 0.01 to 0.5.

Fig. 5(a), (c), (e), and (g) presents the results of class covariance matrix estimation. We can observe that the estimation error of each individual class covariance matrix $\mathbf{C}_{e,t}$ is around twice (below $\sqrt{T} = \sqrt{10}$) to that of $\mathbf{C}_e$. Accordingly, Fig. 5(d), (d), (f), and (h) shows the compared classification results using the estimated covariance matrices derived from different methods. We observe that the classification accuracy of our DACE is comparable with Standard without performing data compression. Moreover, our DACE slightly outperforms Standard at some m/d value in Fig. 5(f), which depicts that our DACE can select informative features to achieve better generalization. Finally, our DACE also outperforms the other three methods learned from compressed data in terms of both estimation precision and classification accuracy.

V. Conclusion

We present a data-aware weighted sampling method for tackling covariance matrix recovery and multiclass classification problems. We theoretically prove that our proposed DACE is an unbiased covariance matrix estimator and can employ less data than other representative algorithms to attain the same performance. The empirical results on both synthetic and real-world data sets support our theory and demonstrate the superior performance over other state-of-the-art methods.

Appendix A
Preliminaries

Lemmas 1 and 2 and the recited theorems set the foundation of proving our main theoretical results.

Lemma 1: Given a vector $\mathbf{x} \in \mathbb{R}^d$, sample m $(m < d)$ entries from $\mathbf{x}$ with replacement by running Algorithm 1. Let $\{p_k\}_{k=1}^d$ and $\mathbf{S} \in \mathbb{R}^{d\times m}$ define as in Algorithm 1. $\{\mathbf{e}_k \in \mathbb{R}^d\}_{k=1}^d$ denote the standard basis vectors. Then, we have

$$\mathbb{E}[\mathbf{S}\mathbf{S}^T\mathbf{x}\mathbf{x}^T\mathbf{S}\mathbf{S}^T] = \sum_{k=1}^{d} \frac{x_k^2}{mp_k}\mathbf{e}_k\mathbf{e}_k^T + \frac{m-1}{m}\mathbf{x}\mathbf{x}^T \tag{10}$$

$$\mathbb{E}[\mathbb{D}(\mathbf{S}\mathbf{S}^T\mathbf{x}\mathbf{x}^T\mathbf{S}\mathbf{S}^T)] = \sum_{k=1}^{d}\left(\frac{1}{mp_k}+\frac{m-1}{m}\right)x_k^2\mathbf{e}_k\mathbf{e}_k^T \tag{11}$$

$$\begin{aligned}&\mathbb{E}[(\mathbb{D}(\mathbf{S}\mathbf{S}^T\mathbf{x}\mathbf{x}^T\mathbf{S}\mathbf{S}^T))^2]\\ &=\sum_{k=1}^{d}\left[\frac{1}{m^3p_k^3}+\frac{7(m-1)}{m^3p_k^2}+\frac{6(m^2-3m+2)}{m^3p_k}\right.\\ &\qquad\left.+\frac{m^3-6m^2+11m-6}{m^3}\right]x_k^4\mathbf{e}_k\mathbf{e}_k^T \qquad (12)\\ &\quad\times\mathbb{E}[\mathbf{S}\mathbf{S}^T\mathbf{x}\mathbf{x}^T\mathbf{S}\mathbf{S}^T\mathbb{D}(\mathbf{S}\mathbf{S}^T\mathbf{x}\mathbf{x}^T\mathbf{S}\mathbf{S}^T)]\\ &=(\mathbb{E}[\mathbb{D}(\mathbf{S}\mathbf{S}^T\mathbf{x}\mathbf{x}^T\mathbf{S}\mathbf{S}^T)\mathbf{S}\mathbf{S}^T\mathbf{x}\mathbf{x}^T\mathbf{S}\mathbf{S}^T])^T\\ &=\sum_{k=1}^{d}\left[\frac{1}{m^3p_k^3}+\frac{6(m-1)}{m^3p_k^2}+\frac{3(m^2-3m+2)}{m^3p_k}\right]x_k^4\mathbf{e}_k\mathbf{e}_k^T\\ &\quad+\frac{m-1}{m^3}\mathbf{x}\mathbf{x}^T\mathbb{D}\left(\left\{\frac{x_k^2}{p_k^2}\right\}\right)+\frac{3(m^2-3m+2)}{m^3}\mathbf{x}\mathbf{x}^T\\ &\quad\cdot\left[\mathbb{D}\left(\left\{\frac{x_k^2}{p_k}\right\}\right)+\frac{m-3}{3}\mathbb{D}(\{x_k^2\})\right] \qquad (13)\\ &\mathbb{E}[(\mathbf{S}\mathbf{S}^T\mathbf{x}\mathbf{x}^T\mathbf{S}\mathbf{S}^T)^2]\\ &=\sum_{k=1}^{d}\left[\frac{4(m-1)}{m^3p_k^2}+\frac{1}{m^3p_k^3}\right]x_k^4\mathbf{e}_k\mathbf{e}_k^T\\ &\quad+\sum_{k=1}^{d}\left[\frac{\|\mathbf{x}\|_2^2(m^2-3m+2)}{m^3}+\frac{m-1}{m^3}\sum_{k=1}^{d}\frac{x_k^2}{p_k}\right]\frac{x_k^2}{p_k}\mathbf{e}_k\mathbf{e}_k^T\\ &\quad+\left[\frac{\|\mathbf{x}\|_2^2(m^3-6m^2+11m-6)}{m^3}+\frac{m^2-3m+2}{m^3}\sum_{k=1}^{d}\frac{x_k^2}{p_k}\right]\mathbf{x}\mathbf{x}^T\\ &\quad+\mathbf{x}\mathbf{x}^T\left[\frac{2(m^2-3m+2)}{m^3}\mathbb{D}\left(\left\{\frac{x_k^2}{p_k}\right\}\right)\right.\\ &\qquad\left.+\frac{m-1}{m^3}\mathbb{D}\left(\left\{\frac{x_k^2}{p_k^2}\right\}\right)\right]\\ &\quad+\left[\frac{2(m^2-3m+2)}{m^3}\mathbb{D}\left(\left\{\frac{x_k^2}{p_k}\right\}\right)+\frac{m-1}{m^3}\mathbb{D}\left(\left\{\frac{x_k^2}{p_k^2}\right\}\right)\right]\mathbf{x}\mathbf{x}^T\end{aligned} \tag{14}$$

where $\mathbb{D}(\{x_k^2\})$ denotes a square diagonal matrix with $\{x_k^2\}_{k=1}^d$ on its diagonal and likewise for other notations.

Lemma 2: Following the same notations defined in Lemma 1, with probability at least $1-\sum_{k=1}^{d}\eta_k$, we have:

$$\|\mathbf{S}\mathbf{S}^T\mathbf{x}\mathbf{x}\mathbf{S}\mathbf{S}^T\|_2 \le \sum_{k\in\Gamma} f^2(x_k,\eta_k,m) \tag{15}$$

where Γ is a set containing at most m different elements of $[d]$ with its cardinality $|\Gamma|\le m$ and $f(x_k,\eta_k,m) = |x_k| + \log((2/\eta_k))[(|x_k|/3mp_k) + |x_k|((1/9m^2p_k^2) + (2/\log(2/\eta_k))((1/mp_k)-(1/m)))^{1/2}]$.

Theorem 3 ([44, p. 76]): Let $\{\mathbf{A}_i\}_{i=1}^L \in \mathbb{R}^{d\times n}$ be independent random matrices with $\mathbb{E}[\mathbf{A}_i]=0$ and $\|\mathbf{A}_i\|_2 \le R$. Define the variance $\sigma^2 = \max\{\|\sum_{i=1}^{L}\mathbb{E}[\mathbf{A}_i\mathbf{A}_i^T]\|_2, \|\sum_{i=1}^{L}\mathbb{E}[\mathbf{A}_i^T\mathbf{A}_i]\|_2\}$. Then, $\mathbb{P}(\|\sum_{i=1}^{L}\mathbf{A}_i\|_2\ge\epsilon)\le(d+n)\exp((-\epsilon^2/2/\sigma^2+R\epsilon/3))$ for all $\epsilon\ge 0$.

Theorem 4 ([25, p. 396]): If $\mathbf{A}\in\mathbb{R}^{d\times d}$ and $\mathbf{A}+\mathbf{E}\in\mathbb{R}^{d\times d}$ are symmetric matrices, then

$$\lambda_k(\mathbf{A})+\lambda_d(\mathbf{E})\le\lambda_k(\mathbf{A}+\mathbf{E})\le\lambda_k(\mathbf{A})+\lambda_1(\mathbf{E}) \tag{16}$$

for $k\in[d]$, where $\lambda_k(\mathbf{A}+\mathbf{E})$ and $\lambda_k(\mathbf{A})$ designate the kth largest eigenvalues.

APPENDIX B
THEORETICAL PROOFS

A. Proof of Lemma 2

Proof: According to the notion defined in Lemma 1, we have

$$\begin{aligned}&\|\mathbf{S}\mathbf{S}^T\mathbf{x}\mathbf{x}^T\mathbf{S}\mathbf{S}^T\|_2\\ &\overset{(a)}{=}\|\mathbf{S}\mathbf{S}^T\mathbf{x}\|_2^2=\left\|\sum_{j=1}^{m}\mathbf{s}_{t_j}\mathbf{s}_{t_j}^T\mathbf{x}\right\|_2^2\\ &=\left\|\sum_{j=1}^{m}\frac{1}{mp_{t_j}}x_{t_j}\mathbf{e}_{t_j}\right\|_2^2=\left\|\sum_{j=1}^{m}\sum_{k=1}^{d}\frac{\delta_{t_jk}}{mp_k}x_k\mathbf{e}_k\right\|_2^2\\ &=\sum_{k=1}^{d}\left(\sum_{j=1}^{m}\frac{\delta_{t_jk}x_k}{mp_k}\right)^2\overset{(b)}{=}\sum_{k\in\Gamma}\left(\sum_{j=1}^{m}\frac{\delta_{t_jk}x_k}{mp_k}\right)^2\end{aligned} \tag{17}$$

where $\Gamma=\{\gamma_t\}_{t=1}^{|\Gamma|}$ is a set with the cardinality $|\Gamma|\le m$ containing at most m different elements of $[d]$.

In (17), (a) holds because $\mathbf{S}\mathbf{S}^T\mathbf{x}\mathbf{x}^T\mathbf{S}\mathbf{S}^T$ is a positive semi-definite matrix of rank 1. δ_{t_jk} returns 1 only when $t_j=k$ and 0 otherwise. $\mathbb{P}(\delta_{t_jk}=1)=\mathbb{P}(t_j=k)=p_k$. (b) holds because we perform random sampling with replacement m times on the d entries of $\mathbf{x}\in\mathbb{R}^d$, and consequently, at most m different entries from $\mathbf{x}$ are sampled.

Let $k=\gamma_1,\gamma_1\in\Gamma$, and we first bound $|\sum_{j=1}^{m}(\delta_{t_j\gamma_1}x_{\gamma_1}/mp_{\gamma_1})|$. Let $a_j=(\delta_{t_j\gamma_1}x_{\gamma_1}/mp_{\gamma_1})-(x_{\gamma_1}/m)$, $j\in[m]$, and we can easily check that $\{a_j\}_{j=1}^m$ are independent with $\mathbb{E}[a_j]=0$, where Theorem 3 can be applied for our analysis. Furthermore, we have

$$\max_{j\in[m]}|a_j|=\max\left\{\frac{|x_{\gamma_1}|}{m}\left(\frac{1}{p_{\gamma_1}}-1\right),\frac{|x_{\gamma_1}|}{m}\right\}\le\frac{|x_{\gamma_1}|}{mp_{\gamma_1}}(:=R) \tag{18}$$

$$\text{and}\quad\sum_{\mathrm{j}=1}^{\mathrm{m}}\mathbb{E}\left[\mathrm{a}_{\mathrm{j}}^2\right]=\frac{x_{\gamma_1}^2}{mp_{\gamma_1}}-\frac{x_{\gamma_1}^2}{m}(=\sigma^2). \tag{19}$$

Thus, by applying Theorem 3, we obtain $\mathbb{P}(|\sum_{j=1}^{m}a_j|\ge\epsilon)\le\eta_{\gamma_1}$, where $\eta_{\gamma_1}=2\exp((-\epsilon^2/2/x_{\gamma_1}^2/(mp_{\gamma_1})-x_{\gamma_1}^2/m+|x_{\gamma_1}|\epsilon/(3mp_{\gamma_1})))$. Then, with probability at least $1-\eta_{\gamma_1}$, we have $|\sum_{j=1}^{m}a_j|\le\epsilon$, i.e., $|\sum_{j=1}^{m}(\delta_{t_j\gamma_1}x_{\gamma_1}/mp_1)|\le|x_{\gamma_1}|+\epsilon$. We then replace ϵ and obtain the function, $f(x_{\gamma_1},\eta_{\gamma_1},m)$, defined in Lemma 4.

Similarly, we bound $|\sum_{j=1}^{m}(\delta_{t_jk}x_k/mp_k)|$ for $k\in[d]$. The lemma holds by using the union bound over cases for all $k\in[d]$. □

B. Proof of Theorem 1

Proof: First, we have

$$\begin{aligned}\mathbb{E}[\widehat{\mathbf{C}}_1] &= \frac{m}{nm-n}\mathbb{E}\sum_{i=1}^{n}\mathbf{S}_i\mathbf{S}_i^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{S}_i\mathbf{S}_i^T \\ &= \overset{\text{by (10)}}{} \frac{m}{nm-n}\sum_{i=1}^{n}\left[\sum_{k=1}^{d}\frac{x_{ki}^2}{mp_{ki}}\mathbf{e}_k\mathbf{e}_k^T + \frac{m-1}{m}\mathbf{x}_i\mathbf{x}_i^T\right] \\ &= \frac{1}{nm-n}\sum_{i=1}^{n}\sum_{k=1}^{d}\frac{x_{ki}^2}{p_{ki}}\mathbf{e}_k\mathbf{e}_k^T + \frac{1}{n}\mathbf{X}\mathbf{X}^T. \end{aligned} \tag{20}$$

Second, by (11) in Lemma 1, we have

$$\begin{aligned}\mathbb{E}[\widehat{\mathbf{C}}_2] &= \frac{m}{nm-n}\sum_{i=1}^{n}\mathbb{E}\left[\mathbb{D}\left(\mathbf{S}_i\mathbf{S}_i^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{S}_i\mathbf{S}_i^T\right)\right]\mathbb{D}(\mathbf{b}_i) \\ &= \frac{1}{nm-n}\sum_{i=1}^{n}\sum_{k=1}^{d}\frac{x_{ki}^2}{p_{ki}}\mathbf{e}_k\mathbf{e}_k^T. \end{aligned} \tag{21}$$

Hence, by (20) and (21), we immediately conclude that $\mathbf{C}_e = \widehat{\mathbf{C}}_1 - \widehat{\mathbf{C}}_2$ is unbiased for $\mathbf{C}$. □

C. Proof of Theorem 2

Proof: For simplicity, we define $\mathbf{A}_i = \mathbf{A}_{i_1} - \mathbf{A}_{i_2} - \mathbf{A}_{i_3}$, where $\mathbf{A}_{i_1} = (m\mathbf{S}_i\mathbf{S}_i^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{S}_i\mathbf{S}_i^T/nm - n)$, $\mathbf{A}_{i_2} = (m\mathbb{D}(\mathbf{S}_i\mathbf{S}_i^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{S}_i\mathbf{S}_i^T)\mathbb{D}(\mathbf{b}_i)/nm - n)$, and $\mathbf{A}_{i_3} = (\mathbf{x}_i\mathbf{x}_i^T/n)$. Then, $\mathbf{C}_e - \mathbf{C} = \sum_{i=1}^{n}\mathbf{A}_i$.

Obviously, $\{\mathbf{A}_i\}_{i=1}^n$ are independent zero-mean random matrices. Hence, Theorem 3 can be directly applied. To bound $\|\mathbf{C}_e - \mathbf{C}\|_2$, we then calculate the corresponding parameters R and σ^2 that characterize the range and variance of $\mathbf{A}_i$, respectively.

We first derive R, i.e., the bound of $\|\mathbf{A}_i\|_2$ for $i \in [n]$. By expanding $\|\mathbf{A}_i\|_2$, we get

$$\begin{aligned}\|\mathbf{A}_i\|_2 &= \|\mathbf{A}_{i_1} - \mathbf{A}_{i_2} - \mathbf{A}_{i_3}\|_2 \le \|\mathbf{A}_{i_1} - \mathbf{A}_{i_2}\|_2 + \|\mathbf{A}_{i_3}\|_2 \\ &\le \|\mathbf{A}_{i_1}\|_2 + \|\mathbf{A}_{i_3}\|_2. \end{aligned} \tag{22}$$

The last inequality in (22) results from

$$\begin{aligned}&\|\mathbf{A}_{i_1} - \mathbf{A}_{i_2}\|_2 \\ &\quad= \max_{k\in[d]}|\lambda_k(\mathbf{A}_{i_1} - \mathbf{A}_{i_2})| \\ &\quad\overset{(a)}{\le} \max\{|\lambda_d(\mathbf{A}_{i_1}) - \lambda_1(\mathbf{A}_{i_2})|, |\lambda_1(\mathbf{A}_{i_1}) - \lambda_d(\mathbf{A}_{i_2})|\} \\ &\quad\overset{(b)}{=} \max\{\lambda_1(\mathbf{A}_{i_2}), |\lambda_1(\mathbf{A}_{i_1}) - \lambda_d(\mathbf{A}_{i_2})|\} \\ &\quad\overset{(c)}{=} \max\{\lambda_1(\mathbf{A}_{i_2}), \lambda_1(\mathbf{A}_{i_1}) - \lambda_d(\mathbf{A}_{i_2})\} \\ &\quad\overset{(d)}{\le} \lambda_1(\mathbf{A}_{i_1}) \overset{(e)}{=} \|\mathbf{A}_{i_1}\|_2\end{aligned}$$

where $\lambda_k(\cdot)$ is the kth largest eigenvalue. The inequality (a) holds because $\lambda_k(\mathbf{A}_{i_1}) - \lambda_1(\mathbf{A}_{i_2}) \le \lambda_k(\mathbf{A}_{i_1} - \mathbf{A}_{i_2}) \le \lambda_k(\mathbf{A}_{i_1}) - \lambda_d(\mathbf{A}_{i_2})$ for any $k \in [d]$, which is attained by applying Theorem 4 with the fact that $\lambda_d(-\mathbf{A}_{i_2}) = -\lambda_1(\mathbf{A}_{i_2})$ and $\lambda_1(-\mathbf{A}_{i_2}) = -\lambda_d(\mathbf{A}_{i_2})$ for $\mathbf{A}_{i_2} \in \mathbb{R}^{d\times d}$. The equality (b) holds because $\lambda_{k\ge 2}(\mathbf{A}_{i_1}) = 0$ from the fact that $\mathbf{A}_{i_1}$ is a positive semidefinite matrix of rank 1 and $\lambda_{k\in[d]}(\mathbf{A}_{i_2}) \ge 0$ since $\mathbf{A}_{i_2}$ is positive semidefinite. The equality (c) follows the fact that $\lambda_1(\mathbf{A}_{i_1}) = \mathrm{Tr}(\mathbf{A}_{i_1}) \ge \mathrm{Tr}(\mathbf{A}_{i_2}) = \sum_{k=1}^{d}\lambda_k(\mathbf{A}_{i_2}) \ge \lambda_d(\mathbf{A}_{i_2}) \ge 0$, where the first equality holds because $\lambda_{k\ge 2}(\mathbf{A}_{i_1}) = 0$, the first inequality results from the fact that the diagonal matrix $\mathbf{A}_{i_2}$ is constructed by the diagonal elements of $\mathbf{A}_{i_1}$ multiplied by positive scalars not bigger than 1, and the second inequality is the consequence of $\lambda_{k\in[d]}(\mathbf{A}_{i_2}) \ge 0$. The equality (d) results from that $\lambda_{k\in[d]}(\mathbf{A}_{i_2}) \ge 0$. The equality (e) holds due to the fact that $\mathbf{A}_{i_1}$ is positive semidefinite.

Now, we only need to bound $\|\mathbf{A}_{i_1}\|_2$ and $\|\mathbf{A}_{i_3}\|_2$. We have

$$\|\mathbf{A}_{i_3}\|_2 = \left\|\frac{\mathbf{x}_i\mathbf{x}_i^T}{n}\right\|_2 = \frac{\|\mathbf{x}_i\|_2^2}{n}. \tag{23}$$

Applying Lemma 2 gets with probability at least $1 - \sum_{k=1}^{d}\eta_{ki}$

$$\|\mathbf{A}_{i_1}\|_2 \le \frac{m}{nm-n}\sum_{k\in\Gamma_i} f^2(x_{ki}, \eta_{ki}, m) \tag{24}$$

where $\Gamma_i = \{\gamma_{ti}\}_{t=1}^{|\Gamma_i|}$ is a set occupying at most m different elements of $[d]$ with its cardinality $|\Gamma_i| \le m$ and $f(x_{ki}, \eta_{ki}, m) = |x_{ki}| + \log((2/\eta_{ki}))[(|x_{ki}|/3mp_{ki}) + |x_{ki}|((1/9m^2p_{ki}^2) + (2/\log(2/\eta_{ki}))((1/mp_{ki}) - (1/m)))^{1/2}]$.

We can derive similar bounds for all $\{\mathbf{x}_i\}_{i=1}^n$. Then, by applying the union bound, with probability at least $1 - \sum_{i=1}^{n}\sum_{k=1}^{d}\eta_{ki}$, we have

$$R = \max_{i\in[n]}\left[\frac{m}{nm-n}\sum_{k\in\Gamma_i} f^2(x_{ki}, \eta_{ki}, m) + \frac{\|\mathbf{x}_i\|_2^2}{n}\right]. \tag{25}$$

Applying the inequality $(\sum_{t=1}^{n}a_t)^2 \le n\sum_{t=1}^{n}a_t^2$, we have

$$\begin{aligned}&f^2(x_{ki}, \eta_{ki}, m) \\ &\le 3x_{ki}^2 + 3\log^2\left(\frac{2}{\eta_{ki}}\right)\frac{x_{ki}^2}{9m^2p_{ki}^2} \\ &\quad + 3\log^2\left(\frac{2}{\eta_{ki}}\right)\frac{x_{ki}^2}{9m^2p_{ki}^2} + 6\log\left(\frac{2}{\eta_{ki}}\right)\left(\frac{x_{ki}^2}{mp_{ki}} - \frac{x_{ki}^2}{m}\right) \\ &\le 3x_{ki}^2 + \log^2\left(\frac{2}{\eta_{ki}}\right)\frac{2x_{ki}^2}{3m^2p_{ki}^2} + \log\left(\frac{2}{\eta_{ki}}\right)\frac{6x_{ki}^2}{mp_{ki}}. \end{aligned} \tag{26}$$

Before continuing characterizing R in (25), we set the sampling probabilities as $p_{ki} = \alpha(|x_{ki}|/\|\mathbf{x}_i\|_1) + (1-\alpha)(x_{ki}^2/\|\mathbf{x}_i\|_2^2)$. It is easy to check that $\sum_{k=1}^{d}p_{ki} = 1$. For $0 < \alpha < 1$, we also have $p_{ki} \ge \alpha(|x_{ki}|/\|\mathbf{x}_i\|_1)$, then plugging it in the second and the third term of (26), respectively, we get

$$\begin{aligned}f^2(x_{ki}, \eta_{ki}, m) &\le U_{ki} \\ U_{ki} &:= 3x_{ki}^2 + \log^2\left(\frac{2}{\eta_{ki}}\right)\frac{2\|\mathbf{x}_i\|_1^2}{3m^2\alpha^2} \\ &\quad + \log\left(\frac{2}{\eta_{ki}}\right)\frac{6|x_{ki}|\|\mathbf{x}_i\|_1}{m\alpha}. \end{aligned} \tag{27}$$

Equipped with (25) and setting $\eta_{ki} = (\eta/nd)$ for all $i \in [n]$ and $k \in [d]$, we bound R with probability at least $1 - \sum_{i=1}^{n}\sum_{k=1}^{d}\eta_{ki} = 1 - \eta$ by

$$R \le \max_{i\in[n]}\left[\frac{m}{nm-n}\sum_{k\in\Gamma_i}U_{ki} + \frac{\|\mathbf{x}_i\|_2^2}{n}\right]$$

$$\leq \max_{i\in[n]}\left[\frac{2}{n}\left(3\|\mathbf{x}_i\|_2^2+\log^2\left(\frac{2nd}{\eta}\right)\frac{2\|\mathbf{x}_i\|_1^2}{3m\alpha^2}+\log\left(\frac{2nd}{\eta}\right)\frac{6\|\mathbf{x}_i\|_1^2}{m\alpha}\right)+\frac{\|\mathbf{x}_i\|_2^2}{n}\right]$$
$$\leq \max_{i\in[n]}\left[\frac{7\|\mathbf{x}_i\|_2^2}{n}+\log^2(\tfrac{2nd}{\eta})\frac{14\|\mathbf{x}_i\|_1^2}{nm\alpha^2}\right] \tag{28}$$

where the second inequality follows from that $(m/m-1)\leq 2$ for $m\geq 2$ and $|\Gamma_i|\leq m$ and the last inequality results from that $\alpha\leq 1$ and $\log((2nd/\eta))\geq 1$ for $n\geq 1$, $d\geq 2$, and $\eta\leq 1$.

We then derive σ^2 by only bounding for $\|\sum_{i=1}^n \mathbb{E}[\mathbf{A}_i\mathbf{A}_i]\|_2$ since $\mathbf{A}_i$ is symmetric. By expanding $\mathbb{E}[\mathbf{A}_i\mathbf{A}_i]$, we obtain

$$\begin{aligned}\mathbf{0}&\preceq \mathbb{E}[\mathbf{A}_i\mathbf{A}_i]\\&=\mathbb{E}\big[\mathbf{A}_{i_1}\mathbf{A}_{i_1}+\mathbf{A}_{i_2}\mathbf{A}_{i_2}+\mathbf{A}_{i_3}\mathbf{A}_{i_3}-\mathbf{A}_{i_1}\mathbf{A}_{i_2}\\&\quad-\mathbf{A}_{i_2}\mathbf{A}_{i_1}-\mathbf{A}_{i_1}\mathbf{A}_{i_3}-\mathbf{A}_{i_3}\mathbf{A}_{i_1}+\mathbf{A}_{i_2}\mathbf{A}_{i_3}+\mathbf{A}_{i_3}\mathbf{A}_{i_2}\big].\end{aligned}$$

In the following, we bound the expectation of each term. Specifically, invoking Lemma 1, we have

$$n^2\mathbb{E}[\mathbf{A}_i\mathbf{A}_i]=\sum_{i=1}^{11}\textcircled{i}-\sum_{i=12}^{22}\textcircled{i},\quad\text{where} \tag{29}$$

$$\textcircled{1}:=\sum_{k=1}^d\left[\frac{4}{m(m-1)p_{ki}^2}+\frac{1}{(m-1)^2mp_{ki}^3}\right]x_{ki}^4\mathbf{e}_k\mathbf{e}_k^T$$
$$\textcircled{2}:=\sum_{k=1}^d\left[\frac{\|\mathbf{x}_i\|_2^2(m-2)}{m(m-1)}+\frac{1}{m(m-1)}\sum_{k=1}^d\frac{x_{ki}^2}{p_{ki}}\right]\frac{x_{ki}^2}{p_{ki}}\mathbf{e}_k\mathbf{e}_k^T$$
$$\textcircled{3}:=\left[\frac{\|\mathbf{x}_i\|_2^2(m^2-5m+6)}{m(m-1)}+\frac{m-2}{m(m-1)}\sum_{k=1}^d\frac{x_{ki}^2}{p_{ki}}\right]\mathbf{x}_i\mathbf{x}_i^T$$
$$\textcircled{4}:=\frac{2(m-2)}{m(m-1)}\mathbf{x}_i\mathbf{x}_i^T\mathbb{D}\left(\left\{\frac{x_{ki}^2}{p_{ki}}\right\}\right)$$
$$\textcircled{5}:=\frac{1}{m(m-1)}\mathbf{x}_i\mathbf{x}_i^T\mathbb{D}\left(\left\{\frac{x_{ki}^2}{p_{ki}^2}\right\}\right)$$
$$\textcircled{6}:=\frac{2(m-2)}{m(m-1)}\mathbb{D}\left(\left\{\frac{x_{ki}^2}{p_{ki}}\right\}\right)\mathbf{x}_i\mathbf{x}_i^T$$
$$\textcircled{7}:=\frac{1}{m(m-1)}\mathbb{D}\left(\left\{\frac{x_{ki}^2}{p_{ki}^2}\right\}\right)\mathbf{x}_i\mathbf{x}_i^T$$
$$\textcircled{8}:=\mathbb{D}(\mathbf{b}_i)\mathbb{D}(\mathbf{b}_i)\sum_{k=1}^d\left[\frac{1}{m(m-1)^2p_{ki}^3}+\frac{7}{m(m-1)p_{ki}^2}+\frac{6(m-2)}{m(m-1)p_{ki}}+\frac{(m-2)(m-3)}{m(m-1)}\right]\times x_{ki}^4\mathbf{e}_k\mathbf{e}_k^T$$
$$\textcircled{9}:=\|\mathbf{x}_i\|_2^2\mathbf{x}_i\mathbf{x}_i^T$$
$$\textcircled{10}:=\sum_{k=1}^d\left(\frac{1}{(m-1)p_{ki}}+1\right)x_{ki}^2\mathbf{e}_k\mathbf{e}_k^T\mathbb{D}(\mathbf{b}_i)\mathbf{x}_i\mathbf{x}_i^T$$
$$\textcircled{11}:=\mathbf{x}_i\mathbf{x}_i^T\sum_{k=1}^d\left(\frac{1}{(m-1)p_{ki}}+1\right)\times x_{ki}^2\mathbf{e}_k\mathbf{e}_k^T\mathbb{D}(\mathbf{b}_i)$$
$$\textcircled{12}:=2\sum_{k=1}^d\left[\frac{1}{m(m-1)^2p_{ki}^3}+\frac{6}{m(m-1)p_{ki}^2}+\frac{3(m-2)}{m(m-1)p_{ki}}\right]\times x_{ki}^4\mathbf{e}_k\mathbf{e}_k^T\mathbb{D}(\mathbf{b}_i)$$
$$\textcircled{13}:=\frac{3(m-2)}{m(m-1)}\mathbf{x}_i\mathbf{x}_i^T\mathbb{D}\left(\left\{\frac{x_{ki}^2}{p_{ki}}\right\}\right)\mathbb{D}(\mathbf{b}_i)$$
$$\textcircled{14}:=\frac{(m-2)(m-3)}{m(m-1)}\mathbf{x}_i\mathbf{x}_i^T\mathbb{D}(\{x_{ki}^2\})\mathbb{D}(\mathbf{b}_i)$$
$$\textcircled{15}:=\frac{3(m-2)}{m(m-1)}\mathbb{D}(\mathbf{b}_i)\mathbb{D}\left(\left\{\frac{x_{ki}^2}{p_{ki}}\right\}\right)\mathbf{x}_i\mathbf{x}_i^T$$
$$\textcircled{16}:=\frac{(m-2)(m-3)}{m(m-1)}\mathbb{D}(\mathbf{b}_i)\mathbb{D}(\{x_{ki}^2\})\mathbf{x}_i\mathbf{x}_i^T$$
$$\textcircled{17}:=\sum_{k=1}^d\frac{x_{ki}^2}{(m-1)p_{ki}}\mathbf{e}_k\mathbf{e}_k^T\mathbf{x}_i\mathbf{x}_i^T,\quad\textcircled{18}:=\|\mathbf{x}_i\|_2^2\mathbf{x}_i\mathbf{x}_i^T$$
$$\textcircled{19}:=\sum_{k=1}^d\frac{x_{ki}^2}{(m-1)p_{ki}}\mathbf{x}_i\mathbf{x}_i^T\mathbf{e}_k\mathbf{e}_k^T,\quad\textcircled{20}:=\|\mathbf{x}_i\|_2^2\mathbf{x}_i\mathbf{x}_i^T$$
$$\textcircled{21}:=\frac{1}{m(m-1)}\mathbf{x}_i\mathbf{x}_i^T\mathbb{D}\left(\left\{\frac{x_{ki}^2}{p_{ki}^2}\right\}\right)\mathbb{D}(\mathbf{b}_i)$$
$$\textcircled{22}:=\frac{1}{m(m-1)}\mathbb{D}(\mathbf{b}_i)\mathbb{D}\left(\left\{\frac{x_{ki}^2}{p_{ki}^2}\right\}\right)\mathbf{x}_i\mathbf{x}_i^T.$$

In Eq. (29), for $m\geq 2$, we have

$$\textcircled{10}-\textcircled{17}=\mathbf{0},\quad\textcircled{11}-\textcircled{19}=\mathbf{0}$$
$$\textcircled{4}-\textcircled{13}+\textcircled{5}-\textcircled{14}-\textcircled{21}=\frac{\mathbf{x}_i\mathbf{x}_i^T}{m(m-1)}\mathbb{D}\left(\left\{\frac{((m-1)/p_{ki})x_{ki}^2}{1+(m-1)p_{ki}}+\frac{(m-2)(m+1-1/p_{ki})x_{ki}^2}{1+(m-1)p_{ki}}\right\}\right)$$
$$\textcircled{6}-\textcircled{15}+\textcircled{7}-\textcircled{16}-\textcircled{22}=\mathbb{D}\left(\left\{\frac{((m-1)/p_{ki})x_{ki}^2}{1+(m-1)p_{ki}}+\frac{(m-2)(m+1-1/p_{ki})x_{ki}^2}{1+(m-1)p_{ki}}\right\}\right)\times\frac{\mathbf{x}_i\mathbf{x}_i^T}{m(m-1)}$$
$$\textcircled{3}+\textcircled{9}-\textcircled{18}-\textcircled{20}=\left[\frac{(6-4m)\|\mathbf{x}_i\|_2^2}{m^2-m}+\frac{m-2}{m^2-m}\sum_{k=1}^d\frac{x_{ki}^2}{p_{ki}}\right]\mathbf{x}_i\mathbf{x}_i^T\preceq\sum_{k=1}^d\frac{x_{ki}^2}{mp_{ki}}\mathbf{x}_i\mathbf{x}_i^T;$$
$$\textcircled{8}-\textcircled{12}\preceq\mathbf{0}$$
$$\textcircled{1}\preceq\sum_{k=1}^d\left[\frac{8x_{ki}^4}{m^2p_{ki}^2}+\frac{4x_{ki}^4}{m^3p_{ki}^3}\right]\mathbf{e}_k\mathbf{e}_k^T$$
$$\textcircled{2}\preceq\sum_{k=1}^d\left[\frac{\|\mathbf{x}_i\|_2^2x_{ki}^2}{mp_{ki}}+\frac{2x_{ki}^2}{m^2p_{ki}}\sum_{k=1}^d\frac{x_{ki}^2}{p_{ki}}\right]\mathbf{e}_k\mathbf{e}_k^T. \tag{30}$$

Then, by applying (29) and (30), we obtain

$$\mathbf{0}\preceq\mathbb{E}[\mathbf{A}_i\mathbf{A}_i]\preceq\frac{1}{n^2}\sum_{k=1}^d\left[\frac{8x_{ki}^4}{m^2p_{ki}^2}+\frac{4x_{ki}^4}{m^3p_{ki}^3}+\frac{\|\mathbf{x}_i\|_2^2x_{ki}^2}{mp_{ki}}+\frac{2x_{ki}^2}{m^2p_{ki}}\sum_{k=1}^d\frac{x_{ki}^2}{p_{ki}}\right]$$

$$\times \mathbf{e}_k\mathbf{e}_k^T + \frac{\mathbf{x}_i\mathbf{x}_i^T}{n^2m(m-1)} \times \mathbb{D}\left(\left\{\frac{\frac{m-1}{p_{ki}}x_{ki}^2}{1+(m-1)p_{ki}} + \frac{(m-2)\left(m+1-\frac{1}{p_{ki}}\right)x_{ki}^2}{1+(m-1)p_{ki}}\right\}\right) + \mathbb{D}\left(\left\{\frac{((m-1)/p_{ki})x_{ki}^2}{1+(m-1)p_{ki}} + \frac{(m-2)(m+1-1/p_{ki})x_{ki}^2}{1+(m-1)p_{ki}}\right\}\right) \cdot \frac{\mathbf{x}_i\mathbf{x}_i^T}{n^2m(m-1)} + \frac{1}{n^2m}\sum_{k=1}^{d}\frac{x_{ki}^2}{p_{ki}}\mathbf{x}_i\mathbf{x}_i^T. \quad (31)$$

With (31) in hand, we can formulate σ^2 as

$$\begin{aligned}\sigma^2 &= \left\|\sum_{i=1}^{n}\mathbb{E}[\mathbf{A}_i\mathbf{A}_i]\right\|_2 \\ &\le \sum_{i=1}^{n}\max_{k\in[d]}\frac{1}{n^2}\left[\frac{8x_{ki}^4}{m^2p_{ki}^2}+\frac{4x_{ki}^4}{m^3p_{ki}^3}+\frac{\|\mathbf{x}_i\|_2^2x_{ki}^2}{mp_{ki}}+\frac{2x_{ki}^2}{m^2p_{ki}}\sum_{k=1}^{d}\frac{x_{ki}^2}{p_{ki}}\right] \\ &\quad + \sum_{i=1}^{n}\max_{k\in[d]}\frac{1}{n^2}\left[\frac{2\|\mathbf{x}_i\|_2^2}{m(m-1)}\left(\frac{((m-1)/p_{ki})x_{ki}^2}{1+(m-1)p_{ki}} + \frac{(m-2)(m+1+1/p_{ki})x_{ki}^2}{1+(m-1)p_{ki}}\right)\right] \\ &\quad + \frac{1}{n^2m}\left\|\sum_{i=1}^{n}\sum_{k=1}^{d}\frac{x_{ki}^2}{p_{ki}}\mathbf{x}_i\mathbf{x}_i^T\right\|_2 \\ &\le \sum_{i=1}^{n}\max_{k\in[d]}\frac{1}{n^2}\left[\frac{8x_{ki}^4}{m^2p_{ki}^2}+\frac{4x_{ki}^4}{m^3p_{ki}^3}+\frac{\|\mathbf{x}_i\|_2^2x_{ki}^2}{mp_{ki}}+\frac{2x_{ki}^2}{m^2p_{ki}}\sum_{k=1}^{d}\frac{x_{ki}^2}{p_{ki}}\right] \\ &\quad + \sum_{i=1}^{n}\max_{k\in[d]}\frac{1}{n^2}\left[\frac{8\|\mathbf{x}_i\|_2^2x_{ki}^2}{mp_{ki}}\right]+\frac{1}{n^2m}\left\|\sum_{i=1}^{n}\sum_{k=1}^{d}\frac{x_{ki}^2}{p_{ki}}\mathbf{x}_i\mathbf{x}_i^T\right\|_2. \end{aligned} \quad (32)$$

As $p_{ki} = \alpha(|x_{ki}|/\|\mathbf{x}_i\|_1) + (1-\alpha)(x_{ki}^2/\|\mathbf{x}_i\|_2^2)$ with $0 < \alpha < 1$, and by plugging $p_{ki} \ge \alpha(|x_{ki}|/\|\mathbf{x}_i\|_1)$ and $p_{ki} \ge (1-\alpha)(x_{ki}^2/\|\mathbf{x}_i\|_2^2)$ into (32), we have

$$\begin{aligned}\sigma^2 &\le \sum_{i=1}^{n}\max_{k\in[d]}\frac{1}{n^2}\left[\frac{8\|\mathbf{x}_i\|_2^4}{m^2(1-\alpha)^2}+\frac{4\|\mathbf{x}_i\|_1^2\|\mathbf{x}_i\|_2^2}{m^3\alpha^2(1-\alpha)}+\frac{\|\mathbf{x}_i\|_2^4}{m(1-\alpha)} + \frac{2\|\mathbf{x}_i\|_2^2}{m^2(1-\alpha)}\sum_{k=1}^{d}\frac{|x_{ki}|\|\mathbf{x}_i\|_1}{\alpha}\right] \\ &\quad + \sum_{i=1}^{n}\max_{k\in[d]}\frac{1}{n^2}\left[\frac{8\|\mathbf{x}_i\|_2^4}{m(1-\alpha)}\right] + \frac{1}{n^2m}\left\|\sum_{i=1}^{n}\sum_{k=1}^{d}\frac{|x_{ki}|\|\mathbf{x}_i\|_1}{\alpha}\mathbf{x}_i\mathbf{x}_i^T\right\|_2 \\ &= \sum_{i=1}^{n}\left[\frac{8\|\mathbf{x}_i\|_2^4}{n^2m^2(1-\alpha)^2}+\frac{4\|\mathbf{x}_i\|_1^2\|\mathbf{x}_i\|_2^2}{n^2m^3\alpha^2(1-\alpha)}+\frac{9\|\mathbf{x}_i\|_2^4}{n^2m(1-\alpha)} + \frac{2\|\mathbf{x}_i\|_2^2\|\mathbf{x}_i\|_1^2}{n^2m^2\alpha(1-\alpha)}\right] + \left\|\sum_{i=1}^{n}\frac{\|\mathbf{x}_i\|_1^2\mathbf{x}_i\mathbf{x}_i^2}{n^2m\alpha}\right\|_2. \end{aligned} \quad (33)$$

By invoking Theorem 3, we obtain that for $\epsilon \ge 0$

$$\mathbb{P}(\|\mathbf{C}_e - \mathbf{C}\|_2 \ge \epsilon) \le 2d\exp\left(\frac{-\epsilon^2/2}{\sigma^2 + R\epsilon/3}\right) (:= \delta) \quad (34)$$

and the following quadratic equation in ϵ:

$$\frac{\epsilon^2}{2\log(2d/\delta)} - \frac{R\epsilon}{3} - \sigma^2 = 0. \quad (35)$$

Solving (35), we get the positive root

$$\begin{aligned}\epsilon &= \log\left(\frac{2d}{\delta}\right)\left[\frac{R}{3}+\sqrt{\left(\frac{R}{3}\right)^2+\frac{2\sigma^2}{\log(2d/\delta)}}\right] \\ &\le \log\left(\frac{2d}{\delta}\right)\frac{2R}{3}+\sqrt{2\sigma^2\log\left(\frac{2d}{\delta}\right)}. \end{aligned} \quad (36)$$

Thus, $\|\mathbf{C}_e-\mathbf{C}\|_2 \le \log((2d/\delta))(2R/3)+(2\sigma^2\log((2d/\delta)))^{1/2}$ holds with probability at least $1-\eta-\delta$ and we complete the proof. □

D. Proof of Corollary 1

Proof: By setting $\|\mathbf{x}_i\|_2 \le \tau$ for all $i \in [n]$, $\varphi := (\|\mathbf{x}_i\|_1/\|\mathbf{x}_i\|_2)$, where $1 \le \varphi \le \sqrt{d}$ and $m < d$ into Theorem 2, we obtain

$$\begin{aligned}\|\mathbf{C}_e - \mathbf{C}\|_2 &\le \widetilde{O}\left(\frac{\tau^2}{n}+\frac{\tau^2\varphi^2}{nm} + \sqrt{\frac{\tau^4}{nm^2}+\frac{\tau^4\varphi^2}{nm^3}+\frac{\tau^4}{nm}+\frac{\tau^4\varphi^2}{nm^2}+\frac{\|\mathbf{C}\|_2\tau^2\varphi^2}{nm}}\right) \\ &\le \widetilde{O}\left(\frac{\tau^2}{n}+\frac{\tau^2\varphi^2}{nm}+\frac{\tau^2\varphi}{m}\sqrt{\frac{1}{n}}+\tau^2\sqrt{\frac{1}{nm}}+\tau\varphi\sqrt{\frac{\|\mathbf{C}\|_2}{nm}}\right) \end{aligned} \quad (37)$$

where the first inequality invokes $\sum_{i=n}^{n}\|\mathbf{x}_i\|_2^4 \le n\tau^4$ and $\mathbf{C} = \sum_{i=1}^{n}(\mathbf{x}_i\mathbf{x}_i^T/n)$ is the original covariance matrix.

We can adopt $\sum_{i=1}^{n}\|\mathbf{x}_i\|_2^4 \le nd\tau^2\|\mathbf{C}\|_2$, which holds because $\sum_{i=1}^{n}\|\mathbf{x}_i\|_2^4 \le \tau^2\sum_{i=1}^{n}\|\mathbf{x}_i\|_2^2$ and $\sum_{i=1}^{n}\|\mathbf{x}_i\|_2^2 = n\mathrm{Tr}(\mathbf{C}) \le nd\|\mathbf{C}\|_2$, and derive

$$\begin{aligned}\|\mathbf{C}_e - \mathbf{C}\|_2 &\le \widetilde{O}\left(\frac{\tau^2}{n}+\frac{\tau^2\varphi^2}{nm}+\tau\sqrt{\|\mathbf{C}\|_2} \times\sqrt{\frac{d}{nm^2}+\frac{d\varphi^2}{nm^3}+\frac{d}{nm}+\frac{d\varphi^2}{nm^2}+\frac{\varphi^2}{nm}}\right) \\ &\le \widetilde{O}\left(\frac{\tau^2}{n}+\frac{\tau^2\varphi^2}{nm}+\frac{\tau\varphi}{m}\sqrt{\frac{d\|\mathbf{C}\|_2}{n}}+\tau\sqrt{\frac{d\|\mathbf{C}\|_2}{nm}}+\tau\varphi\sqrt{\frac{\|\mathbf{C}\|_2}{nm}}\right) \end{aligned} \quad (38)$$

Finally, assigning the smaller one of (37) and (38) to $\|\mathbf{C}_e-\mathbf{C}\|_2$ completes the proof. □

E. Proof of Corollaries 2 and 3

Proof: The proof follows [9, Corollaries 4–6], where the key component $\|\mathbf{C}_e - \mathbf{C}_p\|_2$ is upper bounded by

$\|\mathbf{C}_e-(1/n)\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T\|_2+\|(1/n)\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T-\mathbf{C}_p\|_2$. Then, via Theorem 2 and the Gaussian tail bounds in [9, Proposition 14], we can show that with probability at least $1-\zeta$ for $d\geq 2$

$$\max_{i\in[n]}\|\mathbf{x}_i\|_2 \leq \sqrt{2\mathrm{Tr}(\mathbf{C}_p)\log(nd/\zeta)}$$

$$\|\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T-\mathbf{C}_p\|_2 \leq O(\|\mathbf{C}_p\|_2\sqrt{\log(2/\zeta)/n}). \quad (39)$$

Applying them and Corollary 1 along with the fact that $\|\mathbf{x}_i\|_1\leq\sqrt{d}\|\mathbf{x}_i\|_2$ and $\mathrm{Tr}(\mathbf{C}_p)\leq d\|\mathbf{C}_p\|_2$, we establish

$$\begin{aligned}
&\|\mathbf{C}_e-\mathbf{C}_p\|_2\\
&\leq \left\|\mathbf{C}_e-\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T\right\|_2+\left\|\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T-\mathbf{C}_p\right\|_2\\
&\leq \widetilde{O}\Bigg(\frac{\tau^2}{n}+\frac{\tau^2\varphi^2}{nm}+\frac{\tau^2\varphi}{m}\sqrt{\frac{1}{n}}+\tau^2\sqrt{\frac{1}{nm}}\\
&\qquad +\tau\varphi\sqrt{\frac{\left\|\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T\right\|_2}{nm}}\Bigg)\\
&\quad+\widetilde{O}\left(\|\mathbf{C}_p\|_2\sqrt{\frac{1}{n}}\right)\\
&\leq \widetilde{O}\left(\frac{d^2\|\mathbf{C}_p\|_2}{nm}+\frac{d\|\mathbf{C}_p\|_2}{m}\sqrt{\frac{d}{n}}\right) \quad (40)
\end{aligned}$$

with probability at least $1-\eta-\delta-\zeta$, where we invoke (39) to get $\|(1/n)\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T\|_2\leq\|(1/n)\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T-\mathbf{C}_p\|_2+\|\mathbf{C}_p\|_2\leq O(\|\mathbf{C}_p\|_2)$.

Let $\mathrm{rank}(\mathbf{C}_p)\leq r$, and we have the result for the low-rank case

$$\begin{aligned}
\|[\mathbf{C}_e]_r-\mathbf{C}_p\|_2 &\leq \|[\mathbf{C}_e]_r-\mathbf{C}_e\|_2+\|\mathbf{C}_e-\mathbf{C}_p\|_2\\
&\leq \|[\mathbf{C}_p]_r-\mathbf{C}_e\|_2+\|\mathbf{C}_e-\mathbf{C}_p\|_2\\
&\leq \|[\mathbf{C}_p]_r-\mathbf{C}_p\|_2+\|\mathbf{C}_p-\mathbf{C}_e\|_2+\|\mathbf{C}_e-\mathbf{C}_p\|_2\\
&= 2\|\mathbf{C}_e-\mathbf{C}_p\|_2 \quad (41)
\end{aligned}$$

where the last equality holds because $\mathrm{rank}(\mathbf{C}_p)\leq r$. Then, armed with $\mathrm{Tr}(\mathbf{C}_p)\leq\mathrm{rank}(\mathbf{C}_p)\|\mathbf{C}_p\|_2\leq r\|\mathbf{C}_p\|_2$, we have

$$\begin{aligned}
&\|[\mathbf{C}_e]_r-\mathbf{C}_p\|_2\\
&\leq O(\|\mathbf{C}_e-\mathbf{C}_p\|_2)\\
&\leq O\left(\left\|\mathbf{C}_e-\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T\right\|_2+\left\|\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T-\mathbf{C}_p\right\|_2\right)\\
&\leq \widetilde{O}\left(\frac{rd\|\mathbf{C}_p\|_2}{nm}+\frac{r\|\mathbf{C}_p\|_2}{m}\sqrt{\frac{d}{n}}+\|\mathbf{C}_p\|_2\sqrt{\frac{rd}{nm}}\right) \quad (42)
\end{aligned}$$

with probability at least $1-\eta-\delta-\zeta$.

Due to the symmetry of $\mathbf{C}_p$ and $\mathbf{C}_e$, following [9], we can combine Davis–Kahan Theorem [19], $\|\widehat{\prod}_k-\prod_k\|_2\leq(1/\lambda_k-\lambda_{k+1})\|\mathbf{C}_e-\mathbf{C}_p\|_2$, with the result from Corollary 2 and immediately derive the desired bound in Corollary 3. □

REFERENCES

[1] S. Abbasi-Daresari and J. Abouei, "Toward cluster-based weighted compressive data aggregation in wireless sensor networks," *Ad Hoc Netw.*, vol. 36, pp. 368–385, Jan. 2016.

[2] R. Abrahamsson, Y. Selen, and P. Stoica, "Enhanced covariance matrix estimators in adaptive beamforming," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, vol. 2, Apr. 2007, pp. II-969–II-972.

[3] D. Achlioptas, Z. Karnin, and E. Liberty, "Near-optimal entrywise sampling for data matrices," in *Proc. NIPS*, Dec. 2013, pp. 1565–1573.

[4] D. Achlioptas and F. Mcsherry, "Fast computation of low-rank matrix approximations," *J. ACM*, vol. 54, no. 2, p. 9, Apr. 2007.

[5] F. Pourkamali-Anaraki, "Estimation of the sample covariance matrix from compressive measurements," *IET Signal Process.*, vol. 10, no. 9, pp. 1089–1095, Dec. 2016.

[6] F. Pourkamali-Anaraki and S. Becker, "Preconditioned data sparsification for big data with applications to PCA and K-means," *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 2954–2974, May 2017.

[7] F. Anaraki and S. Hughes, "Memory and computation efficient PCA via very sparse random projections," in *Proc. 31st Int. Conf. Mach. Learn.*, Jan. 2014, pp. 1341–1349.

[8] Y. Anzai, *Pattern Recognition and Machine Learning*. Amsterdam, The Netherlands: Elsevier, 2012.

[9] M. Azizyan, A. Krishnamurthy, and A. Singh, "Extreme compressive sampling for covariance estimation," Jun. 2015, *arXiv:1506.00898*. [Online]. Available: https://arxiv.org/abs/1506.00898

[10] D. Bartz, "Advances in high-dimensional covariance matrix estimation," Ph.D. thesis, Tech. Univ. Berlin, Berlin, Germany, 2016.

[11] J. M. Bioucas-Dias, D. Cohen, and Y. C. Eldar, "Covalsa: Covariance estimation from compressive measurements using alternating minimization," in *Proc. 22nd Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2014, pp. 999–1003.

[12] A. J. Butte, P. Tamayo, D. Slonim, T. R. Golub, and I. S. Kohane, "Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks," *Proc. Nat. Acad. Sci. USA*, vol. 97, no. 22, pp. 12182–12186, Oct. 2000.

[13] T. T. Cai and A. Zhang, "ROP: Matrix recovery via rank-one projections," *Ann. Statist.*, vol. 43, no. 1, pp. 102–138, 2015.

[14] X. Chen, I. King, and M. R. Lyu, "Frosh: FasteR online sketching hashing," in *Proc. UAI*, 2017, pp. 1–10.

[15] X. Chen, M. R. Lyu, and I. King, "Toward efficient and accurate covariance matrix estimation on compressed data," in *Proc. 34th Int. Conf. Mach. Learn.*, Aug. 2017, pp. 767–776.

[16] X. Chen, H. Yang, I. King, and M. R. Lyu, "Training-efficient feature map for shift-invariant kernels," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Jun. 2015, pp. 3395–3401.

[17] Y. Chen, Y. Chi, and A. J. Goldsmith, "Exact and stable covariance estimation from quadratic sampling via convex programming," *IEEE Trans. Inf. Theory*, vol. 61, no. 7, pp. 4034–4059, 2013.

[18] G. Dasarathy, P. Shah, B. N. Bhaskar, and R. D. Nowak, "Sketching Sparse Matrices, Covariances, and Graphs via Tensor Products," *IEEE Trans. Inf. Theory*, vol. 61, no. 3, pp. 1373–1388, Jul. 2015.

[19] C. Davis and W. M. Kahan, "The rotation of eigenvectors by a perturbation. III," *SIAM J. Numer. Anal.*, vol. 7, no. 1, pp. 1–46, 1970.

[20] M. P. Deisenroth, G. Neumann, and J. Peters, "A survey on policy search for robotics," *Found. Trends Robot.*, vol. 2, nos. 1–2, pp. 1–142, Aug. 2013.

[21] P. Drineas, R. Kannan, and M. W. Mahoney, "Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication," *SIAM J. Comput.*, vol. 36, no. 1, pp. 132–157, Jul. 2006.

[22] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, "Subspace sampling and relative-error matrix approximation," in *Approximation, Randomization, and Combinatorial Optimization*. Berlin, Germany: Springer, 2006.

[23] W. Feller, *Introduction to Probability Theory and Its Applications*, vol. 2. Hoboken, NJ, USA: Wiley, 1966.

[24] S. Gleichman and Y. C. Eldar, "Blind compressed sensing," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6958–6975, Oct. 2011.

[25] G. H. Golub and C. F. Van Loan, *Matrix Computations*. 1996.

[26] W. Ha and R. F. Barber, "Robust PCA with compressed data," in *Proc. NIPS*, 2015, pp. 1936–1944.

[27] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, 2011.

[28] L. P. Hansen, "Large sample properties of generalized method of moments estimators," *Econometrica, J. Econ. Soc.*, vol. 50, pp. 1029–1054, Jul. 1982.

[29] J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 92–101, Mar. 2008.

[30] J. T. Holodnak and I. C. Ipsen, "Randomized approximation of the gram matrix: Exact computation and probabilistic bounds," *SIAM J. Matrix Anal. Appl.*, vol. 36, no. 1, pp. 110–137, 2015.
[31] A. Hyvärinen, J. Karhunen, and E. Oja," *Independent Component Analysis*, vol. 46. Hoboken, NJ, USA: Wiley, 2004.
[32] I. T. Jolliffe, *Principal Component Analysis*. Hoboken, NJ, USA: Wiley, 2002.
[33] T. Kariya and H. Kurata, *Generalized Least Squares*. Hoboken, NJ, USA: Wiley, 2004.
[34] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2006, pp. 287–296.
[35] E. Liberty, "Simple and deterministic matrix sketching," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2013, pp. 581–588.
[36] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, pp. 3–30, Jan. 1998.
[37] Y. Mroueh, E. Marcheret, and V. Goel, "Co-occuring directions sketching for approximate matrix multiply," Oct. 2016, *arXiv:1610.07686*. [Online]. Available: https://arxiv.org/abs/1610.07686
[38] D. Papailiopoulos, A. Kyrillidis, and C. Boutsidis, "Provable deterministic leverage score sampling," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 997–1006.
[39] H. Qi and S. M. Hughes, "Invariance of principal components under low-dimensional random projection of the data," in *Proc. 19th IEEE Int. Conf. Image Process.*, Sep./Oct. 2012, pp. 937–940.
[40] J. Schäfer and K. Strimmer, "An empirical Bayes approach to inferring large-scale gene association networks," *Bioinformatics*, vol. 21, no. 6, pp. 754–764, 2005.
[41] T. Shi, D. Tang, L. Xu, and T. Moscibroda, "Correlated compressive sensing for networked data," in *Proc. 30th Conf. Uncertainty Artif. Intell.*, Jul. 2014, pp. 722–731.
[42] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki, "Practical data compression in wireless sensor networks: A survey," *J. Netw. Comput. Appl.*, vol. 35, no. 1, pp. 37–59, Jan. 2012.
[43] J. A. Tropp, "Improved analysis of the subsampled randomized Hadamard transform," *Adv. Adapt. Data Anal.*, vol. 3, no. 01n02, pp. 115–126, 2011.
[44] J. A. Tropp, "An introduction to matrix concentration inequalities," *Found. Trends Mach. Learn.*, vol. 8, pp. 1–230, May 2015.
[45] A. M. Tulino and S. Verdú," *Random Matrix Theory and Wireless Communications*, vol. 1. Boston, MA, USA: Now, 2004.
[46] D. P. Woodruff, "Sketching as a tool for numerical linear algebra," Nov. 2014, *arXiv:1411.4357*. [Online]. Available: https://arxiv.org/abs/1411.4357
[47] S. Wu, S. Bhojanapalli, S. Sanghavi, and A. G. Dimakis, "Single pass PCA of matrix products," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2016, pp. 2585–2593.
[48] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graph. Statist.*, vol. 15, no. 2, pp. 265–286, Jun. 2006.

Xixian Chen received the bachelor's degree from Nanjing University, Nanjing, China, in 2013, and the Ph.D. degree from the Computer Science and Engineering Department, The Chinese University of Hong Kong, Hong Kong.

He has been a Senior Researcher with the Tencent Youtu Lab, Shenzhen, China, since 2018. He has published technical publications in top-tier conferences in his area of expertise. His research interests include machine learning, deep learning, recommendation systems, big data, and computer vision.

Haiqin Yang (M'11) received the B.Sc. degree in computer science from Nanjing University, Nanjing, China, and the M.Phil. and Ph.D. degrees from the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.

He is currently a Machine Learning Research Scientist with Meitu and an Adjunct Assistant Professor with the Department of Computing, The Hang Seng University of Hong Kong, Hong Kong. He has authored two books and over 40 technical publications in journals/conferences in his areas of expertise. His current research interests include machine learning, data mining, and natural language processing.

Dr. Yang received the Young Researcher Award of the Asia Pacific Neural Network Society in 2018. He has initiated and co-organized five international workshops on the topics of scalable machine learning and scalable data analytics. He currently serves on the Editorial Board of *Neurocomputing* and also serves as a program committee member and a reviewer of over 20 top-tier conferences and prestigious journals.

Shenglin Zhao (M'18) received the bachelor's and master's degrees in engineering from the College of Electrical Engineering, Zhejiang University, in 2009 and 2012, respectively, and the Ph.D. degree from the Computer Science and Engineering Department, The Chinese University of Hong Kong, Hong Kong, in 2017.

He has been a Senior Researcher with Tencent Youtu Lab, Shenzhen, China, since 2018. His current research interests include deep learning, machine learning, recommendation systems, computer vision, and spatio-temporal data analysis. He has published over ten refereed journal and conference papers.

Michael R. Lyu (F'04) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, the M.S. degree in computer engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, and the Ph.D. degree in computer engineering from the University of California at Los Angeles, Los Angeles, CA, USA.

He was with the Jet Propulsion Laboratory, Pasadena, CA, USA, Telcordia Technologies, Piscataway, NJ, USA, and the Bell Laboratory, Murray Hill, NJ, USA, and taught at The University of Iowa, Iowa City, IA, USA. He is currently a Professor with the Computer Science and Engineering Department, The Chinese University of Hong Kong, Hong Kong. He has participated in more than 30 industrial projects. He has authored close to 400 papers in the following areas. His current research interests include software engineering, distributed systems, multimedia technologies, machine learning, social computing, and mobile networks.

Dr. Lyu is a fellow of the American Association for the Advancement of Science. He received the best paper awards in the IEEE International Symposium on Software Reliability Engineering (ISSRE) in 1998 and 2003, and the SigSoft Distinguished Paper Award in the International Conference on Software Engineering in 2010. He initiated the ISSRE, and was the Program Chair of the ISSRE in 1996 and the Program Co-Chair of the Tenth International World Web Conference, the Symposium on Reliable Distributed Systems in 2005, the International Conference on e-Business Engineering in 2007, and the International Conference on Services Computing in 2010. He was the General Chair of the ISSRE in 2001, the Pacific Rim International Symposium on Dependable Computing in 2005, and the International Conference on Dependable Systems and Networks in 2011. He has been named by the IEEE Reliability Society as the Reliability Engineer of the Year in 2011, for his contributions to software reliability engineering and software fault tolerance.

Irwin King (F'18) received the B.Sc. degree in engineering and applied science from the California Institute of Technology, Pasadena, CA, USA, and the M.Sc. and Ph.D. degrees in computer science from the University of Southern California, Los Angeles, CA, USA.

He was with AT&T Labs Research, Florham Park, NJ, USA, and also taught a number of courses at the University of California at Berkeley, Berkeley, CA, USA, as a Visiting Professor. He is currently the Associate Dean (Education) of the Faculty of Engineering and a Professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. He has over 30 research and applied grants and industry projects. Some notable projects include the VeriGuide System and the Knowledge and Education Exchange Platform. His research interests include machine learning, social computing, big data, Web intelligence, data mining, and multimedia information processing. In these research areas, he has authored over 200 technical publications in top international journals and conferences. In addition, he has contributed over 30 book chapters and edited volumes.

Dr. King currently serves as the President and a Governing Board Member of the International Neural Network Society. He serves as the General Co-Chair of WSDM 2011, RecSys 2013, and ACML 2015. He is an Associate Editor of the *ACM Transactions on Knowledge Discovery from Data* and the *Journal of Neural Networks*.