Applied Intelligence (2022) 52:3103–3117
https://doi.org/10.1007/s10489-021-02563-4

OGM: Online gaussian graphical models on the fly

Sijia Yang[1] · Haoyi Xiong[2] · Yunchao Zhang[3] · Yi Ling[3] · Licheng Wang[1] · Kaibo Xu[4] · Zeyi Sun[4]

Accepted: 24 May 2021 / Published online: 29 June 2021

Abstract
Gaussian Graphical Model is widely used to understand the dependencies between variables from high-dimensional data and can enable a wide range of applications such as principal component analysis, discriminant analysis, and canonical analysis. With respect to the streaming nature of big data, we study a novel Online Gaussian Graphical Model (OGM) that can estimate the inverse covariance matrix over the high-dimensional streaming data, in this paper. Specifically, given a small number of samples to initialize the learning process, OGM first estimates a low-rank estimation of inverse covariance matrix; then, when each individual new sample arrives, it updates the estimation of inverse covariance matrix using a low-complexity updating rule, without using the past data and matrix inverse. The significant edges of Gaussian graphical models can be discovered through thresholding the inverse covariance matrices. Theoretical analysis shows the convergence rate of OGM to the true parameters is guaranteed under Bernstein-style with mild conditions. We evaluate OGM using extensive experiments. The evaluation results backup our theory.

Keywords Advanced analytics · Online learning over streaming data · Gaussian graphical models

1 Introduction

Gaussian graphical models [1] have been widely used to discover the causalities/dependencies between variables from high-dimensional data. Given a branch of high-dimensional data, the Gaussian graphical model is frequently estimated as the inverse covariance matrix [2]. For example, to understand the neural connections between brain function areas, neuroscientists first deployed a large number of sensors to track the Magnetoencephalography (MEG) signals of different functions over time, then they estimated the sample covariance matrix from the collected data, where each dimension represents a specific function area. Further, a connectome graph representing the relationship of different function areas in the brain was constructed by sparse inverse covariance matrix estimation [3]. In the example, each edge in the connectome graph (i.e., non-zero entries in the inverse covariance matrix) represents a potential neural connection in the brain. In addition to the graph reconstruction, Gaussian graphical models also enabled a wide range of data analytical applications such as Principal Component Analysis (PCA) and Fisher's Discriminant Analysis (FDA) [4, 5]. Relevant work is well surveyed in [2].

For Gaussian graphical model estimation, existing work usually assumed the data samples have been already buffered on the server and proposed to estimate the graphical models using the buffered data in an offline manner (please see also in the recent survey [2]). However, the nature of big data is *streaming*—the data samples sequentially arrive for the real-time data analytics, while the graphical models are requested to update upon each newly arrived data sample with an online manner [6, 7]. Unfortunately, due to the computational complexity of matrix inverse and semidefinite programming, the existing solution is not capable of responding the sequentially

✉ Licheng Wang
wanglc@bupt.edu.cn

✉ Zeyi Sun
sunzeyi@mininglamp.com

1 School of Cyber Space Security, State key Laboratory of Switching and Networking, Beijing University of Posts and Telecommunications, Haidian, Beijing, China

2 Big Data Lab, Baidu Research, Baidu Inc., Haidian, Beijing, China

3 Department of Computer Science, Missouri University of Science and Technology, Rolla, Missouri, 95001, United States

4 Mininglamp Academy of Sciences, Mininglamp Technology, Shanghai, China

arriving data for online graphical model estimation or real-time model updating.

In this paper, with respect to the streaming nature of big data, we propose a novel Online Graphical Model, namely OGM that can estimate the inverse covariance matrix from the sequentially arriving high-dimensional data samples without any assumptions on sparsity. Specifically, given a small number (denoted as k) of buffered samples to cold-start the algorithms, OGM first computed a low-rank estimation of the inverse covariance matrix [8] using the buffered data; then, when each individual new sample arrives, it updates the inverse covariance matrix using a low-complexity updating rule, without using any past data and matrix inverse. Theoretical analysis shows that, with p-dimensional multivariate data, the complexity of OGM graphical model updating is as low as $\mathcal{O}(p^2)$. With n sequentially arrived data samples, OGM can attain a $\mathcal{O}(\sqrt{\log p/n} + \log p/n + k/n)$ statistical rate of convergence in the normalized spectral-norm. In addition, we also provide a low-complexity inference tool to discover the significantly directed dependencies between variables.

We evaluate OGM using extensive experiments based on both synthesized and real-world datasets. The evaluation results show that, compared to the baseline algorithms, OGM on average consumed 1,000 times less computational time to update the model with a newly arriving sample, while it ensures the lowest estimation error in terms of inverse covariance matrix estimation. Further, we evaluate OGM in the settings of online linear classification, where we find the online Fisher's linear discriminant analysis based on the fact that OGM can perform as good as the offline algorithms, while consuming a significantly lower cost for training/updating. In summary, we make contributions as follow.

- We study the problem of online Gaussian graphical model learning and inference, in order to estimate the inverse covariance matrix from the sequentially arriving high-dimensional data samples and make inference to recover the dependencies between variables in the dataset.
- We propose three algorithms for online Gaussian graphical models, including OGM initialization algorithm, OGM online updating algorithm, and OGM inference algorithm. We also analyze the algorithms from theoretical aspects and prove the statistical convergence of OGM algorithms.
- We carry out extensive experiments to evaluate OGM and compare with state of the art algorithms. The experiment results demonstrate that OGM works well under mild conditions and outperforms the baseline algorithms, in a wide range of Gaussian graphical model applications, such as online inverse covariance matrix estimation, online linear discriminant analysis for classification, and online dependency/causality discovery.

The rest of the paper is organized as follows. Section 2 presents the backgrounds and related work and defines the problem studied in this paper. Section 3 presents the framework of OGM and the three algorithms, i.e., OGM initialization, OGM online updating, and OGM inference, with theoretical interpretation. Section 4 analyzes the proposed algorithms from the statistical convergence perspectives. Section 5 demonstrates the advantages of our proposed algorithms based on a prototype system. Section 6 discusses the future work and concludes the paper.

2 Preliminaries and problem setups

2.1 Backgrounds and related work

Gaussian graphical models [9, 10] have been widely used to discover the causalities or dependencies between variables of high-dimensional data and further enable a wide-range data analytical applications. For example, through regularizing the inverse covariance matrix estimation, the linear covariance models have been used to enable the early detection of diseases based on Electronic Health Records (EHR) data [11–13] and dissect the genetic-basis of diseases using genome expression Quantitative Trait Loci (eQTL) data [14, 15] through reconstructing the conditional independences between diseases and gene expression.

The traditional sample-based (inverse) covariance matrix estimators, however, frequently suffer performance degradation under high-dimensional low sample size (HDLSS) settings. Most of the existing work [16–26] assumed a sparsity-induced approach, where the (inverse) covariance matrix was estimated with specific regularization, such as ℓ_1-norm regularization for GLasso [17]. The classical sample-based estimation can be improved, by incorporating the regularization terms, via structure risk minimization [27] principles. Note that, from Bayesian inference point of view [28, 29], the regularized estimation is explained as a Maximum A Posterior probability (MAP) procedure, where regularization term provides prior probability distribution of the (inverse) covariance matrices. For example, ℓ_1-norm regularization can be interpreted as a prior based on Laplace distribution [28]. To the best of our knowledge, all above models are based on offline estimation, which assumes that data samples have been obtained prior to the estimation.

The dependencies discovered from *ill-estimated* graphical models might not be reproducible, with high false discovery rate [30]. Further, the estimation error also affects the performance of graph-based machine learning algorithms [5, 31, 32]. Some algorithms on the topic of online learning of causal models [33] and covariance matrix estimator for the class of structural equation models [34] have been proposed while they do not consider high-dimension settings. To improve the graph estimation under high-dimension settings, authors in [16–26] proposed a group of regularized estimators that recover the graph from high-dimensional data with a certain structure assumption (e.g., sparsity and/or maximal degree). Authors in [35–37] established guaranteed convergence rate for inverse covariance matrix and iterative learning control while they can not support the streaming nature of big data.

2.2 Problem formulation

Suppose a set of p-dimensional data samples $X_1,\ X_2,\ X_3, \ldots$ sequentially arrives for data analytics. We assume every sample $X_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,p})^\top$ is independently and identically drawn from a multivariate distribution with p (possibly) correlated random variables $\{V_1, ..., V_p\}$. We further assume all samples $X_1,\ X_2,\ X_3, \ldots$ are *i.i.d* from the same p-dimensional random vector X. We define the distribution covariance matrix $\Sigma^* = \mathbb{E}\ (XX^\top)$, while the inverse covariance matrix is denoted as $\Theta^* = \Sigma^{*-1}$.

Problem With the sequential arrival of $X_1,\ X_2,\ X_3, \ldots, X_n$, our problem is to propose an online estimator of the inverse covariance matrix $\widehat{\Theta}_n$ using the n samples to approximate Θ^*. Specifically, with respect to the complexity, we have following two constraints:

- To cold-start the inverse covariance matrix estimation, the proposed estimator is allowed to buffer the first k samples. In this case, the estimator starts producing the first inverse covariance matrix estimation until $n = k$.
- With a new data sample arrival (e.g., X_n—the n^{th} sample and $n \geq k+1$), the estimator should update the previously estimated inverse covariance matrix $\widehat{\Theta}_{n-1}$ using X_n to obtain $\widehat{\Theta}_n$ without matrix inverse and past data.

In this paper, we introduce a set of novel estimation algorithms–OGM that enables online graphical model estimation (inverse covariance matrix) meeting above two constraints, using the sequential arrival data $X_1, X_2, \ldots$. Specifically, when a new sample X_n arrives, OGM updates the estimation of inverse covariance matrix from $\widehat{\Theta}_{n-1}$ to $\widehat{\Theta}_n$ with extremely low complexity.

3 OGM: Online gaussian graphical model estimation and inference system

This section introduces three algorithms for Online Graphical Model initialization, online updates, and the inference respectively.

As shown in Fig. 1, the framework of the OGM contains three modules. First, the model is initialized by the initialization module based on k buffered samples, where Algorithm 1 is implemented. Then, when the streaming data arrives, the model is updated within the model update module based on Algorithm 2. Finally, the inference module is exploited to infer the dependencies of the data using Algorithm 3.

3.1 OGM initialization algorithm

Given the first k samples $X_1,\ X_2, \ldots, X_k$ buffered, OGM is used to initialize the graphical model via Eigenvalue Decomposition (EVD) for inverse covariance estimation. As shown in Algorithm 1, given input parameter λ', the algorithm first computes the sample scatter matrix (denoted as $\sum_{l=1}^{k} X_l X_l^\top$), then leverages EVD to approximate the inverse of sample scatter matrix.

Algorithm 1 OGM Initialization Algorithm: Getting $\widehat{T}_k$ using the first k samples and parameter λ'.

1: **Input** initial samples $X_1,\ X_2, \ldots, X_k$, and a tunning parameter λ';
2: **Output** the initial graphical model $\widehat{\Theta}_k$;
3: $\mathbf{U}\mathbf{D}\mathbf{U}^\top \leftarrow \text{EVD}\left(\sum_{l=1}^{k} X_l X_l^\top + k \cdot \lambda' \cdot \mathbf{I}_{p\times p}\right)$;
4: $\mathbf{Q} \leftarrow \mathbf{0}_{p\times p}$;
5: **for** $1 \leq i \leq p$ **do**
6: $\quad \mathbf{Q}_{i,i} \leftarrow 1.0/\mathbf{D}_{i,i}$;
7: **end for**
8: $\widehat{T}_k \leftarrow \mathbf{U}\mathbf{Q}\mathbf{U}^\top$;
9: **Return** $\widehat{T}_k$.

3.1.1 Understanding Algorithm 1

Specifically, the algorithm uses λ' to ensure the positive definiteness of $\left(\sum_{l=1}^{k} X_l X_l^\top + k \cdot \lambda' \cdot \mathbf{I}_{p\times p}\right)$ through improving the eigenvalues. We thus make an assumption as follow.

Assumption 1 *Let us denote the smallest eigenvalue as* $\lambda_{min}(\sum_{l=1}^{k} X_l X_l^\top)$. *We assume that the first* k *buffered samples and the tuning factor* λ' *should be set appropriately, with respect to a certain number* $\mathcal{C}$, *such that*

$$\max\{-\lambda_{min}\left(\sum_{l=1}^{k} X_l X_l^\top\right)/k, 0\} < \lambda' < \frac{\mathcal{C}}{\sqrt{p}} \tag{1}$$

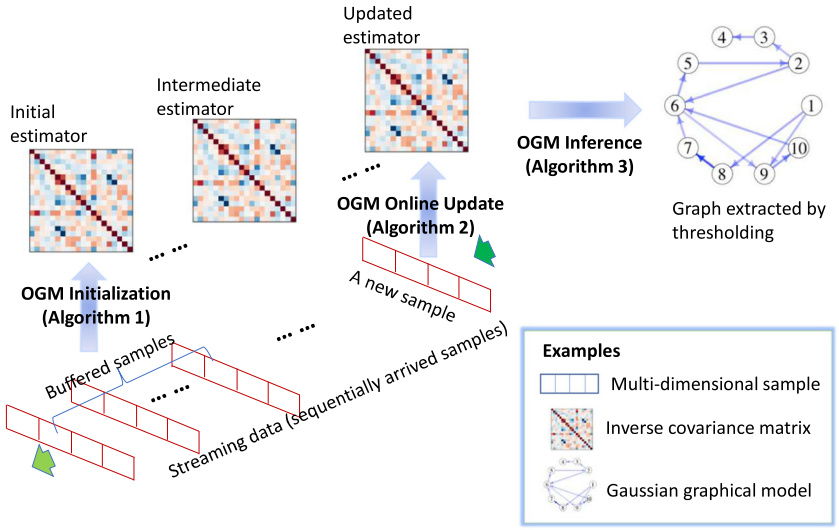


Fig. 1 The framework of OGM and the key algorithms over streaming data – (I) OGM Initialization: estimate a low-rank inverse covariance matrix using a small number of buffered samples; OGM Online Update: iterate the estimated inverse covariance matrix using the new sample and low-complexity updating rules; and (III) OGM Inference: extracting the significant edges of the dependency graph from the estimated inverse covariance matrices through thresholding

to make a well-posed and positive definite estimation. When k is relatively large, then $\sum_{l=1}^{k} X_l X_l^\top$ tends to positive semi-definite and $\max\{-\lambda_{min}\left(\sum_{l=1}^{k} X_l X_l^\top\right)/k, 0\} \to 0$. *In this case, any positive number $\mathcal{C}$ could generate an appropriate tuning factor when p is fixed. Note that $\mathcal{C}$ is critical to the convergence analysis.*

Lines 3–8 of Algorithm 1 compute a matrix inverse approximation based on EVD. Algorithm 1 returns

$$\widehat{T}_k = \left(\sum_{l=1}^{k} X_l X_l^\top + k \cdot \lambda' \cdot \mathbf{I}_{p\times p}\right)^{-1} \tag{2}$$

as the approximation of inverse sample covariance matrix based on the first k samples $X_1, X_2, \ldots, X_k$. We use $\widehat{\Theta}_k = (k \cdot \widehat{T}_k)$ as the initialization of estimator. As a result, We thus has the theorem as follow.

Theorem 1 *With $\lambda' > 0$, the matrix $\widehat{T}_k$ is invertible, symmetric and positive definite.*

Proof For any $1 \le l \le k$, the matrix $\sigma_l = X_l X_l^\top$ should be symmetric. Then the matrix,

$$\Sigma_k = \sum_{l=1}^{k} \sigma_l = \sum_{l=1}^{k} X_l X_l^\top$$

should be symmetric. As Σ_k is a real symmetric matrix, we can write $\Sigma_k = USU^\top$ through eigenvalue decomposition, where the matrix U is a matrix of orthogonal vectors and S is a diagonal matrix of singular values. Thus, we can simply write

$$\widehat{T}_k^{-1} = \sum_{l=1}^{k} X_l X_l^\top + k \cdot \lambda' \cdot \mathbf{I}_{p\times p} = U(S + k \cdot \lambda' \cdot \mathbf{I}_{p\times p})U^\top,$$

which is symmetric. Thus, when $\lambda' > 0$, the diagonal values in $S + k \cdot \lambda' \cdot \mathbf{I}_{p\times p}$ should be all positive. Thus, in this condition, $U(S + k \cdot \lambda' \cdot \mathbf{I}_{p\times p})U^\top$ is invertible and $\widehat{T}_k$ exists as follow

$$\widehat{T}_k = \left(U(S + k \cdot \lambda' \cdot \mathbf{I}_{p\times p})U^\top\right)^{-1} = U(S + k \cdot \lambda' \cdot \mathbf{I}_{p\times p})^{-1}U^\top.$$

Furthermore, $\widehat{T}_k^{-1} = \sum_{l=1}^{k} X_l X_l^\top + k \cdot \lambda' \cdot \mathbf{I}_{p\times p}$ should be positive-definite, while its inverse matrix $\widehat{T}_k$ should be also positive-definite. □

3.2 OGM online updating algorithm

Given the n^{th} arrival data sample X_n for online graphical model updating (suppose n is indexed from $k+1$ such as $n = k+1, k+2, \ldots$), OGM updates the previous graphical model $\widehat{T}_{n-1}$ and produces the new model $\widehat{T}_n$ using a low-complexity updating policy listed in Algorithm 2. Please note that, when $n = k+1$, the previous graphical model updated by Algorithm 2 indeed updates the current model $\widehat{T}_{n-1} = \widehat{T}_k$ (the initial model). Note that, OGM uses $\widehat{\Theta}_n = (n \cdot \widehat{T}_n)$ as an online estimator of Θ^* based on samples $X_1, X_2, \ldots, X_n$ (where first k samples are used for algorithm initialization).

Algorithm 2 OGM Online Update Algorithm: Producing $\widehat{T}_n$ through Updating $\widehat{T}_{n-1}$ with the n^{th} arriving sample X_n.

1: **Input** the current model $\widehat{T}_{n-1}$ and new data X_n;
2: **Output** the updated model $\widehat{T}_n$;
3: $Y \leftarrow \widehat{T}_{n-1} X_n$
4: $\gamma_n \leftarrow 1/\left(1 + \text{trace}(X_n Y^\top)\right)$;
5: $\Theta_n \leftarrow \widehat{T}_{n-1} - \gamma_n \cdot Y Y^\top$;
6: $\widehat{T}_n \leftarrow \frac{1}{2}(\Theta_n + \Theta_n^\top)$.
7: **Return** $\widehat{T}_n$.

3.2.1 Understanding Algorithm 2

The updating rule is derived from the corollary [38] as follows.

Corollary 1 (Inverse of Two Matrices Sum [38]) *Given two matrices A and B, where A is invertible and* $\text{rank}(B) = 1$,

$$(A + B)^{-1} = A^{-1} - \frac{1}{1 + \text{trace}(BA^{-1})} A^{-1} B A^{-1}, \tag{3}$$

if $(A + B)^{-1}$ *exists.*

Before, connecting **Corollary. 1** with the theorem, we would like to prove some lemmas as follow.

Lemma 1 *For any nonzero input sample* X_n*, the matrix* $X_n X_n^\top$ *should be a rank-1 matrix, i.e.,* $\text{rank}(X_n X_n^\top) = 1$.

Proof First of all, for any nonzero matrix the rank should be a positive integer. Then, we have $\text{rank}(X_n X_n^\top) \leq \text{rank}(X_n) = 1$ as X_n is a $p \times 1$ matrix. Thus $\text{rank}(X_n X_n^\top) = 1$ □

Lemma 2 *For* $n \geq k + 1$*, the matrix* $\widehat{T}_{n-1}$ *in Algorithm 2 is always symmetric.*

Proof When $n = k + 1$, then $\widehat{T}_{n-1} = \widehat{T}_k$ is the output of Algorithm 1 and $\widehat{T}_k = \mathbf{UQU}^\top$ is symmetric. Later, when $n \geq k + 2$, $\widehat{T}_{n-1}$ should be the output of Algorithm 2 for the previous step. Then, line 6 of Algorithm 2 ensures the symmetric properties. □

Lemma 3 *When* $n = k + 1$*, the matrix* $\widehat{T}_{n-1}$ *in Algorithm 2 is invertible. In the same setting.*

$$\widehat{T}_n = \left(\widehat{T}_{n-1}^{-1} + X_n X_n^\top\right)^{-1}, \tag{4}$$

if $(\widehat{T}_{n-1}^{-1} + X_n X_n^\top)^{-1}$ *exists.*

Proof When $n = k + 1$, $\widehat{T}_{n-1} = \widehat{T}_k = \mathbf{UQU}^\top$. Considering the lines 3–7 in Algorithm 1, we can obtain that $\widehat{T}_k$ is invertible and $\widehat{T}_{n-1}^{-1} = \widehat{T}_k^{-1} = \mathbf{UDU}^\top$. Line 3 of Algorithm 2 sets $Y = \widehat{T}_{n-1}$ while line 4 defines

$$\gamma_n = \frac{1}{1 + \text{trace}(X_n Y^\top)} = \frac{1}{1 + \text{trace}(X_n X_n^\top \widehat{T}_{n-1})}$$

as $\widehat{T}_n^\top = \widehat{T}_n$. Since $\text{rank}(X_n X_n) = 1$ and $\widehat{T}_{n-1}$ is invertible, we can use Corollary 1 to obtain that for $n = k + 1$

$$\Theta_n = \left(\widehat{T}_{n-1}^{-1} + X_n X_n^\top\right)^{-1}, \tag{5}$$

if $(\widehat{T}_{n-1}^{-1} + X_n X_n^\top)^{-1}$ exists. Further, as $\widehat{T}_{n-1}^{-1} = \widehat{T}_k^{-1} = \mathbf{UDU}^\top$ is symmetric and $X_n X_n^\top$ is symmetric, we can conclude $\widehat{T}_{n-1}^{-1} + X_n X_n^\top$ an symmetric matrix. Then, the inverse of such symmetric matrix Θ_n should be symmetric. In this way, $\widehat{T}_n = 1/2(\Theta_n + \Theta_n) = \Theta_n$. □

Based on above lemmas, we can derive the theorem as follows.

Theorem 2 *For* $n \geq k + 1$ $\widehat{T}_n = \Theta_n$ *and*

$$\widehat{T}_n = \left(\widehat{T}_{n-1}^{-1} + X_n X_n^\top\right)^{-1}, \tag{6}$$

if $(\widehat{T}_{n-1}^{-1} + X_n X_n^\top)^{-1}$ *and* $\widehat{T}_{n-1}^{-1}$ *both exist, as the recursive rule for incremental updates.*

Proof We have already proved the case for $n = k + 1$. For the case that $n \geq k + 2$, we use the Corollary. 1 to prove the theorem recursively. More specific, for any $n' \geq k + 2$ with the non-zero input $X_{n'}$ and the current model $\widehat{T}_{n'-1}$, suppose the preconditions that (P1) $\widehat{T}_{n'-1}$ is invertible and symmetric and (P2) the inverse $(\widehat{T}_{n'-1}^{-1} + X_{n'} X_{n'}^\top)^{-1}$ exists are given. We can draw the conclusion that (C1) $\Theta_{n'} = (\widehat{T}_{n'-1}^{-1} + X_{n'} X_{n'}^\top)^{-1}$ and (C2) $\Theta_{n'}$ is symmetric (inverse of a symmetric matrix is symmetric and the sum of symmetric matrix is symmetric) and (C3) $\widehat{T}_{n'} = \Theta_{n'}$. In this way, the consequences obtained for $\widehat{T}_{n'}$ case further establish the preconditions for $\widehat{T}_{n'+1}$ case, while we already prove that the preconditions hold for $n' = k + 2$. □

Lemma 4 *Using the recursion and Theorems. 1 and 2, we can derive that*

$$\widehat{T}_n = \left(\widehat{T}_k^{-1} + \sum_{l=k+1}^{n} X_l X_l^\top\right)^{-1} = \left(\sum_{l=1}^{n} X_l X_l^\top + k \cdot \lambda' \cdot \mathbf{I}_{p \times p}\right)^{-1}, \tag{7}$$

With this lemma, we can understand Algorithm 2 as an online approximation to the inverse of sample covariance matrix $\bar{\Sigma} = 1/n \sum_{l=1}^{n} X_l X_l^\top$*, where* $\sum_{l=1}^{n} X_l X_l^\top = n\bar{\Sigma}$ *demonstrates the covariance structure of data and* $k \cdot \lambda' \cdot \mathbf{I}_{p \times p}$ *ensures the positive definite and non-singularity by the initialization.*

Note that above lemma is critically important in the convergence analysis of the algorithm in Section 4.

3.2.2 Updating complexity

Note that the overall computational complexity of Algorithm 2 is bounded by $\mathcal{O}(p^2)$, where p refers to the dimension of data sample. The time complexity to compute line 3 of Algorithm 2, i.e., a $p \times p$ matrix multiplying a $p \times 1$

matrix, is $\mathcal{O}(p^2)$, while the time consumption of line 4 is $\mathcal{O}(p)$, due to the trace calculation for a $p \times p$ matrix. Finally, the complexity of line 5 is $\mathcal{O}(p^2)$.

With above two algorithms, OGM can solve the online graphical model estimation problem. With any n sequential arrival samples, we use $\widehat{\Theta}_n = (n \cdot \widehat{T}_n)$ as an estimator of Θ^* with provably guarantee of statistical convergence. The analysis of proposed algorithms are introduced in Section of Algorithm Analysis.

3.3 OGM inference algorithm

Given the online graphical model estimation $\widehat{T}_n$ and a confidence interval for significance thresholding denoted as α (by default $\alpha = 0.05$), Algorithm 3 extracts the directed graph representing the dependencies between variables through thresholding, then generates the adjacent matrix of the graph (denoted as $\mathcal{G}_n$). When $(\mathcal{G}_n)_{i,j} = 1.0$, OGM suggests a potential dependency from variable V_i to V_j (i.e., V_i depends on V_j significantly); on the other hand, when $(\mathcal{G}_n)_{i,j} = 0.0$, the two variables are conditionally independent from V_i to V_j. This algorithm is adapted based on the theory proposed in Section 3.2 in [26] and Section 6 in [39].

Algorithm 3 OGM Inference Algorithm: Extracting the directed dependency graph $\mathcal{G}_n$ using the online estimator $\widehat{T}_n$.

1: **Input** the Online Graphical Model $\widehat{T}_n$, number of dimensions p, index of data n, and confidence interval α;
2: **Output** the adjacent matrix $\mathcal{G}$ of the directed graph that represents dependencies
3: $\mathcal{G}_n \leftarrow \mathbf{0}_{p\times p}$;
4: $\rho \leftarrow \Phi^{-1}\left(1 - \frac{\alpha}{p(p-1)}\right)/\sqrt{n}$
5: **for** $i = 0$ to p **do**
6: **for** $j = 0$ to p **do**
7: $thr \leftarrow \rho \cdot \sqrt{((\widehat{T}_n)_{i,j})^2 - ((\widehat{T}_n)_{i,i})^2}$;
8: **if** $\widehat{T}_{i,j} \geq thr$ **then**
9: $(\mathcal{G}_n)_{i,j} \leftarrow 1.0$;
10: **end if**
11: **end for**
12: **end for**
13: **Return** $\mathcal{G}_n$.

4 Algorithm analysis

In this section, we present the analytical results of OGM as follow: we first present the preliminaries and key assumptions of our work; then we provide the analytical results on the statistical convergence of OGM with sketched proofs.

4.1 Assumptions and lemmas

Prior to introducing the theoretical analysis, we would like to first present and discuss one key assumption that helps us bound the norms and variances. The assumption is as follows.

Assumption 2 *Suppose n samples $X_1, X_2, \ldots, X_n$ are i.i.d from the same random vector X. We assume the random vector X is zero-mean and there exists a positive number C_1 that upper-bounds the ℓ_2-norm of X, i.e.,*

$$\mathbb{E}(X) = 0 \ \text{ and } \ \|X\|_2^2 \leq C_1. \tag{8}$$

As X is zero-mean, we then estimate the population covariance $\Sigma^* = \mathbb{E}(XX^\top)$. We further assume there exists a positive number C_2 that lower-bounds the eigenvalues of Σ^* such as

$$\lambda_{min}(\Sigma^*) \geq 1/C_2. \tag{9}$$

Above assumption has also been made in a series of work [19, 40, 41] to achieve the similar goals. Based on the above assumptions, we can derive a lemma as follows.

Lemma 5 *We can easily use Assumption 2 to obtain the upper bound of eigenvalue of Σ^*, such that*

$$\lambda_{max}(\Sigma^*) \leq \text{trace}(\mathbb{E}(XX^\top)) \leq \mathbb{E}|X|_2^2 \leq C_1. \tag{10}$$

Lemma 6 *In the same way, for any n samples $X_1, X_2, X_3 \ldots, X_n$ i.i.d from random vector X, let us define the sample estimation of covariance matrix as*

$$\bar{\Sigma}_n = \frac{1}{n}\sum_{l=1}^{n} X_l X_l^\top.$$

Then, we can also use Assumption 2 to obtain the upper bound of eigenvalues of $\bar{\Sigma}_n$, such that

$$\lambda_{max}(\bar{\Sigma}_n) \leq \text{trace}\left(\frac{1}{n}\sum_{l=1}^{n} X_l X_l^\top\right) \leq C_1. \tag{11}$$

Lemma 7 *Let us denote the population inverse covariance matrix as $\Theta^* = \Sigma^{*-1}$. Then, we can use **Assumption 1** and above lemmas to obtain the upper/lower bounds eigenvalues, such that*

$$\begin{aligned} 1/C_1 \leq \frac{1}{\lambda_{max}(\Sigma^*)} &= \lambda_{min}(\Theta^*) \\ &\leq \lambda_{max}(\Theta^*) = \frac{1}{\lambda_{min}(\Sigma^*)} \leq C_2. \end{aligned} \tag{12}$$

4.2 Preliminary results and corollaries

With above lemmas in mind, we present the key preliminary results [41] related to the statistical convergence of sample covariance matrix estimator.

Corollary 2 (Sample Covariance Matrix Asymptotic Rate [41]) *Given n independent samples $X_1, X_2, \ldots, X_n$ i.i.d from the random vector X and distribution covariance matrix $\Sigma^* = \mathbb{E}\, XX^\top$, we define the sample covariance matrix estimator as $\bar{\Sigma}_n = \sum_{l=1}^{n} X_l X_l^\top$. The sample estimator enjoys a statistical rate of convergence in spectral-norm as:*

$$\|\bar{\Sigma}_n - \Sigma^*\|_2 = \mathcal{O}_p\left(\sqrt{\frac{\log p}{n}} + \frac{\log p}{n}\right). \tag{13}$$

*The constants in (13) all depend on C_2 and C_1. The proof of **Corollary 2**, under **Assumptions 2** has been provided in [41] based on Bounds for Sums of Independent Random Matrices and Matrix Bernstein Inequality.*

4.3 Analytic results and proofs

Here, we present the statistical convergence of OGM as follows.

Theorem 3 (Main Result: Normalized Spectral-norm Statistical Rate of Convergence) *Let us denote the true inverse covariance matrix $\Theta^* = \Sigma^{*-1} - \left(\mathbb{E}(XX^\top)\right)^{-1}$. With appropriate settings of the k first buffered samples and the tuning factor λ', the online estimator $\widehat{\Theta}_n = (n \cdot \widehat{T}_n)$ (and $\widehat{T}_n$ is the output of Algorithm 2) converges to Θ^* in a statistical rate as follows.*

$$\frac{\|\widehat{\Theta}_n - \Theta^*\|_2}{\|\widehat{\Theta}_n\|_2} = \mathcal{O}_p\left(\sqrt{\frac{\log p}{n}} + \frac{\log p + k}{n}\right). \tag{14}$$

Proof The proof of **Theorem.** 3 can be outlined as follows. Consider the Cauchy–Schwarz inequality, triangle inequality, and **Assumption 1**, we can derive the estimation of spectral-norm statistical rate of convergence as follows:

$$\begin{aligned}\frac{\|\widehat{\Theta}_n - \Theta^*\|_2}{\|\widehat{\Theta}_n\|_2} &= \frac{\|\widehat{\Theta}_n(\Sigma^* - \widehat{\Theta}_n^{-1})\Theta^*\|_2}{\|\widehat{\Theta}_n\|_2} \le \|\Theta^*\|_2 \|\Sigma^* - \widehat{\Theta}_n^{-1}\|_2 \\ &\le C_1 \cdot \left(\|\Sigma^* - \bar{\Sigma}_n\|_2 + \|\bar{\Sigma}_n - \widehat{\Theta}_n^{-1}\|\right).\end{aligned} \tag{15}$$

The original problem can be reduced to the estimation of (i): $\|\Sigma^* - \bar{\Sigma}_n\|_2$, and (ii): $\|\bar{\Sigma}_n - \widehat{\Theta}_n^{-1}\|$. First, we have already introduced *Corollary 2* to obtain the upper bound of $\|\Sigma^* - \bar{\Sigma}_n\|_2$ based on Matrix Bernstein Inequality. Then, we present **Lemma 4** to analyze $\widehat{\Theta}_n^{-1}$, and derive the upper bound of $\|\bar{\Sigma}_n - \widehat{\Theta}_n^{-1}\|$. Such that

$$\begin{aligned}\widehat{\Theta}_n^{-1} &= \left(n \cdot \widehat{T}_n\right)^{-1} = \frac{1}{n} \cdot \widehat{T}_k^{-1} + \frac{1}{n} \sum_{l=k+1}^{n} X_l X_l^\top \\ &= \bar{\Sigma}_n + \frac{k \cdot \lambda'}{n} \cdot \mathbf{I}_{p\times p}.\end{aligned} \tag{16}$$

With appropriate setting of λ' as input parameters, there exists

$$\|\bar{\Sigma}_n - \widehat{\Theta}_n^{-1}\|_2 = \frac{k \cdot \lambda'}{n} \cdot \|\mathbf{I}_{p\times p}\|_2 = \mathcal{O}\left(\frac{k}{n}\right), \tag{17}$$

where $\|\mathbf{I}_{p\times p}\|_2 = \sqrt{p}$ and it is assumed that $\lambda' \le \mathcal{O}(1/\sqrt{p})$. With all above results combined, we can conclude that

$$\frac{\|\widehat{\Theta}_n - \Theta^*\|_2}{\|\widehat{\Theta}_n\|_2} = \mathcal{O}_p\left(\sqrt{\frac{\log p}{n}} + \frac{\log p + k}{n}\right). \tag{18}$$

□

Note that, though we have to make several assumptions to facilitate the proof of statistical convergence of OGM, the algorithms indeed work well under wider settings. For example, in our experiments, we evaluate OGM with fixed $\lambda' = 1.0$ (*Assumptions 1* not hold) using nonzero-mean/unbounded samples (*Assumption 2* not hold). OGM still performs well.

5 Experiments

In this section, we evaluate OGM using three experiments. We focus on demonstrating the theoretical properties of OGM for online inverse covariance matrix estimation, and further understand its performance on the inference and classification tasks. We realize a OGM prototype system, where we implement our proposed algorithms and baseline algorithms to demonstrate the advantage of our proposed algorithms.

5.1 Online graph estimation

To validate **Theorem 1**, we evaluate asymptotic properties of OGM using a synthesized dataset, where we compare OGM with baseline algorithms for estimation of the inverse covariance matrices. OGM estimates the graph in online manner, while TSVD and sample-based estimators gather all samples together and estimate inverse covariance matrices in an offline manner.

5.1.1 Experiment setups

We setup the datasets and baseline algorithms for experiments as follows.

Datasets Synthesis We first define a p-variate Gaussian distribution $\mathcal{N}(\mathbf{0}, \Sigma^*)$, where $\mathbf{0}$ refers to a p-dimensional vector with all elements zero. The covariance matrix Σ^* is a symmetric semi-definite matrix, where each element $(\Sigma^*)_{i,j} = 0.8^{|i-j|}$. We generate the synthesized datasets that are randomly drawn from $\mathcal{N}(\mathbf{0}, \Sigma^*)$, shuffle the datasets, and sequentially input to the graph estimators, so as to simulate the settings of online estimation.

Baseline Algorithms We also evaluate other high-dimensional inverse covariance matrix estimators, such as Truncated SVD [42] (TSVD) and the inverse of sample covariance matrix estimator (Sample), under the same settings. More specifically, these two algorithms are implemented as follows.

- *Truncated SVD (TSVD)* - Given k offline samples for initialization and $n - k$ online samples, this algorithm gathers all these samples together and forms $\mathbf{X} = [X_1, X_2, \ldots, X_k, X_{k+1}, \ldots, X_n]^\top$. TSVD then performs singular value decomposition $\mathbf{X} = \mathbf{LSR}^\top$, further carries out optimal singular value shrinkage [42] and obtain the top singular values/vectors $\mathbf{X} \approx \mathbf{L}'\mathbf{S}'\mathbf{R}'^\top$. Finally, TSVD estimates the inverse covariance matrix as $n\mathbf{R}'\mathbf{S}^{-2}\mathbf{R}'^\top$, which could be considered as a low-rank approximation to $(\frac{1}{n}\mathbf{X}^\top\mathbf{X})^{-1}$.
- *Sample-based Estimator* - Given k offline samples for initialization and $n - k$ online samples, this algorithm also gathers all these samples together and forms $\mathbf{X} = [X_1, X_2, \ldots, X_k, X_{k+1}, \ldots, X_n]^\top$. It simply estimates the sample-based inverse covariance matrix as $(\frac{1}{n}\mathbf{X}^\top\mathbf{X} + \frac{k\cdot\lambda'}{n}\mathbf{I}_{p\times p})^{-1}$. Note that this algorithm uses the same setting to the parameter λ' as OGM.

5.1.2 Experiment results

We evaluate the algorithms from following two aspects.

Estimation Error We compare OGM (i.e., $\widehat{\Theta}_n$) to $\Theta^* = \Sigma^{*-1}$ and measure the estimation error, i.e., $\|\widehat{\Theta}_n - \Theta^*\|_2$ and $\|\widehat{\Theta}_n - \Theta^*\|_\infty$, with increasing number of online data samples. Figure 2 presents the comparisons of ℓ_2-norm estimation errors between OGM, TSVD and sample-based estimator with varying number of dimensions $p = 20, 40, 80, 160, \ldots, 1280$, while Fig. 3 presents the comparisons of ℓ_∞-norm estimation errors. In each figure, we illustrate the trends of estimation error reduction with increasing number of (online) samples n. Specifically, the X-axis of each figure is $(n - k)$ referring to the number of data samples that sequentially arrive for online estimation, while k is set to $k = 20$ indicating the number of buffered samples for initialization. For example, when $(n-k) = 100$, OGM is initialized with 20 buffered samples and updated by 100 online samples sequentially; TSVD and sample-based estimator are estimated using the same 120 samples (i.e., the combination of aforementioned 20 buffered samples and 100 online samples) in an offline manner.

It is obvious that the estimation errors of all OGM and sample-based estimator reduces with increasing number of online samples $(n - k)$, while the ℓ_2-norm error of TSVD-based estimator seems to be constant. Obviously, the trends of estimation error reduction vs. the increasing n, shown in each figure, validates the ℓ_2-norm estimation error rate addressed in **Theorem 1**. Specifically, when p is fixed, the estimation error reduces at $\mathcal{O}(\sqrt{1/n})$ rate. Since we don't fix $\|\Theta^*\|_2$ to the constant with varying p, according to the assumption introduced in **Theorem 1**, it doesn't make sense to compare the estimation error between different p.

Computational Time In addition to the estimation error, we also evaluate the time consumption of OGM to update the inverse covariance matrix estimation per new sample (i.e., time consumption of Algorithm 2). Figure 4 presents the computational time consumed by each algorithm under various settings. It is obvious that the computational time of the offline estimators including sample estimator and TSVD needs linear increases with the number of samples arrived. Offline methods such as TSVD outperforms OGM to recover the inverse covariance matrix with lower estimation error, however, OGM consumes significantly less time. Moreover, the time consumption of OGM for each update only depends on the number of dimensions which support the computational complexity analysis.

5.2 Online classification

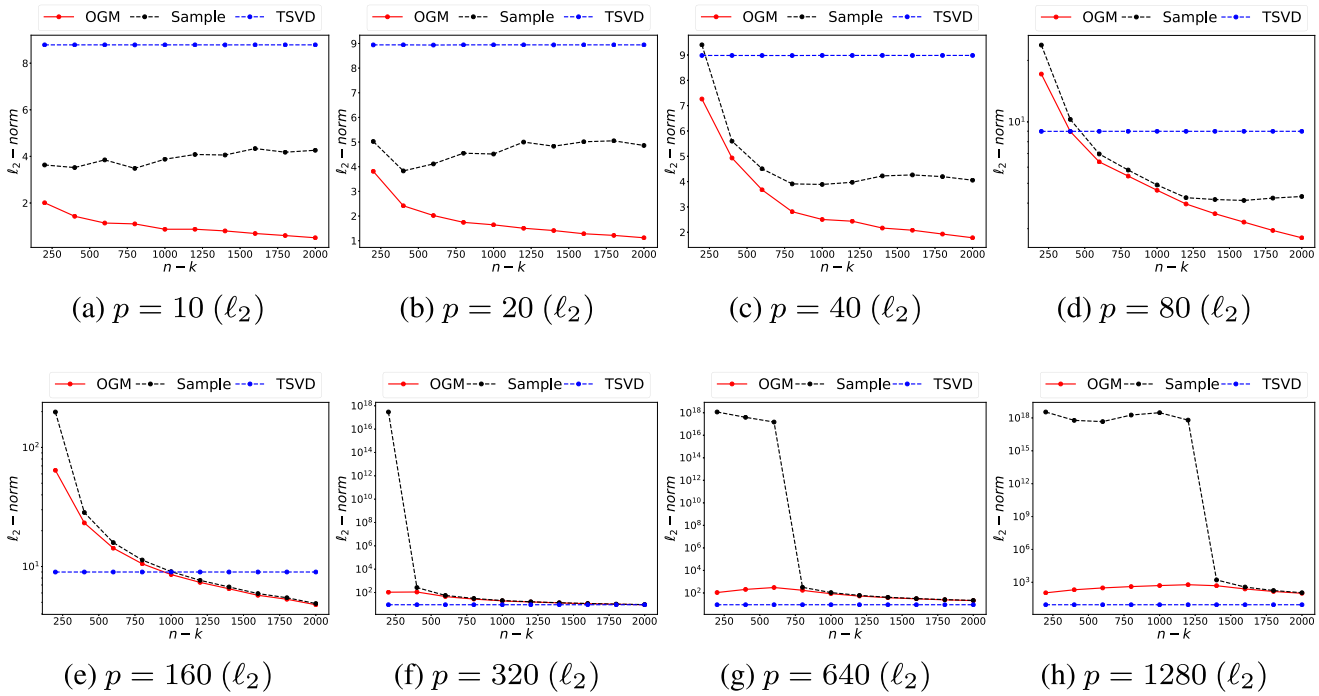
In this experiment, we validate the performance of estimated inverse covariance matrices through leveraging for classification tasks.

5.2.1 Experiment setups

Given the online graphical model estimators, we derive a simple online linear classifier models for binary classification. Given two multivariate Gaussian distributions $\mathcal{N}(\mu^+, \Sigma)$ and $\mathcal{N}(\mu^-, \Sigma)$ with different means μ^+/μ^- and the same covariance matrix Σ, we assume the sample x is i.i.d drawn from the two distributions with equal prior. The optimal classification rule here should be

$$\text{sign}\left(\left(x - \frac{\mu^+ + \mu^-}{2}\right)^\top \Sigma^{-1}(\mu^+ - \mu^-)\right), \tag{19}$$

where $\text{sign}(\cdot) \rightarrow \{\pm 1\}$. Specifically, we follow the experiment setups in [43]. The true covariance matrix is set as a $p \times p$ positive definite matrix with each entry



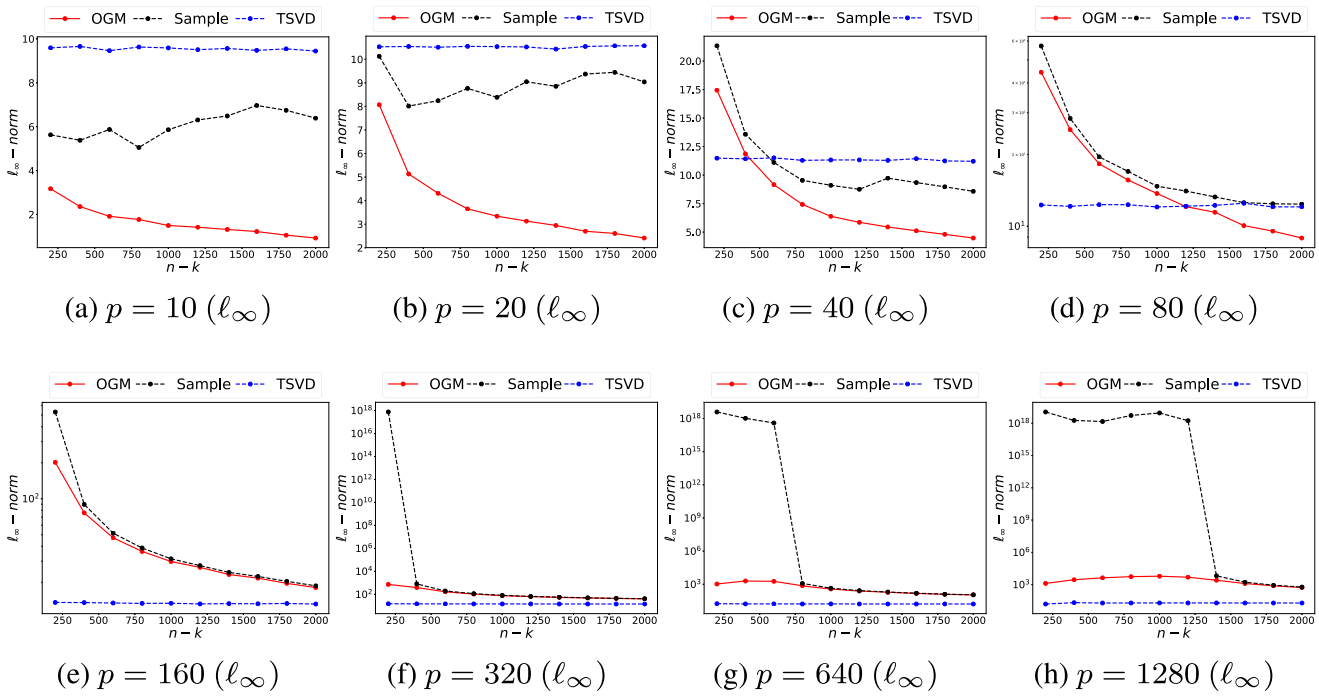
(a) $p = 10$ (ℓ_2) (b) $p = 20$ (ℓ_2) (c) $p = 40$ (ℓ_2) (d) $p = 80$ (ℓ_2)

(e) $p = 160$ (ℓ_2) (f) $p = 320$ (ℓ_2) (g) $p = 640$ (ℓ_2) (h) $p = 1280$ (ℓ_2)

Fig. 2 Overall Performance Comparison: ℓ_2-norm Estimation Error

$\Sigma_{i,j} = 0.8^{-|i-j|}$. The mean of positive class is set as $\mu^+ = (1, 1, \ldots, 1, 0, 0, \ldots, 0)$, i.e., first 10 elements are all set to 1 while rest elements are 0. Finally, the mean of the negative class is $\mu^- = (0, 0, \ldots, 0)$. Note that we perform two sets of experiments with $p = 800$ and $p = 1600$.

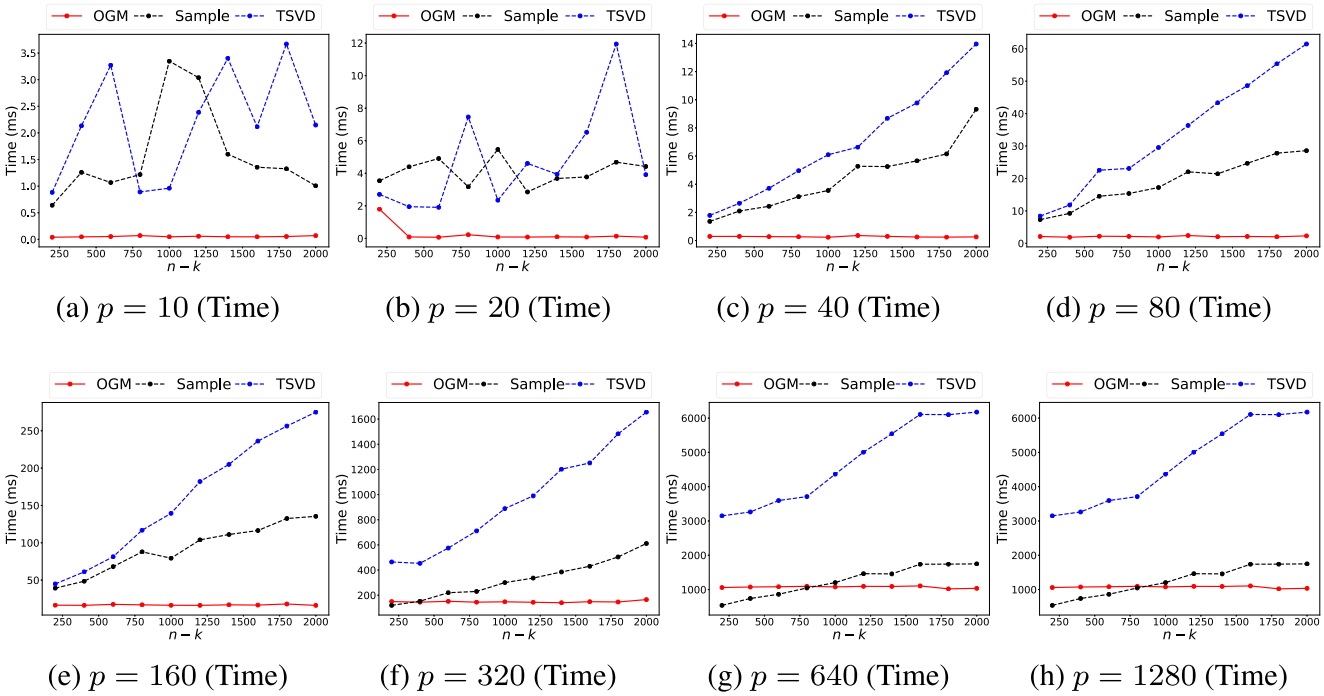
Given the online samples labeled by $+1$ or -1



(a) $p = 10$ (ℓ_∞) (b) $p = 20$ (ℓ_∞) (c) $p = 40$ (ℓ_∞) (d) $p = 80$ (ℓ_∞)

(e) $p = 160$ (ℓ_∞) (f) $p = 320$ (ℓ_∞) (g) $p = 640$ (ℓ_∞) (h) $p = 1280$ (ℓ_∞)

Fig. 3 Overall Performance Comparison: ℓ_∞-norm Estimation Error



(a) $p = 10$ (Time) (b) $p = 20$ (Time) (c) $p = 40$ (Time) (d) $p = 80$ (Time)

(e) $p = 160$ (Time) (f) $p = 320$ (Time) (g) $p = 640$ (Time) (h) $p = 1280$ (Time)

Fig. 4 Overall Time Consumption Comparison

that sequentially arrive for training, the binary Linear Discriminant Analysis (LDA) classifies the input x using the following rule:

$$\text{sign}\left(\left(x - \frac{\bar{\mu}_n^+ + \bar{\mu}_n^-}{2}\right)^\top \widehat{\Theta}_n(\bar{\mu}_n^+ - \bar{\mu}_n^-)\right), \tag{20}$$

where the vectors $\bar{\mu}_n^+$ and $\bar{\mu}_n^-$ refer to the online mean estimators for the samples labeled with $+1$ and -1 respectively, and $\widehat{\Theta}_n$ is OGM estimators based on the all online samples. With the rule in (20), an online fisher's linear discriminant analysis can be enabled. Note that traditional fisher's linear discriminant analysis is based on the inverse of between-class covariance matrix, while we use OGM instead. Our online classification rule in (20) can be considered an online approximation to the optimal rule in (19), using sequentially arrival training samples.

5.2.2 Experiment results

We compare the algorithms from two perspectives as follows.

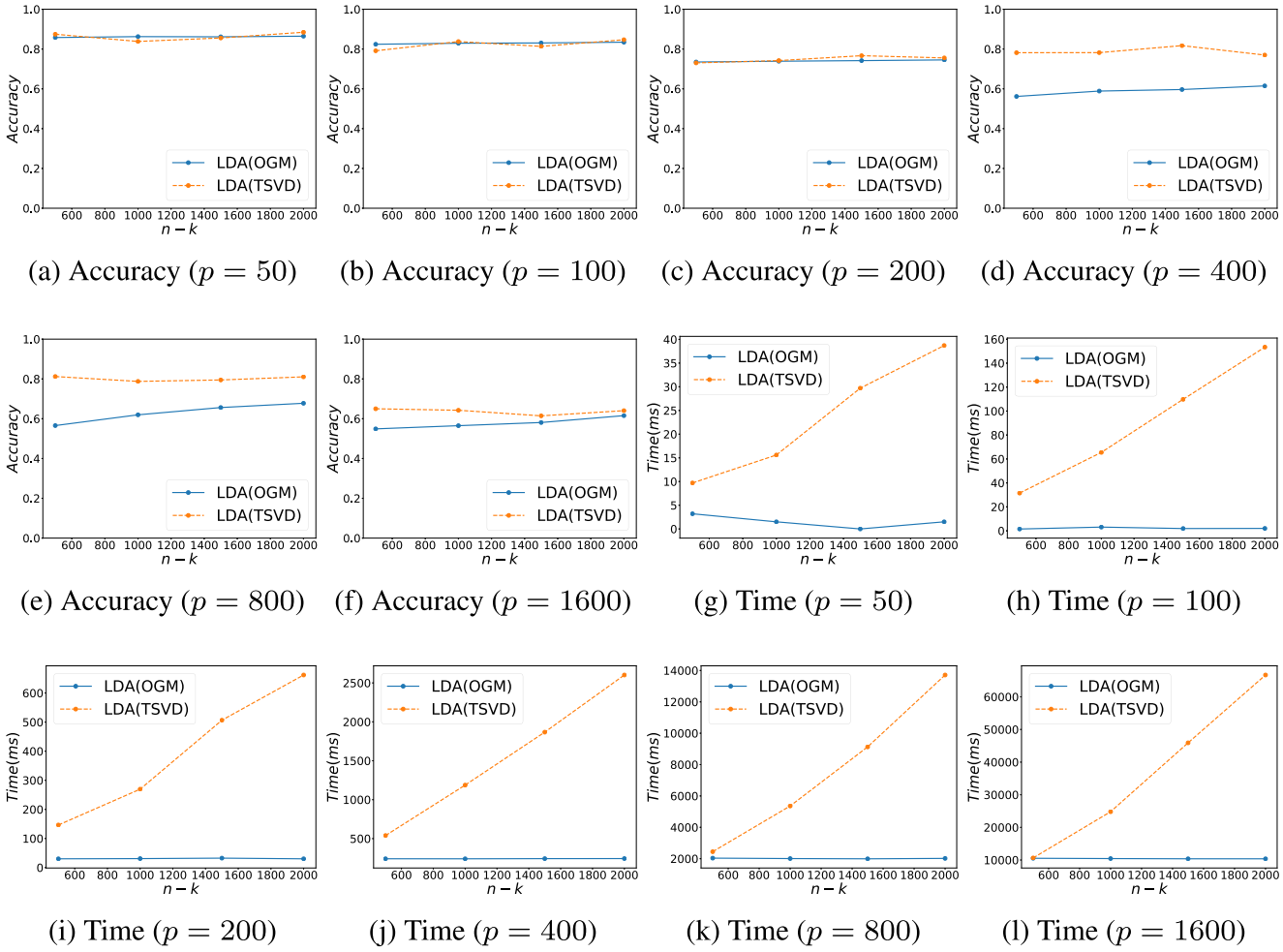
Accuracy Comparison Figures 5a–f present the accuracy comparison between the online LDA based on OGM (entitled as "LDA(OGM)") and the offline LDA based on TSVD (entitled as "LDA(TSVD)"). In terms of overall classification accuracy, it shows LDA(TSVD) performs better than LDA(OGM) with a higher accuracy. However, with more new samples arrival (i.e., increasing $n - k$), such an advantage of accuracy marginally decreases, while the performance of LDA(OGM) significantly improved. We also investigate that LDA(TSVD) doesn't perform consistently, as it only preserves the the top k singular values/vectors while the button $(p - k)$ singular value/vectors are all shrunken. Note that the optimal setting of $k = 20$ is selected through 10 folder cross validation.

Computational Time Figure 5g–l present the comparison of computational time, where the time consumption to update the model per sample is evaluated. Due to the low computational complexity of OGM (i.e., $\mathcal{O}(p^2)$ depending on the number of dimensions p only), the time consumption of LDA(OGM) is constant over the increasing number of arrival samples, while the updating time per sample of LDA(TSVD) scales linearly with the increasing number of arrival samples arrived.

5.3 Online inference

In this experiment, we evaluate the performance of OGM in statistical inference tasks. Specifically, we hope to use OGM to discover significant correlations [26, 44–46] between variables from the online data.

Specifically, the online data in the experiments are all randomly drawn from a p-dimension Gaussian Distributions $\mathcal{N}(\mathbf{0}, \Theta^{-1})$, where $\Theta = \mathbf{L}\mathbf{L}^\top$ and $\mathbf{L}$ is a $p \times p$



(a) Accuracy ($p = 50$) (b) Accuracy ($p = 100$) (c) Accuracy ($p = 200$) (d) Accuracy ($p = 400$)

(e) Accuracy ($p = 800$) (f) Accuracy ($p = 1600$) (g) Time ($p = 50$) (h) Time ($p = 100$)

(i) Time ($p = 200$) (j) Time ($p = 400$) (k) Time ($p = 800$) (l) Time ($p = 1600$)

Fig. 5 Time Consumption of Online Linear Discriminant Analysis

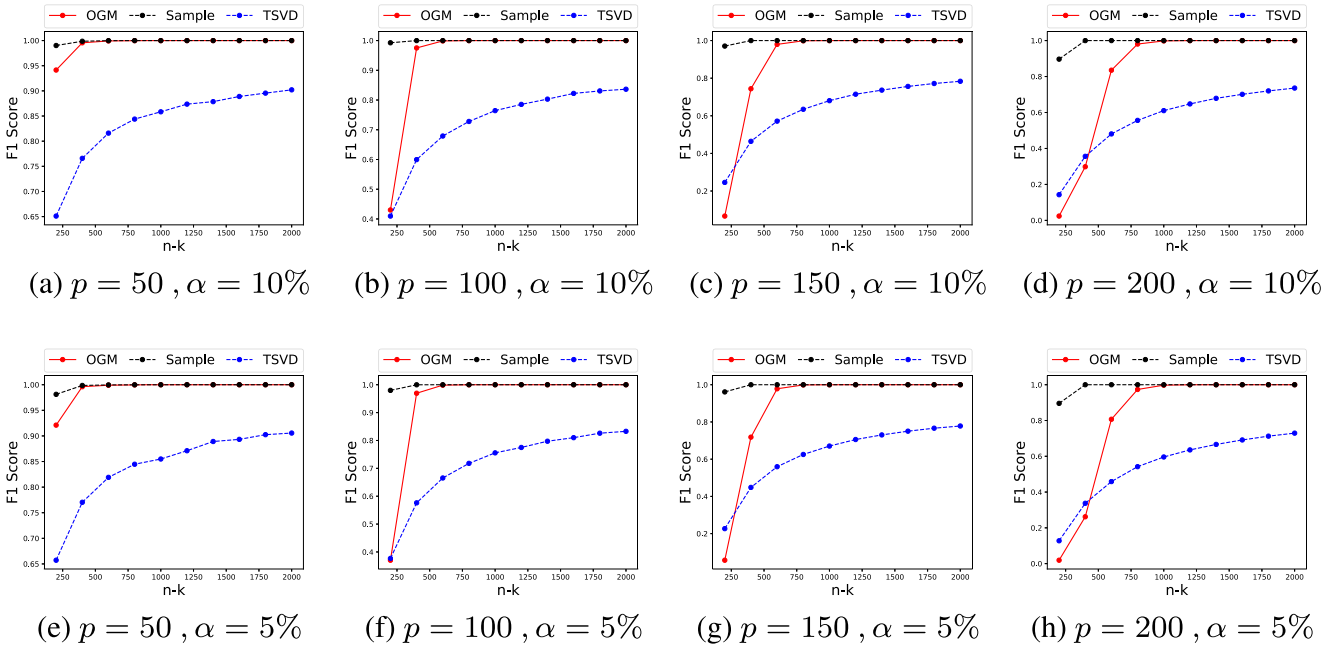
matrix. Each entry $\mathbf{L}_{i,j}$ is drawn from a binary distribution with 50% probability to be 1 and 50% probability to be 0. In this way, $\Theta = \mathbf{L}\mathbf{L}^\top$ becomes a sparse matrix with a number of entries of zero. We consider the zero entry, i.e., $\Theta_{i,j} = 0$ refers to the conditional independence between the random variables located at the i^{th} and the j^{th} dimensions, given other variables. On the contrary, every nonzero entry $\Theta_{i,j} \neq 0$ refers to the potential causalities between the corresponding two variables. Thus, the online inference task here is to use online samples that sequentially arrive and OGM to first estimate the inverse covariance matrix $\widehat{\Theta}_n$, then adopt Algorithm 3 to extract the significant edges [44] using confidence interval (i.e., $1 - \alpha$ where α is defined as the input of Algorithm 3) and further recover the structure of the graph. Finally, we compare the extracted graph with Θ, and calculate the F1-score of the discovered significant edges versus the nonzero entries in Θ.

Let us denote the set of nonzero elements in Θ as $\text{supp}(\Theta) = \{\Theta_{i,j} : i \neq j \text{ and } \Theta_{i,j} \neq 0\}$ and we define $\text{supp}(\mathcal{G}_n) = \{(\mathcal{G}_n)_{i,j} : i \neq j \text{ and } (\mathcal{G}_n)_{i,j} \neq 0\}$ accordingly. Then, the precision, recall and F1-score of dependency discovery is defined as follow.

$$\begin{aligned} \text{Precision}(\Theta, \mathcal{G}_n) &= \frac{|\text{supp}(\mathcal{G}_n) \cap \text{supp}(\Theta)|}{|\text{supp}(\mathcal{G}_n)|} \\ \text{Recall}(\Theta, \mathcal{G}_n) &= \frac{|\text{supp}(\mathcal{G}_n) \cap \text{supp}(\Theta)|}{|\text{suppsupp}(\Theta)|} \\ \text{F1}(\Theta, \mathcal{G}_n) &= \frac{2 \cdot \text{Precision}(\Theta, \mathcal{G}_n) \cdot \text{Recall}(\Theta, \mathcal{G}_n)}{\text{Precision}(\Theta, \mathcal{G}_n) + \text{Recall}(\Theta, \mathcal{G}_n)} \end{aligned} \tag{21}$$

To simplify our research, we estimate the $F1 - score$ to balance the precision and recall for evaluation and comparison.

As a reference, we compare the performance of OGM with offline methods based on sample and TSVD inverse covariance matrix estimator. The offline methods first buffer all sequentially arrival samples, then estimate



(a) $p = 50$, $\alpha = 10\%$ (b) $p = 100$, $\alpha = 10\%$ (c) $p = 150$, $\alpha = 10\%$ (d) $p = 200$, $\alpha = 10\%$

(e) $p = 50$, $\alpha = 5\%$ (f) $p = 100$, $\alpha = 5\%$ (g) $p = 150$, $\alpha = 5\%$ (h) $p = 200$, $\alpha = 5\%$

Fig. 6 F1-Score of Significant Edge Discovery under Various Confidence Interval Settings

the inverse covariance matrix using matrix inverse and TSVD respectively. Both of them use Algorithm 3 to select significant edges. Figure 6 presents the F1-score comparison with the various dimensions $p = 50$, 100, 150 and 200, under 95% (i.e., $\alpha = 5\%$) and 90% (i.e., $\alpha = 10\%$) confidence intervals. As the figure shows, the sample based methods outperform others, due to its oracle properties when a large number of samples arrive. However, the F1-score of significant edges discovered by OGM converges very fast to 1 (significantly faster than TSVD), when more samples arrive. Even when the dimensions are very large (e.g., $p = 200$ or 150), OGM can still converge to the oracle within 1000 arrival samples, while TSVD is with a F1-score less than 90%.

Note that all experiments above are all based on $k = 20$ buffered samples for model initialization.

6 Discussion and conclusion

In this paper, we study the problem of Online Gaussian Graphical Model learning using the incoming data that arrive sequentially. We propose three algorithms, namely OGM, for online graphical model initialization, graphical model online updating, and graphical inference, respectively. Theoretical analysis on the asymptotic properties of OGM, with a Bernstein-style statistical rate of convergence, is given. It shows that OGM can converge to the inverse of true covariance matrix at $\mathcal{O}(\sqrt{\log p/n}+\log p/n+k/n)$ rate with $\mathcal{O}(p^2)$ computational complexity for model updating, where k refers to the number of samples buffered for model initialization, n is the total number of samples arrive, and p refers to the dimension of each sample. Such convergence rate demonstrates the suboptimality of OGM, while three folders of experiments empirically validate the performance and advantage of the proposed methods. Specifically, while offline methods consume (sub)-linear time to re-estimate the model when a new sample arrives, the time consumption for OGM updating only depends on the dimension of the samples.

The limitation of this study is still significant. In the experiment, we compare OGM with the sample-based solution and Truncated SVD (TSVD), while not with other well-known inverse covariance matrix estimators such as GLasso [47], CLIME [48], and/or de-biased GLasso [4, 26]. The reason is that our solution can be viewed as an online approximation to an estimator initialized by TSVD and updated by samples directly. Furthermore, the aforementioned methods require an extensive computation power to solve the optimization problem, while all methods evaluated here are light weighted with low extremely complexity. Indeed, we tried GLasso in our experiments, which leads to better statistical performance but poor time efficiency (especially compared to OGM in online learning settings). Furthermore, OGM offers a simple yet effective inference tool (derived from [26, 39]) to discover the conditional independence/dependence between variables from the online Gaussian graphical models. In our future work, we plan to study novel combinatorial inference tools [2] for OGM with better statistical power.

Acknowledgements This work was supported by the National Key Research and Development Program of China (2018YFE0126000), the National Natural Science Foundation of China (NSFC) (No. 61972050), the Beijing Natural Science Foundation (No. L191012) and the 111 Project (No. B08004). This work was done under the joint efforts between Baidu Research and Mininglamp Academy of Sciences on the topics of federated online advertising.

References

1. Uhler C (2019) Gaussian graphical models: An algebraic and geometric perspective. Chapter in Handbook of Graphical Models
2. Tony Cai T, Ren Z, Zhou HH (2016) Estimating structured high-dimensional covariance and precision matrices: Optimal rates and adaptive estimation. Electron J Stat 10(1):1–59
3. Huntenburg J, Abraham A, Loula J, Liem F, Dadi K, Varoquaux G (2017) Loading and plotting of cortical surface representations in nilearn. Res Ideas Outcomes 3:e12342
4. Xiong H, Cheng W, Bian J, Wenqing Hu, Sun Z, Guo Z (2018) Dbsda: Lowering the bound of misclassification rate for sparse linear discriminant analysis via model debiasing. IEEE Trans Neural Netw Learn Syst 30(3):707–717
5. Bian J, Yang S, Xiong H, Wang L, Fu Y, Sun Z, Guo Z, Wang J (2020) Crledd: Regularized causalities learning for early detection of diseases using electronic health record (ehr) data. IEEE Transactions on Emerging Topics in Computational Intelligence
6. Gaber MM, Zaslavsky A, Krishnaswamy S (2005) Mining data streams: a review. ACM Sigmod Record 34(2):18–26
7. Yang Q, Wu X (2006) 10 challenging problems in data mining research. Int J Inform Technol Dec Making 5(04):597–604
8. Johnstone IM (2001) On the distribution of the largest eigenvalue in principal components analysis. Annals stat. 295–327
9. Lauritzen SL (1996) Graphical models, vol 17. Clarendon Press, Oxford
10. Jordan MI (1998) Learning in graphical models, vol 89. Springer Science & Business Media, Berlin
11. Xiong H, Zhang J, Huang Yu, Leach K, Barnes LE (2017) Daehr: A discriminant analysis framework for electronic health record data and an application to early detection of mental health disorders. ACM Trans Int Syst Technol (TIST) 8(3):47
12. Bian J, Barnes L, Chen G, Xiong H (2017) Early detection of diseases using electronic health records data and covariance-regularized linear discriminant analysis. In: IEEE International conference on biomedical health informatics. IEEE
13. Yang S, Xiong H, Kaibo Xu, Wang L, Bian J, Sun Z (2021) Improving covariance-regularized discriminant analysis for ehr-based predictive analytics of diseases. Appl Intell 51(1):377–395
14. Cheng W, Shi Yu, Zhang X, Wang W (2016) Sparse regression models for unraveling group and individual associations in eqtl mapping. BMC bioinformatics 17(1):136
15. Cheng W, Guo Z, Zhang X, Wang W (2016) Cgc: A flexible and robust approach to integrating co-regularized multi-domain graph for clustering. ACM Trans Know Discov Data (TKDD) 10(4):46
16. Huang JZ, Liu N, Pourahmadi M, Liu L (2006) Covariance matrix selection and estimation via penalised normal likelihood. Biometrika 85–98
17. Friedman J, Hastie T, Tibshirani R (2008) Sparse inverse covariance estimation with the graphical lasso. Biostatistics 9(3):432–441
18. Fan J, Feng Y, Yichao Wu (2009) Network exploration via the adaptive lasso and scad penalties. Annals Appl Stat 3(2):521
19. Ravikumar P, Wainwright MJ, Raskutti G, Yu B (2011) High-dimensional covariance estimation by minimizing l1-penalized log-determinant divergence. Electron J Stat 5:935–980
20. Cai T, Liu W, Xi L (2011) A constrained ℓ_1 minimization approach to sparse precision matrix estimation. J Am Stat Assoc 106(494):594–607
21. Liu Q, Ihler AT (2011) Learning scale free networks by reweighted l1 regularization. In: AISTATS, pp 40–48
22. Liu H, Han F, Zhang C-H (2012) Transelliptical graphical models. In: NIPS, pp 809–817
23. Tony Cai T, Zhou HH (2012) Minimax estimation of large covariance matrices under l1 norm. Stat Sin 22(4):1319–1378
24. Xue L, Ma S, Zou H (2012) Positive-definite ℓ_1-penalized estimation of large covariance matrices. J Am Stat Assoc 107(500):1480–1491
25. Liu H, Wang L, Zhao T (2014) Sparse covariance matrix estimation with eigenvalue constraints. J Comput Graph Stat 23(2):439–459
26. Jankova J, van de Geer S (2015) Confidence intervals for high-dimensional inverse covariance estimation. Electron J Stat 9(1):1205–1229
27. Vapnik VN (2000) The nature of statistical learning theory. Springer, New York
28. Wang H (2012) Bayesian graphical lasso models and efficient posterior computation. Bayesian Anal 7(4):867–886
29. Park T, Casella G (2008) The bayesian lasso. J Am Stat Assoc 103(482):681–686
30. Liu W (2013) Gaussian graphical model estimation with false discovery rate control. Annals Stat 41(6):2948–2978
31. Tan KM, Wang Z, Liu H, Zhang T (2018) Sparse generalized eigenvalue problem: optimal statistical rates via truncated rayleigh flow. J R Stat Soc Series B Stat Methodol 80(5):1057
32. Bian J, Xiong H, Yanjie Fu, Huan J, Guo Z (2020) Mp2sda: Multi-party parallelized sparse discriminant learning. ACM Trans Know Discov Data (TKDD) 14(3):1–22
33. Kummerfeld E, Danks D (2013) Tracking time-varying graphical structure. In: Advances in neural information processing systems (NIPS), pp 1205–1213
34. Kummerfeld E, Danks D (2012) Online learning of time-varying causal structures. In: UAI workshop on causal structure learning
35. Cao X, Khare K, Ghosh M (2016) Posterior graph selection and estimation consistency for high-dimensional bayesian dag models. Ann Stat 47:318–348
36. Xiang R, Khare K, Ghosh M (2015) High dimensional posterior convergence rates for decomposable graphical models. Electron J Stat 9:2828–2854
37. Meng D, Moore KL (2020) Contraction mapping-based robust convergence of iterative learning control with uncertain, locally lipschitz nonlinearity. IEEETrans Syst Man Cybern Syst 50(2):442–454
38. Miller KS (1981) On the inverse of the sum of matrices. Math Mag 54(2):67–72
39. Janková J, van de Geer S (2017) Honest confidence regions and optimality in high-dimensional precision matrix estimation. Test 26(1):143–162
40. Rothman AJ, Bickel PJ, Levina E, Ji Z (2008) Sparse permutation invariant covariance estimation. Electron J Stat 2:494–515
41. Joel A et al (2015) Tropp an introduction to matrix concentration inequalities. Found Trends® in Mach Learn 8(1-2):1–230
42. Gavish M, Donoho DL (2014) The optimal hard threshold for singular values is 4 sqrt 3. IEEE Trans Inf Theory 60(8):5040–5053
43. Cai T, Liu W (2011) A direct estimation approach to sparse linear discriminant analysis. J Am Stat Assoc 106(496):1566–1577

44. Koller D, Friedman N (2009) Probabilistic graphical models: principles and techniques. MIT Press, Massachusetts
45. Mohan K, Pearl J (2014) Graphical models for recovering probabilistic and causal queries from missing data. In: Advances in neural information processing systems, pp 1520–1528
46. Pearl J (2011) The structural theory of causation. In: M̃cKay Illari P, Russo F, Williamson J (eds) Causality in the Sciences, chapter 33. Clarendon Press, Oxford, pp 697–727
47. Witten DM, Tibshirani R (2009) Covariance-regularized regression and classification for high dimensional problems. J R Stat Soc Ser B (Stat Methodol) 71(3):615–636
48. Cai TT, Ma Z, Wu Y (2013) Sparse pca: Optimal rates and adaptive estimation. Annals Stat 41(6):3074–3110

Sijia Yang received MSc in Information, Communications and Technology Business Management from Telecom Ecole de Management, Paris, France 2015 and Bachelor of Engineering Degree from Zhejiang Gongshang University, Zhejiang, China, 2011. She is currently working towards her PhD degree in Cyberspace Security in Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include cyber security, data analytics, and machine learning.

Haoyi Xiong received the Ph.D. degree in computer science from Telecom SudParis jointly with Universite Pierre et Marie Curie, Paris, France, in 2015. From 2016 to 2018, he was a Tenure-Track Assistant Professor with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO, USA (formerly known as University of Missouri at Rolla). From 2015 to 2016, he was a Research Associate with the Department of Systems and Information Engineering, University of Virginia, Charlottesville, VA, USA. He joined Big Data Laboratory, Baidu Research, Beijing, China in 2018 as a Staff R&D Engineer and Research Scientist, where he is currently a Principal R&D Architect and Research Scientist. He also holds an honorary appointment as a Graduate Faculty Scholar affiliated to the ECE PhD Program at University of Central Florida, Orlando FL, United States. His current research interests include automated deep learning (AutoDL), ubiquitous computing, artificial intelligence, and cloud computing. He has published more than 70 papers in top computer science conferences and journals.

Yunchao Zhang obtained his bachelor's degree with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO, USA. In the summers of 2018 and 2019, he joined the Big Data Laboratory, Baidu, Inc., Beijing, China, as a Research Intern. His research interests are in spatiotemporal data mining, reinforcement learning, and operations research.

Yi Ling is currently pursing the PhD degree with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO, USA. He received B.Eng degree in Computer Science from Beijing Forestry University, Beijing, China. His research interests are in data analytics, real-time computing and networking.

Licheng Wang received the B.S. degree in engineering from Northwest Normal University, Lanzhou, China, in 1995, the M.S. degree in mathematics from Nanjing University, Nanjing, China, in 2001, and the Ph.D. degree in engineering from Shanghai Jiao Tong University, Shanghai, China, in 2007. He is currently the Full Professor with Beijing University of Posts and Telecommunications, Beijing, China. His current research interests include cryptography, blockchain, and future Internet architecture.

Dr. Kaibo Xu received his Bachelor degree (1998) in Computer Science from Beijing University of Chemical Technology and his Master (2005) and PhD (2010) in Computer Science from the University of the West of Scotland. He worked as a Teaching Assistant (1998-2004), Lecturer (2004-2009), Associate Professor (2009-2017) at Beijing Union University. He has supervised more than 20 master and doctoral students who are successful in their academic and industrial careers. As the principal investigator, he has received 7 governmental funds and 5 industrial funds with the total amount of 5M in the Chinese dollar. Dr. Kaibo Xu has also consulted extensively and been involved in many industrial projects. He worked as the Chief-Information-Officer (CIO) of Yunbai Clothing Retail Group, China (2016-2019). Currently, he is serving as the vice president and principal scientist of Mininglamp Tech. His research interests include graph mining, knowledge graph and knowledge reasoning.

Dr. Zeyi Sun received the B.Eng. degree in material science and engineering from Tongji University, Shanghai, China, in 2002, the M.Eng. degree in manufacturing from the University of Michigan Ann Arbor, Ann Arbor, MI, USA, in 2010, and the Ph.D. degree in industrial engineering and operations research from the University of Illinois at Chicago, Chicago, IL, USA, in 2015. He served as an Assistant Professor with the Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, Rolla, MO, USA. from 2015 to 2020. Currently, he is a senior research scientist with Mininglamp Academy of Sciences, Mininglamp Technology, Beijing, China. His research interest is mainly focused on using reinforcement learning algorithms to solve dynamic decision-making problem formulated by Markov Decision Process.