Toward Efficient and Accurate Covariance Matrix Estimation on Compressed Data

Xixian Chen [1 2] **Michael R. Lyu** [1 2] **Irwin King** [1 2]

Abstract

Estimating covariance matrices is a fundamental technique in various domains, most notably in machine learning and signal processing. To tackle the challenges of extensive communication costs, large storage capacity requirements, and high processing time complexity when handling massive high-dimensional and distributed data, we propose an efficient and accurate covariance matrix estimation method via data compression. In contrast to previous data-oblivious compression schemes, we leverage a data-aware weighted sampling method to construct low-dimensional data for such estimation. We rigorously prove that our proposed estimator is unbiased and requires smaller data to achieve the same accuracy with specially designed sampling distributions. Besides, we depict that the computational procedures in our algorithm are efficient. All achievements imply an improved tradeoff between the estimation accuracy and computational costs. Finally, the extensive experiments on synthetic and real-world datasets validate the superior property of our method and illustrate that it significantly outperforms the state-of-the-art algorithms.

1. Introduction

Covariance matrices play a fundamental role in machine learning and statistics owing to their capability to retain the second-order information of data samples (Feller, 1966). For example, Principal Component Analysis (PCA) along with its extensions (Zou et al., 2006), Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA) (Anzai, 2012) are powerful for dimension reduction and denoising, which require the estimation of a covariance matrix from a given collection of data points. Other prominent examples include Generalized Least Squares (GLS) regression that requires the estimation of the noise covariance matrix (Kariya & Kurata, 2004), Independent Component Analysis (ICA) that relies on pre-whitening based on the covariance matrix (Hyvärinen et al., 2004), and Generalized Method of Moments (GMM) (Hansen, 1982) that improves the effectiveness by a precise covariance matrix.

Many practical applications also rely on covariance matrix directly (Bartz, 2016). In biology, gene relevance networks and gene association networks are straightforwardly inferred from the covariance matrix (Butte et al., 2000; Schäfer & Strimmer, 2005). In modern wireless communications, protocols optimize the bandwidth based on covariance estimates (Tulino & Verdú, 2004). In array signal processing, the capon beamformer linearly combines the sensors to minimize the noise in the signal, which is closely related to the portfolio optimization on covariance matrices (Abrahamsson et al., 2007). For policy learning in the field of robotics, it requires reliable estimates of the covariance matrix between policy parameters (Deisenroth et al., 2013).

Calculation of a covariance matrix usually requires enormous computational resources in the form of communication and storage because large and high-dimensional data are now routinely gathered at an exploding rate from many distributed remote sites, such as sensor networks, surveillance, and distributed databases (Haupt et al., 2008; Shi et al., 2014; Ha & Barber, 2015). In particular, high communication cost of transmitting the distributed data from the remote sites to the fusion center (i.e., a destination to conduct complex data analysis tasks) will require tremendous bandwidth and power consumption (Srisooksai et al., 2012; Abbasi-Daresari & Abouei, 2016). Formally, given a data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$ with d features and n instances collected from the remote sites, the covariance matrix is computed in the fusion center by $\mathbf{C} \triangleq \frac{1}{n}\mathbf{X}\mathbf{X}^T - \bar{\mathbf{x}}\bar{\mathbf{x}}^T$, where $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \in \mathbb{R}^d$ (Feller, 1966). For simplicity

[1] Shenzhen Research Institute, The Chinese University of Hong Kong, Shenzhen, China. [2] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong. Correspondence to: Xixian Chen <xxchen@cse.cuhk.edu.hk>, Michael R. Lyu <lyu@cse.cuhk.edu.hk>, Irwin King <king@cse.cuhk.edu.hk>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

of discussion, we temporarily assume the empirical mean is zero, i.e., $\bar{\mathbf{x}} = \mathbf{0}$. The covariance matrix can be written as $\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$ consequently (Azizyan et al., 2015). Then, it takes $O(nd)$ communication burden to transmit data from numerous remote sites to the fusion center to form the full data set $\mathbf{X}$, $O(nd)$ storage in total to store $\mathbf{X}$ in remote sites, and $O(nd+d^2)$ storage with $O(nd^2)$ time to calculate $\mathbf{C}$ in the fusion center. When $n, d \gg 1$, the overall cost is prohibitively expensive for practical scenarios like wireless sensors which have narrow transmission bandwidth, limited storage, and low power supply.

To tackle such computational challenges, compressed data can be leveraged to estimate the covariance matrix, which essentially has roots in compressed sensing. One solution is to process each data point by multiplying it with a single projection matrix $\mathbf{S} \in \mathbb{R}^{d\times m}$ whose entry follows the Gaussian distribution $\mathcal{N}(0, \frac{1}{m})$ (Mahoney, 2011). Thus, storing $\mathbf{S}^T\mathbf{X}$ and the estimated covariance matrix requires $O(mn+d^2)$ space in total, sending $\mathbf{S}^T\mathbf{X}$ to the fusion center incurs a $O(mn)$ communication cost, and calculating $\mathbf{S}^T\mathbf{X}$ and the covariance matrix estimator $\frac{1}{n}\mathbf{S}\mathbf{S}^T\mathbf{X}\mathbf{X}^T\mathbf{S}\mathbf{S}^T$ takes $O(mdn + m^2n + m^2d + md^2)$ time. This method substantially reduces all computational costs if $m \ll n, d$. Note that synchronizing only a seed between remote sites and the fusion center allows pseudo-random number generators to reconstruct an identical $\mathbf{S}$, which avoids sending $\mathbf{S}$ directly and imposes a negligible computational burden.

However, the example solution has two critical drawbacks. The first is that the operations on the Gaussian matrix is inefficient. One could use a sparse projection matrix (Li et al., 2006), structured matrix (Ailon & Chazelle, 2009) or sampling matrix (Drineas et al., 2006b) to achieve a better tradeoff between computational cost and estimation precision. The second problem is that applying a single projection matrix to all data points cannot consistently estimate the covariance matrix, i.e., the estimator cannot converge to the actual covariance matrix even if the sample size n grows to infinity with d fixed. This issue is demonstrated both theoretically and empirically in (Azizyan et al., 2015) and also briefly described in (Gleichman & Eldar, 2011; Anaraki & Hughes, 2014; Anaraki & Becker, 2017).

In this paper, we thus adopt n distinct projection matrices for n data vectors (Azizyan et al., 2015; Anaraki & Hughes, 2014; Anaraki & Becker, 2017; Anaraki, 2016) to achieve consistent covariance matrix estimation, and construct a specific sampling matrix to increase both its efficiency and accuracy. On the whole, we do not make statistical assumptions on the distributed data $\mathbf{X} \in \mathbb{R}^{d\times n}$ with $n, d \gg 1$, nor do we impose structural assumptions on the covariance matrix $\mathbf{C}$ such as being low-rank or sparse. Our goal is to compress data and recover $\mathbf{C}$ efficiently and accurately, and the contributions in our work are summarized as follows:

- First, in contrast to all existing methods (Azizyan et al., 2015; Anaraki & Hughes, 2014; Anaraki & Becker, 2017; Anaraki, 2016) that are based on data-oblivious projection matrices, we propose to estimate the covariance matrix based on the data compressed by a weighted sampling scheme. This strategy is data-aware with a capacity to explore the most important entries. Hence, we require considerably fewer entries to achieve an equal estimation accuracy.

- Second, we provide error analysis for the derived unbiased covariance estimator, which rigorously demonstrates that our method can compress data to a much smaller volume than other methods. The proofs also indicate our probability distribution is specifically designed to render a covariance matrix estimation based on the compressed data as accurate as possible.

- Third, we specify our method by an efficient algorithm whose computational complexity is superior to other methods. By additionally considering the best tradeoff between the estimation accuracy and the compression ratio, our algorithm ultimately incurs a significantly lower computational cost than the other methods.

- Finally, we validate our method on both synthetic and real-world datasets, which demonstrates a better performance than the other methods.

The remainder of this paper is organized as follows. In Section 2, we review the prior work. In Section 3, we present our method along with theoretical analysis and emphasize its achievements. In Section 4, we provide extensive empirical results, and in Section 5 we conclude the whole work.

2. Related Work

There have been several investigations of ways to achieve accurate covariance matrix estimation from the low-dimensional compressed observations constructed by applying a distinct projection matrix $\{\mathbf{S}_i\}_{i=1}^n \in \mathbb{R}^{d\times m}$ to each data vector $\{\mathbf{x}_i\}_{i=1}^n \in \mathbb{R}^d$. The work of (Qi & Hughes, 2012) adopts a Gaussian matrix to compress data via $\mathbf{S}_i^T\mathbf{x}_i$, and recovers them by $\mathbf{S}_i(\mathbf{S}_i^T\mathbf{S}_i)^{-1}(\mathbf{S}_i^T\mathbf{x}_i)$. Because $\mathbf{S}_i(\mathbf{S}_i^T\mathbf{S}_i)^{-1}\mathbf{S}_i^T$ is a strictly m-dimensional orthogonal projection drawn uniformly at random, it can capture the information of all entries in each data vector uniformly and substantively. Then, $\frac{1}{n}\sum_{i=1}^n \mathbf{S}_i(\mathbf{S}_i^T\mathbf{S}_i)^{-1}\mathbf{S}_i^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{S}_i(\mathbf{S}_i^T\mathbf{S}_i)^{-1}\mathbf{S}_i^T$ up to a known scaling factor is expected to constitute accurate and consistent covariance matrix estimation. This estimator can be modified to an unbiased one, and its error analysis is thoroughly provided in (Azizyan et al., 2015). However, a Gaussian matrix is dense and unstructured, which imposes an extra computational burden. Also, many matrix inversions take a considerable amount of time, and the whole square matrix has to be loaded into the memory. Biased estimator $\frac{1}{n}\sum_{i=1}^{n}\mathbf{S}_i\mathbf{S}_i^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{S}_i\mathbf{S}_i^T$ is thus proposed in (Anaraki & Hughes, 2014) to improve the efficiency by avoiding matrix inversions and assigning $\mathbf{S}_i$ to be a sparse matrix. This method is less accurate because $\mathbf{S}_i\mathbf{S}_i^T$ approximates only an m-dimensional random orthogonal projection. Its another disadvantage is that the result only holds for data samples under statistical assumptions. Based on (Anaraki & Hughes, 2014), another study proposes an unbiased estimator (Anaraki, 2016), but it still adopts an unstructured sparse matrix that is insufficiently computation-efficient and fails to provide the error bounds to characterize the estimation error versus the compression ratio. Recently, sampling matrices $\mathbf{S}_i \in \mathbb{R}^{d\times m}$ constructed via uniform sampling *without replacement* have been employed (Anaraki & Becker, 2017). This approach is efficient, but it only results in poor accuracy if data are compressed directly by $\mathbf{S}_i^T\mathbf{x}_i \in \mathbb{R}^d$ because $\mathbf{S}_i\mathbf{S}_i^T$ is an m-dimensional orthogonal projection drawn only from d deterministic orthogonal spaces/coordinates, and the $d-m$ entries of each vector are removed. To avoid sacrificing much accuracy, use of the computationally efficient Hadamard matrix (Tropp, 2011) before sampling has also been proposed in (Anaraki & Becker, 2017). It flattens out whole entries, particularly those with large magnitudes, to all coordinates to ensure that poor uniform sampling with a small sampling size still obtains some information among all entries. However, the Hadamard matrix involves deterministic orthogonal projection and is unable to capture the information uniformly in all coordinates of each vector, which results in the need for numerous samples to achieve sufficient accuracy. (Anaraki & Becker, 2017) constitutes the current state of the art in the tradeoff between the estimation accuracy and computational efficiency. Throughout the paper, we group the foregoing representative methods into *Gauss-Inverse* (Azizyan et al., 2015; Qi & Hughes, 2012), *Sparse* (Anaraki & Hughes, 2014; Anaraki, 2016), and *UniSample-HD* (Anaraki & Becker, 2017), and the unbiased estimators produced by these methods are adopted in the subsequent theoretical and empirical comparisons.

A number of other methods have been proposed to recover covariance matrix from compressed data (Chen et al., 2013; Bioucas-Dias et al., 2014; Dasarathy et al., 2015; Cai et al., 2015). These methods are only applicable to low-rank, sparse, or statistically-assumed covariance matrices.

Interesting work has also been done in the area of low-rank matrix approximation via randomized techniques. In addition to simply embedding the data $\mathbf{X}$ into space spanned by a single random projection matrix $\mathbf{S}$, a representative study (Halko et al., 2011) improves approximation accuracy by replacing the random projection matrix $\mathbf{S}$ with a low-dimensional data-aware matrix $\mathbf{XS}'$, where $\mathbf{S}'$ is a random projection matrix. However, $\mathbf{X}$ has to be low-rank, and computing $\mathbf{XS}'$ requires one extra pass through all entries in $\mathbf{X}$. It is not suitable for our settings, where we do not impose structural assumptions on the covariance matrix, nor do we fully observe all data. Moreover, (Azizyan et al., 2015) demonstrates both theoretically and empirically that a single projection matrix for all data points cannot consistently and accurately estimate the covariance matrix. The problem also exist in (Wu et al., 2016; Mroueh et al., 2016) aiming for a fast approximation of matrix products in a single pass, which only results in an inconsistent covariance matrix estimation and suits the low-rank case.

Among randomized techniques, it is also worth briefly discussing sampling approaches in matrix approximation. Literature in (Drineas et al., 2006a; Papailiopoulos et al., 2014; Woodruff, 2014; Holodnak & Ipsen, 2015) proposes to leverage column sampling in which the sampling probabilities in the sampling matrix are either the column norms or leverage scores. Other work (Woodruff, 2014; Achlioptas & Mcsherry, 2007; Achlioptas et al., 2013) performs element-wise sampling on the entire matrix based on the relative magnitudes over all data entries. These researches employ different sampling distributions to sample entries in a matrix. However, they have to observe all data fully to calculate the sampling distributions, which also requires one or more extra passes. In addition, their sampling probabilities are designed for matrix approximation, which cannot be trivially extended to covariance matrix estimation because the exact covariance matrix in our setting cannot be calculated in advance. Note that although the uniform sampling in matrix approximation is a simple one-pass algorithm, it performs poorly on many problems because usually there exists structural non-uniformity in the data which has been verified in (Anaraki & Becker, 2017).

3. Our Approach

In this section, we first introduce the definition and background to our overall work. We then justify and present our method of data compression and covariance matrix estimation, followed by the primary results and analysis.

3.1. Preliminaries

Let $[k]$ denote a set of integers $\{1,2,\ldots,k\}$. Given a matrix $\mathbf{X}\in\mathbb{R}^{d\times n}$, for $j\in[d]$, $i\in[n]$, we let $\mathbf{x}_i\in\mathbb{R}^d$ denote the i-th column of $\mathbf{X}$, and x_{ji} denote the (j,i)-th element of $\mathbf{X}$ or j-th element of $\mathbf{x}_i$. Let $\{\mathbf{X}_t\}_{t=1}^k$ denote the set of matrices $\{\mathbf{X}_1,\mathbf{X}_2,\ldots,\mathbf{X}_k\}$, and $x_{ji,t}$ denote the (j,i)-th element of $\mathbf{X}_t$. Let $\mathbf{X}^T$ denote the transpose of $\mathbf{X}$, and $\mathrm{Tr}(\mathbf{X})$ denote its trace. Let $|x|$ denote the absolute value of x. Let $\|\mathbf{X}\|_2$ and $\|\mathbf{X}\|_F$ denote the spectral norm and Frobenius norm of $\mathbf{X}$, respectively. Let $\|\mathbf{x}\|_q=(\sum_{j=1}^d|x_j|^q)^{1/q}$ for $q\geq 1$ be the ℓ_q-norm of $\mathbf{x}\in\mathbb{R}^d$. Let $\mathbb{D}(\mathbf{x})$ be a square diagonal matrix with the elements of vector $\mathbf{x}$ on the main diagonal, and $\mathbb{D}(\mathbf{X})$ also be a square diagonal matrix whose main diagonal has only the main diagonal elements of $\mathbf{X}$.

3.2. Method and Algorithm

As discussed previously, *Gauss-Inverse* (Azizyan et al., 2015; Qi & Hughes, 2012) and *Sparse* (Anaraki & Hughes, 2014; Anaraki, 2016) suffer from deficiencies in either computational efficiency or estimation accuracy, whereas *UniSample-HD* (Anaraki & Becker, 2017) is less accurate but offers a good tradeoff between estimation accuracy and computational efficiency. We thus propose the adoption of weighted sampling matrices $\{\mathbf{S}_i\}_{i=1}^n \in \mathbb{R}^{d\times m}$ to compress data via $\mathbf{S}_i^T\mathbf{x}_i$ and then back-project the compressed data into the original space via $\mathbf{S}_i\mathbf{S}_i^T\mathbf{x}_i$. The recovered data is then used for covariance matrix estimation as shown in Eq. (1). Hence, a high computational efficiency is maintained. Although $\mathbf{S}_i$ removes at least $d-m$ entries from the i-th vector, the remainders can be the most informative and are retained. With the carefully designed sampling probabilities, our unbiased estimator $\mathbf{C}_e$ performs as accurately as or more accurately than its counterparts asymptotically in terms of matrix spectral norm $\|\mathbf{C}_e - \mathbf{C}\|_2$. Note we have not quantified the error in any other entry-wise norm (e.g., the Frobenius norm) that could be uninformative on the quality of the approximate invariant subspace and unstable regarding the additive random error (Anaraki, 2016; Achlioptas et al., 2013; Gittens, 2011).

Algorithm 1 The proposed algorithm.

```
Input:
    Data $\mathbf{X} \in \mathbb{R}^{d\times n}$, sampling size $m$, and $0 < \alpha < 1$.
Output:
    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:  Pass the compressed data $\mathbf{Y}$, sampling indices $\mathbf{T}$, $\mathbf{v}$, $\mathbf{w}$, and $\alpha$ to the fusion center.
7:  for all $i \in [n]$ do
8:      Initialize $\mathbf{S}_i \in \mathbb{R}^{d\times m}$ and $\mathbf{P} \in \mathbb{R}^{d\times n}$ with $\mathbf{0}$
9:      for all $j \in [m]$ do
10:         Let $p_{t_{ji}i} = \alpha\frac{|y_{ji}|}{v_i} + (1-\alpha)\frac{y_{ji}^2}{w_i}$, and $s_{t_{ji}j,i} = \frac{1}{\sqrt{mp_{t_{ji}i}}}$
11: Compute $\mathbf{C}_e$ as defined in Eq. (1) by using $\{\mathbf{S}_i\}_{i=1}^n$, $\mathbf{T}$, $\mathbf{P}$, and $\mathbf{Y}$.
```

We here summarize our method in Algorithm 1. In a nutshell, we employ a weighted sampling that is able to explore the most important entries to reduce estimation error $\|\mathbf{C}_e - \mathbf{C}\|_2$. Steps 1 to 5 in our proposed algorithm show how to compress distributed data in many remote sites. In step 5, each entry is retained with probability proportional to the combination of its relative absolute value and square value, and such sampling probability is designed to make $\|\mathbf{C}_e - \mathbf{C}\|_2$ as small as possible. Step 6 shows the communication procedure, and steps 7 to 11 reveal how to construct an unbiased covariance matrix estimator in the fusion center from compressed data. In many computing cases, it is possible to manipulate vectors of length $O(d)$ in memory, and thus when compressing data via weighted sampling, only one pass is required to move data from the external space to memory. Hence, our algorithm is also applicable to streaming data. For a covariance matrix defined as $\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^T - \bar{\mathbf{x}}\bar{\mathbf{x}}^T$, we can exactly calculate $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$ in the fusion center via $\bar{\mathbf{x}} = \frac{1}{n}\sum_{j=1}^g \mathbf{u}_j$, where $\{\mathbf{x}_i\}_{i=1}^n$ are from $g \ll n$ remote sites, and $\mathbf{u}_j \in \mathbb{R}^d$ is the summation of all data vectors in the j-th remote site before being compressed. Doing so makes no deviation on the following error analysis and imposes only a negligible computational burden.

3.3. Primary Provable Results

In this part, we introduce the proposed covariance matrix estimator. In Algorithm 1, we employ $\mathbf{Y}$, $\mathbf{T}$, $\mathbf{v}$, and $\mathbf{w}$ to calculate $\{\mathbf{S}_i\}_{i=1}^n$. It can be verified that using only $\{\mathbf{S}_i\}_{i=1}^n$ and $\mathbf{Y}$ is able to obtain $\{\mathbf{S}_i^T\mathbf{x}_i\}_{i=1}^n$. Thus, we describe our estimator via $\{\mathbf{S}_i\}_{i=1}^n$ and $\{\mathbf{S}_i^T\mathbf{x}_i\}_{i=1}^n$ in the following theorem, which shows our estimator is unbiased.

Theorem 1. *Assume $\mathbf{X} \in \mathbb{R}^{d\times n}$ and the sampling size $2 \le m < d$. Sample m entries 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 sampling matrix, respectively. Then, the unbiased estimator for the target covariance matrix $\mathbf{C} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T = \frac{1}{n}\mathbf{X}\mathbf{X}^T$ can be recovered as*

$$\mathbf{C}_e = \widehat{\mathbf{C}}_1 - \widehat{\mathbf{C}}_2, \tag{1}$$

where $\mathbb{E}[\mathbf{C}_e] = \mathbf{C}$, $\widehat{\mathbf{C}}_1 = \frac{m}{nm-n}\sum_{i=1}^n \mathbf{S}_i\mathbf{S}_i^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{S}_i\mathbf{S}_i^T$, and $\widehat{\mathbf{C}}_2 = \frac{m}{nm-n}\sum_{i=1}^n \mathbb{D}(\mathbf{S}_i\mathbf{S}_i^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{S}_i\mathbf{S}_i^T)\mathbb{D}(\mathbf{b}_i)$ with $b_{ki} = \frac{1}{1+(m-1)p_{ki}}$.

Note that at most m entries in each $\mathbf{b}_i$ have to be calculated because each $\mathbf{S}_i\mathbf{S}_i^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{S}_i\mathbf{S}_i^T$ has at most m non-zero entries in its diagonal. Now, having achieved the above unbiased estimator $\mathbf{C}_e$, we analyze its properties. We precisely upper bound the estimation error for the original estimator $\mathbf{C}$ in the matrix spectral norm.

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

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

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

A large R and σ^2 work against the accuracy of $\mathbf{C}_e$. Accordingly, our sampling probabilities are designed to make R and σ^2 as small as possible to improve the accuracy. In the proof of Theorem 2, we also show that the selection of $q = 1, 2$ in $\frac{|x_{ki}|^q}{\sum_{k=1}^d |x_{ki}|^q}$ used for constructing the sampling probability $p_{ki} = \alpha \frac{|x_{ki}|}{\|\mathbf{x}_i\|_1} + (1-\alpha)\frac{x_{ki}^2}{\|\mathbf{x}_i\|_2^2}$ is necessary and sufficient to make the error bound considerably tight.

Furthermore, α balances the performance by ℓ_1-norm based sampling $\frac{|x_{ki}|}{\|\mathbf{x}_i\|_1}$ and ℓ_2-norm based sampling $\frac{x_{ki}^2}{\|\mathbf{x}_i\|_2^2}$. ℓ_2 sampling penalizes small entries more than ℓ_1 sampling. Hence ℓ_2 sampling is more likely to select larger entries to decrease error. However, as seen from the proof in the *appendix*, different from ℓ_1 sampling, ℓ_2 sampling is unstable and sensitive to small entries, and it can make estimation error incredibly high if extremely small entries are picked. Hence, if α varies from 1 to 0, the estimation error will decrease and then increase, which is also empirically verified in the *appendix*.

The error bound in Theorem 2 involves many data-dependent quantities, whereas our primary interest lies in studying the tradeoff between the computational efficiency and estimation accuracy by employing weighted sampling to compress data and estimate covariance matrix. To clarify, we modify Theorem 2 and make the bound explicitly dependent on n, d, and m with the constraint $2 \le m < d$.

Corollary 1. *Given* $\mathbf{X} \in \mathbb{R}^{d\times n}$ *and the sampling size* $2 \le m < d$*, let* $\mathbf{C}$ *and* $\mathbf{C}_e$ *be created by Algorithm 1. Define* $\frac{\|\mathbf{x}_i\|_1}{\|\mathbf{x}_i\|_2} \le \varphi$ *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*

$$\|\mathbf{C}_e - \mathbf{C}\|_2 \le \min\{\widetilde{O}\Big(f + \frac{\tau^2\varphi}{m}\sqrt{\frac{1}{n}} + \tau^2\sqrt{\frac{1}{nm}}\Big), \widetilde{O}\Big(f + \frac{\tau\varphi}{m}\sqrt{\frac{d\|\mathbf{C}\|_2}{n}} + \tau\sqrt{\frac{d\|\mathbf{C}\|_2}{nm}}\Big)\}, \qquad (3)$$

where $f = \frac{\tau^2}{n} + \frac{\tau^2\varphi^2}{nm} + \tau\varphi\sqrt{\frac{\|\mathbf{C}\|_2}{nm}}$*, and* $\widetilde{O}(\cdot)$ *hides the logarithmic factors on* η*,* δ*,* m*,* n*,* d*, and* α*.*

The formulation above explores the fact that $1 \le \|\mathbf{x}_i\|_1/\|\mathbf{x}_i\|_2 \le \sqrt{d}$ by the Cauchy-Schwarz inequality. Before proceeding, we make several remarks to make a comparison with the following representative work: *Gauss-Inverse*, *UniSample-HD*, and *Sparse*. The first two methods provide error analysis without assuming data distribution, which is shown in (Azizyan et al., 2015; Anaraki & Becker, 2017) and illustrated in our *appendix*. In the following remarks, only our method is sensitive to φ, and we also employ the fact that $\frac{1}{nd}\|\mathbf{X}\|_F^2 \le \|\mathbf{C}\|_2 \le \max_{i\in[n]}\|\mathbf{x}_i\|_2^2 = \tau^2$ to simplify all asymptotic bounds.

Remark 1. Eq. (3) with $\varphi = \sqrt{d}$ indicates the error bound for our estimator $\mathbf{C}_e$ in the worst case, where the magnitudes of each entry in all of the input data vectors are the same (i.e., highly uniformly distributed). Even in this case, our error bound has a leading term of order $\min\{\widetilde{O}\Big(\frac{\tau^2 d}{nm} + \tau\sqrt{\frac{d\|\mathbf{C}\|_2}{nm}} + \frac{\tau^2}{m}\sqrt{\frac{d}{n}}\Big), \widetilde{O}\Big(\frac{\tau^2 d}{nm} + \frac{\tau d}{m}\sqrt{\frac{\|\mathbf{C}\|_2}{n}}\Big)\}$, which is the same as *Gauss-Inverse* ignoring logarithmic factors. In contrast, as the magnitudes of the entries in each data vector become uneven, φ gets smaller, leading to 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 Eq. (3) becomes $\min\{\widetilde{O}\Big(\frac{\tau^2}{n} + \tau^2\sqrt{\frac{1}{nm}}\Big), \widetilde{O}\Big(\frac{\tau^2}{n} + \tau\sqrt{\frac{d\|\mathbf{C}\|_2}{nm}}\Big)\}$, which is tighter than the leading term of *Gauss-Inverse* by a factor of at least $\sqrt{d/m}$. As explained in the next section, *Gauss-Inverse* also lacks computational efficiency.

Remark 2. As our target is to compress data to a smaller m that is not comparable to d in practice, $O(d-m)$ can be approximately regarded as $O(d)$. Then, the error of *UniSample-HD* is $\widetilde{O}\Big(\frac{\tau^2 d}{nm} + \tau\sqrt{\frac{d\|\mathbf{C}\|_2}{nm}} + \frac{\tau^2 d}{m}\sqrt{\frac{1}{nm}}\Big)$, which is asymptotically worse than our bound. When n is sufficiently large, the leading term of its error becomes $\widetilde{O}\Big(\tau\sqrt{\frac{d\|\mathbf{C}\|_2}{nm}} + \frac{\tau^2 d}{m}\sqrt{\frac{1}{nm}}\Big)$, which can be weaker than the leading term in our method by a factor of 1 to $\sqrt{d/m}$ when $\varphi = \sqrt{d}$, and at least d/m when $\varphi = O(1)$.

However, if m is sufficiently close to d, which is not meaningful for practical usage, $O(d-m) = O(1)$ will hold and the error of *UniSample-HD* becomes $\widetilde{O}\Big(\frac{\tau^2 d}{nm} + \tau\sqrt{\frac{d\|\mathbf{C}\|_2}{nm}} + \frac{\tau^2}{m}\sqrt{\frac{d}{nm}}\Big)$. This bound may slightly outperform ours by a factor of $\sqrt{d/m} = O(1)$ when $\varphi = \sqrt{d}$, but is still worse than ours when $\varphi = O(1)$. These results also coincide with the fact that *UniSample-HD* adopts uniform sampling *without replacement* combined with the Hadamard matrix, but we employ weighted sampling *with replacement*.

Remark 3. The *Sparse* method, which employs a sparse matrix for each $\mathbf{S}_i$, is not sufficiently accurate as demonstrated in our experiments. Moreover, there is no error analysis available for its unbiased estimator to characterize the estimation error versus the compression ratio.

Thus far, we have not made statistical nor structural assumptions concerning the input data or covariance matrix to derive our provable results. However, motivated by (Azizyan et al., 2015), it is also straightforward to extend our results to the statistical data and a low-rank covariance matrix estimation. The derived results below are polynomially equivalent to those in *Gauss-Inverse* (Azizyan et al., 2015). Corollary 2 shows the (low-rank) covariance matrix estimation on Gaussian data, and Corollary 3 indicates the derived covariance estimator also guarantees the accuracy of the principal components regarding the subspace learning.

Corollary 2. *Given $\mathbf{X} \in \mathbb{R}^{d \times n}$ ($2 \leq d$) and 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 $\mathcal{N}(\mathbf{0}, \mathbf{C}_p)$. Let $\mathbf{C}_e$ be constructed by Algorithm 1 with the sampling size $2 \leq m < d$. Then, with probability at least $1 - \eta - \delta - \zeta$,*

$$\frac{\|\mathbf{C}_e - \mathbf{C}_p\|_2}{\|\mathbf{C}_p\|_2} \leq \widetilde{O}\Big(\frac{d^2}{nm} + \frac{d}{m}\sqrt{\frac{d}{n}}\Big); \tag{4}$$

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

$$\frac{\|[\mathbf{C}_e]_r - \mathbf{C}_p\|_2}{\|\mathbf{C}_p\|_2} \leq \widetilde{O}\Big(\frac{rd}{nm} + \frac{r}{m}\sqrt{\frac{d}{n}} + \sqrt{\frac{rd}{nm}}\Big), \tag{5}$$

where $[\mathbf{C}_e]_r$ is the solution to $\min_{rank(A) \leq r} \|\mathbf{A} - \mathbf{C}_e\|_2$, 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 k-th largest eigenvalue of $\mathbf{C}_p$. Then, with probability at least $1 - \eta - \delta - \zeta$,*

$$\frac{\|\widehat{\prod}_k - \prod_k\|_2}{\|\mathbf{C}_p\|_2} \leq \frac{1}{\lambda_k - \lambda_{k+1}} \widetilde{O}\Big(\frac{d^2}{nm} + \frac{d}{m}\sqrt{\frac{d}{n}}\Big), \tag{6}$$

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.

The proof details of all our theoretical results are relegated to the *appendix*. We leverage the Matrix Bernstein inequality (Tropp, 2015) and establish the error bound of our proposed estimator on an arbitrary sampling probability in order to determine which sampling probability brings the best estimation accuracy. The employment of the Matrix Bernstein inequality involves controlling the range and variance of all zero-mean random matrices, whose derivations differ from those in (Azizyan et al., 2015; Anaraki & Becker, 2017) because of different data compression schemes. Moreover, to obtain the desired tight bound for the range and variance, we precisely provide a group of closed-form equalities or concentration inequalities for various quantities (see our proposed **Lemma 1** and **Lemma 2** along with their proofs in the *appendix*).

3.4. Computational Complexity

Recall that we have n data samples in the d-dimensional space, and let m be the target compressed dimension. Regarding estimating $\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$, the computational comparisons between our method and the representative baseline methods are presented in Table 1, in which *Standard* method means that we compute $\mathbf{C}$ directly without data compression. For the definition of covariance matrix $\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^T - \bar{\mathbf{x}}\bar{\mathbf{x}}^T$, extra computational costs (i.e., $O(gd)$ storage, $O(gd)$ communication cost, and $O(nd)$ time) must be added to the last four compression methods in the table, where $g \ll n$ is the number of the entire remote sites. All detailed analysis is relegated to the *appendix*.

T_G and T_S in Table 1 represent the time taken to generate the Gaussian matrices and sparse matrices by fast pseudo-random number generators like Mersenne Twister (Matsumoto & Nishimura, 1998), which can be enormous (Anaraki & Becker, 2017) and proportional to nmd and nd^2, respectively, up to certain small constants. Hence, our method can be regarded as the most efficient when d is large. Furthermore, by using the smallest m to obtain the same estimation accuracy as the other methods, our approach incurs the least computational burden.

4. Empirical Studies

In this section, we empirically verify the properties of the proposed method and demonstrate its superiority. We compare its estimation accuracy with that of *Gauss-Inverse*, *Sparse*, and *UniSample-HD*. We also report the time comparisons.

We run all algorithms on both synthetic and real-world datasets whose largest dimension is around and below 10^5. Such dimension is not very high in modern data analysis, but this limitation is due to that reporting the estimation error by calculating the spectral norm of a covariance matrix with its size larger than $10^5 \times 10^5$ will take intolerable amount of memory and time. The parameter selection on α is deferred to the *appendix*, and we empirically set $\alpha = 0.9$. To allow a fair comparison of the time consumptions measured by FLOPS, we implement all algorithms in C++ and run them in a single thread mode on a standard workstation with Intel CPU@2.90GHz and 128GB RAM.

Table 1. Computational costs on the storage, communication, and time.

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

4.1. Experiments on Synthetic Datasets

To clearly examine the performance, we compare all methods on six synthetic datasets: $\{\mathbf{X}_i\}_{i=1}^3 \in \mathbb{R}^{1024\times 20000}$, $\mathbf{X}_4 \in \mathbb{R}^{1024\times 200000}$, $\mathbf{X}_5 \in \mathbb{R}^{2048\times 200000}$, and $\mathbf{X}_6 \in \mathbb{R}^{65536\times 200000}$, which are generated based on the generative model (Liberty, 2013). Specifically, given a matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$ from such model, it is formally defined as $\mathbf{X} = \mathbf{UFG}$, where $\mathbf{U} \in \mathbb{R}^{d\times k}$ defines the signal column space with $\mathbf{U}^T\mathbf{U} = \mathbf{I}_k$ $(k \le d)$, the square diagonal matrix $\mathbf{F} \in \mathbb{R}^{k\times k}$ contains the diagonal entries $f_{ii} = 1-(i-1)/k$ that gives linearly diminishing signal singular values, and $\mathbf{G} \in \mathbb{R}^{k\times n}$ is the signal coefficient with $g_{ij} \sim \mathcal{N}(0,1)$ that is the Gaussian distribution. We let $k \approx 0.005d$, then setting $d = 1024$ and $n = 20000$ completes the creation of data $\mathbf{X}_1$. For $\mathbf{X}_2$, it is defined as $\mathbf{DX}$, where each entry in the square diagonal matrix $\mathbf{D}$ is defined by $d_{ii} = 1/\beta_i$, and β_i is randomly sampled from the integer set [15]. Regarding $\mathbf{X}_3$, it is constructed in the same way as $\mathbf{X}_1$ except that $\mathbf{F}$ now becomes an identity matrix. Next, $\{\mathbf{X}_i\}_{i=4}^6$ follow the same generation strategy of $\mathbf{X}_2$ except for the n and d.

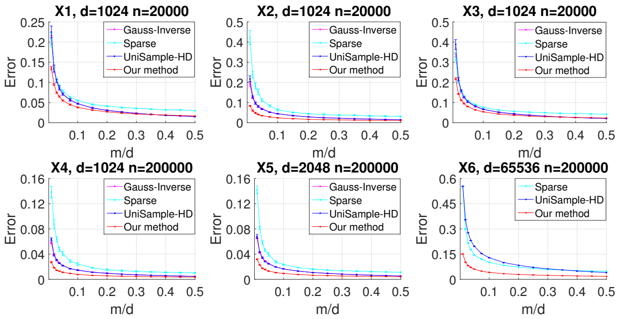

Figure 1. Accuracy comparisons of covariance matrix estimation on synthetic datasets. The estimation error is measured by $\|\mathbf{C}_e - \mathbf{C}\|_2/\|\mathbf{C}\|_2$ with $\mathbf{C}_e$ calculated by all compared methods, and cf $= m/d$ is the compression ratio.

In Figure 1, we plot the relative estimation error averaged over ten runs with its standard deviation versus the naive compression ratio cf $= m/d$. Note that a large cf is not necessary for practical usage, and our aim is to compress data to a smaller volume. In Figure 2, we report the running time taken in both the compressing and recovering stages, which preliminarily depicts the efficiency of the different methods and indicates how much power should be spent in the practical computation.

Generally, our method displays the least error and deviation for all datasets and its error decreases dramatically with an increase at a small cf. This observation indicates that our method can achieve sufficient estimation accuracy by using substantially fewer data entries than the other methods. For $\mathbf{X}_1$ ($\varphi = 0.81\sqrt{d}$), the magnitudes of the data entries are highly uniformly distributed, and thus our method can be regarded as uniform sampling with replacement, which may perform slightly worse than *UniSample-HD* and *Gauss-Inverse* if cf becomes large enough. After allowing the magnitudes to vary within a moderately larger range in $\mathbf{X}_2$ ($\varphi = 0.55\sqrt{d}$), our method considerably outperforms the other three methods. Its improvement comes from that *only* our method is sensitive to φ and a smaller φ produces a tighter result, as demonstrated by Remarks 1 and 2. However, the error of each method in $\mathbf{X}_3$ ($\tau/\sqrt{\|\mathbf{C}\|_2} = 5.5, \varphi = 0.81\sqrt{d}$) is larger than that in $\mathbf{X}_1$ ($\tau/\sqrt{\|\mathbf{C}\|_2} = 4.3, \varphi = 0.81\sqrt{d}$), respectively. It is because of that almost *all* methods are sensitive to $\tau/\sqrt{\|\mathbf{C}\|_2}$, and the error $\|\mathbf{C}_e - \mathbf{C}\|_2/\|\mathbf{C}\|_2$ increases when $\tau/\sqrt{\|\mathbf{C}\|_2}$ rises. Such phenomenon is demonstrated via dividing numerous error bounds in Remarks 1 and 2 by $\|\mathbf{C}\|_2$. Our method also achieves the best performance in $\mathbf{X}_4$. Although the φ and $\tau/\sqrt{\|\mathbf{C}\|_2}$ in $\mathbf{X}_4$ are approximately equal with those in $\mathbf{X}_2$, yet the proved error bounds with Remarks 1 and 2 reveal that a larger n in $\mathbf{X}_4$ will lead to smaller estimation errors given the same cf. Finally, our method also achieves the best accuracy when the dimension d increases in both $\mathbf{X}_5$ and $\mathbf{X}_6$. Besides, taking more data (i.e., enlarging n) as suggested by $\mathbf{X}_4$ can be considered to reduce the error in $\mathbf{X}_5$ and $\mathbf{X}_6$. Note that *Gauss-Inverse* has not been run on $\mathbf{X}_6$ since it costs enormous time.

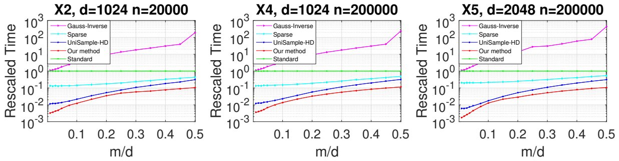

Figure 2. Time comparisons of covariance matrix estimation on synthetic datasets. Rescaled time results from the running time normalized by that spent in the *Standard* way of calculating $\mathbf{C} = \mathbf{XX}^T/n$ without data compression, and it is plotted in log scale.

Turning to *Gauss-Inverse*, it becomes highly accurate when cf increases but requires much more time than *Standard* (see Figure 2) so that its usage might be ruled out in practice. However, *Gauss-Inverse* remains a good choice when we are in urgent need of reducing the storage and communication burden. *Sparse*, which has no error analysis of its unbiased estimator, generally performs less accurately than the others but requires less time than *Standard*. *UniSample-HD* is efficient while it still consumes more time than our method. Also, its accuracy is inferior to our method especially when cf is small. In conclusion, our method is capable of compressing data to a very small size while guaranteeing both estimation accuracy and computational efficiency.

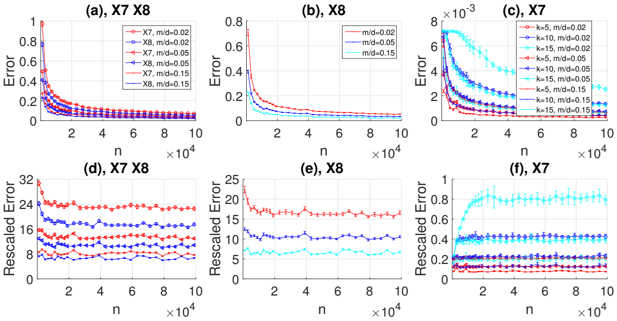

Figure 3. Convergence rates of our method for the settings in Corollaries 2 and 3.

As confirmed in Figure 1, a large n benefits the estimation accuracy. Thus, we study its effect more quantitatively. We conduct experiments following the settings as defined in Corollaries 2 and 3, and their results in Eqs. (4)-(6) clearly show that the errors decay in $1/\sqrt{n}$ convergence rate if $d \ll n$. We run our method on another two synthetic datasets $\{\mathbf{X}_t\}_{t=7}^{8} \in \mathbb{R}^{d \times n}$ that follow the d-dimensional multivariate normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{C}_{pt})$, where the (i,j)-th element of $\mathbf{C}_{p7} \in \mathbb{R}^{d \times d}$ is $0.5^{|i-j|/50}$, and $\mathbf{C}_{p8} \in \mathbb{R}^{d \times d}$ is a low-rank matrix that satisfies $\min_{\text{rank}(\mathbf{A}) \leq r} \|\mathbf{A} - \mathbf{C}_{p7}\|_2$. We take $d = 1000$, $r = 5$, $m/d = \{0.02, 0.05, 0.15\}$, $k = \{5, 10, 15\}$, and vary n from 1000 to 100000. In Figure 3, the top three plots report the errors as defined in the LHS of Eqs. (4)-(6), respectively. Then, dividing such errors by $1/\sqrt{n}$ obtains the bottom three plots accordingly.

The observation that the curves in plots (d)-(f) are roughly flat validates that the error bounds induced by our method decay rapidly with n in the $1/\sqrt{n}$ convergence rate, which coincides with Eqs. (4)-(6). In addition to the fast error convergence for the low-rank matrix $\mathbf{C}_{p8}$, our method can also obtain an increasingly better estimation accuracy for a high-rank covariance matrix $\mathbf{C}_{p7}$ if we enlarge n, which is displayed in plot (a). Besides, considering the omitted plot where the eigengap $\lambda_k - \lambda_{k+1}$ of $\mathbf{C}_{p7}$ decreases with k, the fact that the errors in plot (c) increase with k also coheres with Eq. (6). To conclude, our method also adapts well to the specific settings in Corollaries 2 and 3, and all induced error bounds indeed satisfy a $1/\sqrt{n}$ convergence rate.

4.2. Experiments on Real-world Datasets

In the second set of experiments, we use nine publicly available real-world datasets (Chang & Lin, 2011; Blake & Merz, 1998; Amsaleg, 2010), some of which are gathered from many distributed sensors. Their statistics are displayed in Figure 4. We again compare the estimation accuracy of the proposed method against the other three approaches. As can be seen from the figure, our method consistently exhibits superior accuracy over all $\text{cf} = m/d$, and its error decreases dramatically when cf grows. The error of the other three methods also decreases with cf but is still large at a small cf. Besides, our method enjoys the least deviation. In summary, these results confirm that our method can compress data to the lowest volume with the best accuracy, thereby substantially reducing storage, communication, and processing time cost in practice.

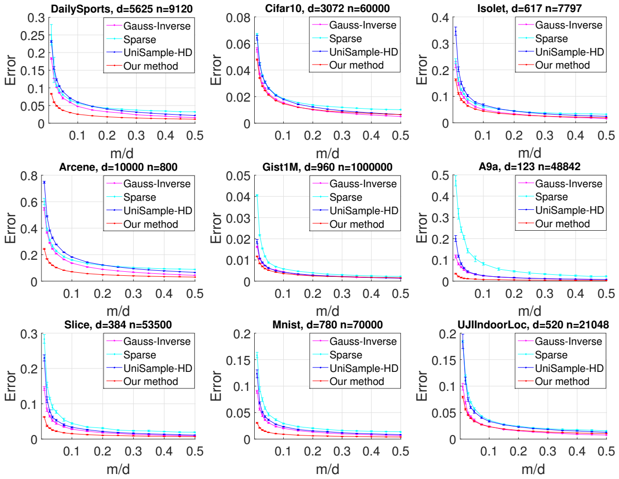

Figure 4. Accuracy comparisons of covariance matrix estimation on real-world datasets.

5. Conclusion

In this paper, we describe a weighted sampling method for accurate and efficient calculation of an unbiased covariance matrix estimator. The analysis demonstrates that our method can employ a smaller data volume than the other approaches to achieve an equal accuracy, and is highly efficient regarding the communication, storage, and processing time. The empirical results of the algorithm's application to both synthetic and real-world datasets further support our analysis and demonstrate its significant improvements over other state-of-the-art methods.

Compared with the sampling-with-replacement scheme in this paper, we seek to make more achievements via a sampling-without-replacement scheme in the future work. Analyzing the corresponding unbiased estimator will pose significant technical challenges in this research direction.

Acknowledgments

We truly thank Akshay Krishnamurthy for the fruitful discussions and interpretations on (Azizyan et al., 2015). We also thank Yuxin Su for the assistance on the experiments. The work described in this paper was fully supported by the National Natural Science Foundation of China (Project No. 61332010), the Research Grants Council of the Hong Kong Special Administrative Region, China ((No. CUHK 14208815 and No. CUHK 14234416 of the General Research Fund), and 2015 Microsoft Research Asia Collaborative Research Program (Project No. FY16-RES-THEME-005).

References

Abbasi-Daresari, S. and Abouei, J. Toward cluster-based weighted compressive data aggregation in wireless sensor networks. *Ad Hoc Networks*, 36:368–385, 2016.

Abrahamsson, R., Selen, Y., and Stoica, P. Enhanced covariance matrix estimators in adaptive beamforming. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 2, pp. II–969. IEEE, 2007.

Achlioptas, D. and Mcsherry, F. Fast computation of low-rank matrix approximations. *Proceedings of the annual ACM symposium on Theory of computing*, 54(2):9, 2007.

Achlioptas, D., Karnin, Z. S., and Liberty, E. Near-optimal entrywise sampling for data matrices. In *Advances in Neural Information Processing Systems*, pp. 1565–1573, 2013.

Ailon, N. and Chazelle, B. The fast johnson-lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009.

Amsaleg, L. Datasets for approximate nearest neighbor search, 2010.

Anaraki, F. Estimation of the sample covariance matrix from compressive measurements. *IET Signal Processing*, 2016.

Anaraki, F. and Becker, S. Preconditioned data sparsification for big data with applications to pca and k-means. *IEEE Transactions on Information Theory*, 2017.

Anaraki, F. and Hughes, S. Memory and computation efficient pca via very sparse random projections. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1341–1349, 2014.

Anzai, Y. *Pattern Recognition & Machine Learning*. Elsevier, 2012.

Azizyan, M., Krishnamurthy, A., and Singh, A. Extreme compressive sampling for covariance estimation. *arXiv preprint arXiv:1506.00898*, 2015.

Bartz, D. Advances in high-dimensional covariance matrix estimation. 2016.

Bioucas-Dias, J., Cohen, D., and Eldar, Y. C. Covalsa: Covariance estimation from compressive measurements using alternating minimization. In *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, pp. 999–1003. IEEE, 2014.

Blake, C.L. and Merz, C.J. UCI repository of machine learning databases, 1998.

Butte, A. J., Tamayo, P., Slonim, D., Golub, T. R., and Kohane, I. S. Discovering functional relationships between rna expression and chemotherapeutic susceptibility using relevance networks. *Proceedings of the National Academy of Sciences*, 97(22):12182–12186, 2000.

Cai, T. T., Zhang, A., et al. Rop: Matrix recovery via rank-one projections. *The Annals of Statistics*, 43(1):102–138, 2015.

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

Chen, Y., Chi, Y., and Goldsmith, A. Exact and stable covariance estimation from quadratic sampling via convex programming. 2013.

Dasarathy, G., Shah, P., Bhaskar, B. N., and Nowak, R. D. Sketching sparse matrices, covariances, and graphs via tensor products. *Information Theory, IEEE Transactions on*, 61(3):1373–1388, 2015.

Deisenroth, M. P., Neumann, G., Peters, J., et al. A survey on policy search for robotics. *Foundations and Trends in Robotics*, 2(1-2):1–142, 2013.

Drineas, P., Kannan, R., and Mahoney, M. W. Fast monte carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006a.

Drineas, P., Mahoney, M. W., and Muthukrishnan, S. Subspace sampling and relative-error matrix approximation. In *Approximation, Randomization, and Combinatorial Optimization*. 2006b.

Feller, W. introduction to probability theory and its applications. vol. ii.[an]. 1966.

Gittens, A. The spectral norm error of the naive nystrom extension. *arXiv preprint arXiv:1110.5305*, 2011.

Gleichman, S. and Eldar, Y. C. Blind compressed sensing. *Information Theory, IEEE Transactions on*, 57(10): 6958–6975, 2011.

Ha, W. and Barber, R. F. Robust pca with compressed data. In *Advances in Neural Information Processing Systems*, pp. 1936–1944, 2015.

Halko, N., Martinsson, P., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

Hansen, L. P. Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the Econometric Society*, pp. 1029–1054, 1982.

Haupt, J., Bajwa, W. U., Rabbat, M., and Nowak, R. Compressed sensing for networked data. *IEEE Signal Processing Magazine*, 25(2):92–101, 2008.

Holodnak, J. T. and Ipsen, I. C. Randomized approximation of the gram matrix: Exact computation and probabilistic bounds. *SIAM Journal on Matrix Analysis and Applications*, 36(1):110–137, 2015.

Hyvärinen, A., Karhunen, J., and Oja, E. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

Kariya, T. and Kurata, H. *Generalized least squares*. John Wiley & Sons, 2004.

Li, P., Hastie, T. J., and Church, K. W. Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006.

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

Mahoney, M. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3 (2):123–224, 2011.

Matsumoto, M. and Nishimura, T. Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Transactions on Modeling and Computer Simulation*, 1998.

Mroueh, Y., Marcheret, E., and Goel, V. Co-occuring directions sketching for approximate matrix multiply. *arXiv preprint arXiv:1610.07686*, 2016.

Papailiopoulos, D., Kyrillidis, A., and Boutsidis, C. Provable deterministic leverage score sampling. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 997–1006. ACM, 2014.

Qi, H. and Hughes, S. M. Invariance of principal components under low-dimensional random projection of the data. In *Image Processing*. IEEE, 2012.

Schäfer, J. and Strimmer, K. An empirical bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764, 2005.

Shi, T., Tang, D., Xu, L., and Moscibroda, T. Correlated compressive sensing for networked data. In *UAI*, pp. 722–731, 2014.

Srisooksai, T., Keamarungsi, K., Lamsrichan, P., and Araki, K. Practical data compression in wireless sensor networks: A survey. *Journal of Network and Computer Applications*, 35(1):37–59, 2012.

Tropp, J. A. Improved analysis of the subsampled randomized hadamard transform. *Advances in Adaptive Data Analysis*, 3(01n02):115–126, 2011.

Tropp, J. A. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8(1-2):1–230, 2015.

Tulino, A. M. and Verdú, S. *Random matrix theory and wireless communications*, volume 1. Now Publishers Inc, 2004.

Woodruff, D. P. Sketching as a tool for numerical linear algebra. *arXiv preprint arXiv:1411.4357*, 2014.

Wu, S., Bhojanapalli, S., Sanghavi, S., and Dimakis, A. Single pass pca of matrix products. In *Advances In Neural Information Processing Systems*, pp. 2577–2585, 2016.

Zou, H., Hastie, T., and Tibshirani, R. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.