UNBALANCED DISTRIBUTED ESTIMATION AND INFERENCE FOR PRECISION MATRICES

Ensiyeh Nezakati, Eugen Pircalabelu

LIDAM Discussion Paper ISBA
2021 / 31

UCLouvain

LIDAM
ISBA | Louvain Institute of Data Analysis and Modeling in economics and statistics

Unbalanced distributed estimation and inference for precision matrices

Ensiyeh Nezakati[†] NEZAKATI.ENSIYEH@UCLOUVAIN.BE

Eugen Pircalabelu[†] EUGEN.PIRCALABELU@UCLOUVAIN.BE

[†]*Institute of Statistics, Biostatistics and Actuarial Sciences*
UCLouvain
Voie du Roman Pays 20, 1348 Louvain-la-Neuve, Belgium

Abstract

This paper studies the estimation of Gaussian graphical models in the unbalanced distributed framework. Unbalanced distributing is an effective approach when the available machines are of different powers or when the existing dataset comes from different resources with different sizes and can not be aggregated in one single computer. In this paper, we propose a new aggregated estimator of the precision matrix and justify such an approach by both theoretical and practical arguments. The limit distribution and consistency of this estimator are investigated. Furthermore, a procedure for performing statistical inference is proposed. On the practical side, a simulation study and real data examples are illustrated. We show that the performance of the distributed estimator is similar to that of the non-distributed estimator using the full data.

Keywords: Gaussian graphical models, Precision matrix, Lasso penalization, Unbalanced distributed setting, De-biased estimator, Confidence distribution.

1. Introduction

Precision matrix estimation plays an important role in statistical and machine learning, especially in the framework of probabilistic graphical modeling. Nowadays, due to the recent advancements in data collection, contemporary applications usually require fast methods and despite recent advances, estimation of the precision matrix remains challenging when the dimension p is large, owing to the fact that the number of parameters to be estimated is of order $\mathcal{O}(p^2)$, unless sparsity is enforced. A large body of literature has studied the estimation problem of the precision matrix in the Gaussian graphical model context in the high-dimensional setting. We refer to Meinshausen and Bühlmann (2006), Yuan and Lin (2007), Banerjee et al. (2008), Friedman et al. (2008), Cai et al. (2011), Wang et al. (2016) and Cai et al. (2016) among many others for a treatment on the subject.

Many datasets nowadays may be too large to fit and be read efficiently into a single ordinary computer. Moreover, sometimes datasets from different resources are private and due to security and privacy concerns one is not allowed to aggregate all the data onto one location. As such, much of the attention has been focused on the setting where the sample size n is large and most approaches propose splitting the observations into K independent sub-samples that are analyzed in parallel. Once the analysis is performed, estimates are combined together into a final estimate that is treated as the final output of the method. A wide range of literature has studied combination methods for parallel estimators in the

context of linear, generalized linear models and kernel estimators. Most of these studies focused on averaging estimators by taking a simple average. The reader can refer to Zhang et al. (2015), Rosenblatt and Nadler (2016), Guo et al. (2017), Lin et al. (2017), Lee et al. (2017) and Battey et al. (2018) among many others. Regarding the estimation of the precision matrix, there is limited literature using parallel analysis. Arroyo and Hou (2016) studied the problem of estimating a precision matrix in a Gaussian graphical model from a set of K distributed samples via a simple average method. They performed an additional local thresholding step on each machine, in order to obtain an estimator that has at most B non-zero entries. This threshold should be selected carefully, in order to ensure that the less important entries are set to zero. Nevertheless, when some of the available machines are more powerful than others, it is not efficient to distribute a dataset on different machines with equally sizes. Instead, one could place larger datasets on powerful machines and smaller datasets on the others. In this case, one deals with unbalanced sub-samples and just taking a simple average is not an optimal approach for aggregating estimators. In this paper, a new approach is introduced to aggregate the estimators using unbalanced sub-samples and it is shown that the performance of this estimator is similar to that of the centralized non-distributed estimator.

In this paper, using a confidence distribution (CD) approach, we propose a new estimator of the precision matrix which is not sensitive to the unbalanced sub-samples and we show the consistency and asymptotic normality of this estimator. The CD concept was first introduced by Fisher (1956) and then formally defined by Efron (1993). Generally, this approach uses a distribution function to estimate the parameter of interest. Xie et al. (2011) developed a general framework for combining CD functions as a means to perform a meta-analysis. Liu et al. (2015) introduced a meta-analysis for heterogeneous studies by combining the confidence density functions derived from the summary statistics of individual studies. Recently, Tang et al. (2020) developed a strategy using a CD method to combine de-biased Lasso-type coefficient estimators in generalized linear models when the sample size is much larger than the number of parameters. They showed that the resulting combined estimator achieves the same estimation efficiency as the centralized maximum likelihood estimator.

In addition to splitting the dataset on observations, splitting the data in terms of features is much more challenging, because subsets of variables are mutually dependent on each other and as such, one needs a new estimation method that accounts for the fact that each machine has access to only a relatively small partition of the entire dataset. Splitting on the features in the case of a block diagonal precision matrix is also investigated in this paper. Given a partition, after splitting the sample into K independent sub-samples, variables are split into M subsets and distributed onto different machines and as a result, both the sample size and number of parameters decrease. The rest of the paper is organized as follows.

In Section 2, we introduce the notation and preliminaries. In Section 3, we present a methodology for performing distributed estimation when each machine receives only a sub-sample of the data. We continue in Section 4 by proposing a final estimator that pools together separate estimators. Theoretical properties like consistency and asymptotic distribution of this estimator are also investigated in this section. As a special case, the distributed procedure on features for a block diagonal matrix is proposed in Section 5. Moreover, using the properties of the new estimator, statistical inference for the general

estimator is developed in Section 6. In Section 7, the performance of the estimator is evaluated at the hand of a controlled simulation study and in Section 8, the performance on a real data set is illustrated. We close off with a discussion on the method and possible extensions in Section 9.

2. Notation and preliminaries

In Gaussian graphical models, it is assumed that a p-dimensional random vector $\dot{\mathbf{X}} = (X^1, \cdots, X^p)^\top$ follows a multivariate normal distribution $\mathcal{N}_p(\mathbf{0}, \mathbf{\Sigma})$ with mean vector $\mathbf{0}$ and covariance matrix $\mathbf{\Sigma}$. The components of $\dot{\mathbf{X}}$ correspond to the vertex set $\mathcal{V} = \{1, \cdots, p\}$ of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the edge set $\mathcal{E}$ describes the conditional dependence relationship between components $X^1, \cdots, X^p$. A pair (a, b) is included in the edge set $\mathcal{E}$ if and only if the variables X^a and X^b are not independent given all remaining variables. Under $\dot{\mathbf{X}} \sim \mathcal{N}_p(\mathbf{0}, \mathbf{\Sigma})$, a pair of variables is conditionally independent given all remaining variables if and only if the corresponding entry in the precision matrix $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ is zero. Elements of the precision matrix may thus be interpreted as the edge weights. Denote the maximum number of non-zero entries of $\mathbf{\Theta}$ per row with d and the set of non-zero entries of $\mathbf{\Theta}$ with S, having cardinality s. The set of non-active components is denoted by S^c. Before starting the discussion, we introduce some order notation which is needed later in the paper. For two sequences $\{a_n; n \geq 1\}$ and $\{b_n; n \geq 1\}$, $b_n = \mathcal{O}(a_n)$ if there are positive numbers M_0 and N_0 such that $|\, \frac{b_n}{a_n} \,| \leq M_0$ for all $n \geq N_0$. Similarly, for a random sequence $\{X_n; n \geq 1\}$, we write $X_n = \mathcal{O}_p(a_n)$ if for every $\epsilon > 0$, there exist finite numbers $M_0 > 0$ and $N_0 > 0$ such that $P(|\, \frac{X_n}{a_n} \,| > M_0) < \epsilon$ for all $n \geq N_0$. Analogously, we write $b_n = \Omega(a_n)$ if there are positive numbers M_0 and N_0 such that $|\, \frac{b_n}{a_n} \,| \geq M_0$ for all $n \geq N_0$. We write $b_n \asymp a_n$ if both $b_n = \mathcal{O}(a_n)$ and $b_n = \Omega(a_n)$ hold. Moreover, $b_n = o(a_n)$ if $\lim_{n\to\infty} \frac{b_n}{a_n} = 0$. In the case of a random sequence $\{X_n; n \geq 1\}$, we write $X_n = o_p(a_n)$ if $\frac{X_n}{a_n} \xrightarrow{p} 0$, as $n \to \infty$, where the notation $\xrightarrow{p}$ denotes convergence in probability.

In the sequel, we introduce notation at the sample level. Consider an i.i.d sample $\dot{\mathbf{X}}_1, \cdots, \dot{\mathbf{X}}_n$ from $\dot{\mathbf{X}}$, where for each $i = 1, \cdots, n$ we have $\dot{\mathbf{X}}_i = (X_i^1, \cdots, X_i^p)^\top$. Consider this sample in a matrix form as $\mathbf{X} = (\dot{\mathbf{X}}_1, \dot{\mathbf{X}}_2, \cdots, \dot{\mathbf{X}}_n)^\top$ which is of dimension $n \times p$. Based on the sample $\mathbf{X}$, our goal is to estimate the structure of the graph $\mathcal{G}$, or equivalently, find the zero pattern of $\mathbf{\Theta}$. The usual assumption in this context is that the underlying graph is sparse, which means that a certain sparsity is imposed on the maximum node degree (d) of the precision matrix. According to the relation between graph edges and entries of the precision matrix in Gaussian graphical models, this sparsity reflects that the related graph has a rather low number of edges. A naive estimator for $\mathbf{\Theta}$ obtained by inverting the sample covariance $\hat{\mathbf{\Sigma}} = \frac{1}{n}\mathbf{X}^\top\mathbf{X}$ is practically never sparse for any real data. Furthermore, if $n < p$ the sample covariance matrix is rank-deficient and thus not even invertible. One common approach to overcome these problems is to impose an additional ℓ_1 penalty on the elements of $\mathbf{\Theta}$ when estimating it. This kind of regularization effectively forces some of the elements of $\mathbf{\Theta}$ to zero, thus resulting in sparse solutions. There exists a wide variety of methods making use of ℓ_1 regularization, see for example Friedman et al. (2008), Hsieh et al. (2014), Cai et al. (2016), etc.

3. Distributed penalized estimation

In this section, we introduce new distributed estimators of the precision matrix $\mathbf{\Theta}$ and investigate their properties. To this end, we consider two general regularity assumptions (see for example, Ravikumar et al., 2011; Jankova and Van De Geer, 2015) on the true precision matrix $\mathbf{\Theta}$. Before listing the assumptions, we keep track of two metrices on the covariance matrix $\mathbf{\Sigma}$ and the precision matrix $\mathbf{\Theta}$. The first metric is the ℓ_∞ norm of $\mathbf{\Sigma}$ which is denoted by $\kappa_{\mathbf{\Sigma}} := |||\mathbf{\Sigma}|||_\infty$, where $|||\mathbf{A}|||_\infty = \max_a \sum_{b=1}^p \mid \mathbf{A}_{ab} \mid$ is the maximum of the absolute row sums for an arbitrary matrix $\mathbf{A}$. To propose the second metric, consider the Hessian matrix of the negative log-likelihood function $L(\mathbf{\Theta}) = \text{trace}(\hat{\mathbf{\Sigma}}\mathbf{\Theta}) - \log\det(\mathbf{\Theta})$ as $\mathbf{\Gamma}$. It can be easily shown that $\mathbf{\Gamma} = \mathbf{\Sigma} \otimes \mathbf{\Sigma}$, where $\otimes$ is the Kronecker product. The second metric that we keep track of is $\kappa_{\mathbf{\Gamma}} = |||\big(\mathbf{\Gamma}_{SS}\big)^{-1}|||_\infty$ which is the ℓ_∞ norm of the inverse of the matrix $\mathbf{\Gamma}_{SS} = \big[\mathbf{\Sigma} \otimes \mathbf{\Sigma}\big]_{S\times S} \in \mathbb{R}^{s\times s}$, where the subscript $S \times S$ is used to denote a sub-matrix of $\big[\mathbf{\Sigma} \otimes \mathbf{\Sigma}\big]$ whose rows and columns are indexed by the elements of S. The two following assumptions are needed for the theoretical results:

(A1) The irrepresentability condition holds for the true precision matrix $\mathbf{\Theta}$, i.e., there exists $\alpha \in (0, 1]$ such that $\max_{e\in S^c} ||\mathbf{\Gamma}_{eS}\big(\mathbf{\Gamma}_{SS}\big)^{-1}||_1 \leq 1 - \alpha$, where $\|\mathbf{x}\|_1$ is the ℓ_1 norm of the vector defined as $\|\mathbf{x}\|_1 = \sum_a \mid \mathbf{x}_a \mid$ for an arbitrary vector $\mathbf{x}$.

This assumption is necessary and sufficient for the graphical Lasso estimator to exhibit model selection consistency. This means that the non-edge (non-active) features can not be too correlated with the edge-based (active) features. This is so since if the non-edge features were highly correlated with the edge-based ones, it would become much harder to recover important features.

(A2) The eigenvalues of the precision matrix $\mathbf{\Theta}$ are bounded, i.e., there exists $\Lambda \asymp 1$ such that $\frac{1}{\Lambda} \leq \Lambda_{\min}(\mathbf{\Theta}) \leq \Lambda_{\max}(\mathbf{\Theta}) \leq \Lambda$, where $\Lambda_{\min}(\mathbf{\Theta})$ and $\Lambda_{\max}(\mathbf{\Theta})$ are the minimum and maximum eigenvalues of $\mathbf{\Theta}$, respectively.

Considering the two above assumptions, suppose now at the sample level that n samples are divided randomly into K sub-samples with size n_k for the k-th sub-sample ($k = 1, \cdots, K$). Denote the k-th sub-sample in matrix form as $\mathbf{X}_k = (\dot{\mathbf{X}}_{1,k}, \dot{\mathbf{X}}_{2,k}, \cdots, \dot{\mathbf{X}}_{n_k,k})^\top$ which is of dimension $n_k \times p$. By splitting the sample into K sub-samples, one can analyze each sub-sample per machine. For each sub-matrix $\mathbf{X}_k (k = 1, \cdots, K)$, the graphical Lasso estimator $\mathbf{L}_k$, defined by Friedman et al. (2008), is the solution to the optimization problem

$$\mathbf{L}_k = \arg\min_{\mathbf{\Theta}\in S^p_{++}} \Big\{\text{trace}\big(\hat{\mathbf{\Sigma}}_k\mathbf{\Theta}\big) - \log\det(\mathbf{\Theta}) + \lambda_k\|\mathbf{\Theta}\|_{1,\text{off}}\Big\}, \tag{1}$$

where $\hat{\mathbf{\Sigma}}_k = \frac{1}{n_k}(\mathbf{X}_k)^\top\mathbf{X}_k$ is the sample covariance in the k-th sub-sample, $\|\cdot\|_{1,\text{off}}$ is the ℓ_1 off-diagonal penalty of the matrix defined as $\|\mathbf{\Theta}\|_{1,\text{off}} = \sum_{a\neq b} \mid \mathbf{\Theta}_{ab} \mid$, the quantity $\mathbf{\Theta}_{ab}$ is the (a, b)-th element of $\mathbf{\Theta}$ and S^p_{++} is the space of positive definite matrices of dimension $p \times p$. The graphical Lasso estimator (1) is biased due to the ℓ_1 penalty which is added to the loss function. It is known (see Ravikumar et al., 2011) that this estimator satisfies in the Karush-Kuhn-Tucker (KKT) condition as $\hat{\mathbf{\Sigma}}_k - \hat{\mathbf{\Theta}}^{-1} + \lambda_k\hat{\mathbf{D}}_k = \mathbf{0}$, where the matrix $\hat{\mathbf{D}}_k$ belongs to the sub-differential of the off-diagonal norm $\|\cdot\|_{1,\text{off}}$ evaluated at $\hat{\mathbf{L}}_k$. Jankova

and Van De Geer (2015) proposed the de-biased graphical Lasso estimator in order to correct the bias, which for our setting can be constructed using the k-th sub-sample as

$$\hat{\mathbf{L}}_k = 2\mathbf{L}_k - \mathbf{L}_k\hat{\mathbf{\Sigma}}_k\mathbf{L}_k. \tag{2}$$

This de-biased estimator has the appealing property that each entry of the matrix is asymptotically normally distributed and one can easily construct confidence intervals for the entries of $\mathbf{\Theta}$.

Remark 1 Under the irrepresentability assumption (A1) with a constant $\alpha \in (0, 1]$ and assumption (A2), suppose $\mathbf{L}_k$ is the solution to the optimization problem (1) with tuning parameter $\lambda_k \asymp \sqrt{\log p/n_k}$. Using Theorem 1 of Jankova and Van De Geer (2015), under the sparsity condition

$$d = o\big(\frac{\sqrt{n_k}}{C\log p}\big),$$

with

$$C = \frac{1}{\alpha}\kappa_\Gamma \max\big\{\frac{1}{\alpha}\kappa_\Gamma\kappa_\Sigma, \kappa_\Sigma(\log p)^{-1/2}, \kappa_\Sigma^3\kappa_\Gamma(\log p)^{-1/2}\big\}, \tag{3}$$

$\hat{\mathbf{L}}_{ab,k}$ is asymptotically normally distributed for each $(a, b) \in \mathcal{V} \times \mathcal{V}$ as

$$\sqrt{n_k}(\hat{\mathbf{L}}_{ab,k} - \mathbf{\Theta}_{ab})/\sigma_{ab} = \mathbf{Z}_{ab,k} + o_p(1), \quad n_k \to \infty, \tag{4}$$

where $\mathbf{Z}_{ab,k}$ converges weakly to $\mathcal{N}(0, 1)$, $\hat{\mathbf{L}}_{ab,k}$ is the (a, b)-th element of $\hat{\mathbf{L}}_k$ and $\sigma_{ab}^2 = \mathbf{\Theta}_{aa}\mathbf{\Theta}_{bb} + \mathbf{\Theta}_{ab}^2$. Moreover, using their Lemma 2, under the sparsity assumption

$$d \le \frac{\sqrt{n_k}}{\log p(1 + 8/\alpha)\max\{\kappa_\Sigma\kappa_\Gamma, \kappa_\Sigma^3\kappa_\Gamma^2\}},$$

the estimator $\hat{\sigma}_{ab,k}^2 = \mathbf{L}_{aa,k}\mathbf{L}_{bb,k} + \mathbf{L}_{ab,k}^2$ is consistent for σ_{ab}^2 and one may substitute σ_{ab} by $\hat{\sigma}_{ab,k}$ in the application. Furthermore, under the above assumptions, it follows that the convergence rate of the de-biased estimator $\hat{\mathbf{L}}_k$ is

$$\| \hat{\mathbf{L}}_k - \mathbf{\Theta} \|_{\max} = \mathcal{O}_p\bigg(\max\big\{\sqrt{\log p/n_k}, 1/\alpha^2\kappa_{\mathbf{\Gamma}}^2\kappa_{\mathbf{\Sigma}}d\log p/n_k\big\}\bigg),$$

where $\| \cdot \|_{\max}$ is the $\ell_{\max}$ norm defined as $\| \cdot \|_{\max} = \max_{ab} | \mathbf{A}_{ab} |$ for an arbitrary matrix $\mathbf{A}$.

The asymptotic normal distribution of $\hat{\mathbf{L}}_k (k = 1, \cdots, K)$ leads us to construct a new aggregated estimator using K estimators from the sub-samples. This estimator is introduced in the next section.

4. Combined estimator across the sub-samples

The estimator in (2) is defined at the level of the k-th sub-sample and as such one creates a sequence of estimators $\hat{\mathbf{L}}_k (k = 1, \cdots, K)$. One can envision multiple ways to combine estimators $\hat{\mathbf{L}}_k$ to obtain a pooled, final estimator. One simple method consists in averaging

over the sub-samples. Lee et al. (2017) devised a communication-efficient approach to average de-biased Lasso estimators for a distributed sparse regression in the high-dimensional setting. They showed that their approach converges at the same rate as Lasso as long as the dataset was not split across too many machines. We introduce next based on Schweder and Hjort (2002) and Singh et al. (2005) the "confidence distribution" concept which will play a central role in our approach when combining different estimators. Suppose $Y_1, \cdots, Y_n$ are n independent random samples from a population model $F_\theta(\cdot)$ and $\mathcal{X}$ is the sample space corresponding to the data $\mathbf{Y}_n = (Y_1, \cdots, Y_n)^\top$. Let θ be the parameter of interest associated with $F_\theta(\cdot)$, and let Θ be the parameter space of θ.

Definition 2 A function $H_n(\cdot) \equiv H_n(\mathbf{Y}, \cdot)$ on $\mathcal{X} \times \Theta \to [0, 1]$ is called a CD for the parameter θ, if it satisfies the following two requirements: (R1) for every given $\mathbf{Y}_n \in \mathcal{X}$, $H_n(\cdot)$ is a continuous cumulative distribution function for θ; (R2) at the true parameter $\theta = \theta_0$, $H_n(\theta_0) \equiv H_n(\mathbf{Y}_n, \theta_0)$ as a function of the sample $\mathbf{Y}_n$ follows the uniform distribution $U[0, 1]$. When the derivative exists, the CD density $h_n(\theta) = \frac{d}{d\theta} H_n(\theta)$ is also known as a confidence density.

Based on Definition 2, the CD function should have two properties. The first property is that, for a given sample, the CD should be a distribution function on the parameter space. The second property requires that the CD function contains the "right" information about the true parameter value θ_0 for making correct inferences.

Now consider the asymptotic normal distribution in (4) which directly satisfies the two requirements of the CD definition. Let $\hat{f}_{ab,k}(\cdot)$ be the data-driven asymptotic normal density of $\hat{\mathbf{L}}_{ab,k}$ which can be obtained from (4) by substituting $\hat{\sigma}_{ab,k}$ for σ_{ab}. One can obtain the combined estimator by maximizing the pseudo-log-likelihood function obtained from the asymptotic densities $\hat{f}_{ab,k}(\cdot)$ as follows

$$\hat{\mathbf{\Delta}}_{ab} = \arg\max_{\mathbf{\Theta}_{ab}} \log \prod_{k=1}^{K} \hat{f}_{ab,k}(\mathbf{\Theta}_{ab}). \tag{5}$$

Whenever the above confidence distribution-based estimator is derived, the general theory of maximum likelihood estimation provides standard statistical tests and other useful results for statistical inference. By taking the derivative of the log-likelihood function (5) with respect to $\mathbf{\Theta}_{ab}$ and setting it to 0, we obtain the final estimator as:

$$\hat{\mathbf{\Delta}}_{ab} = \frac{1}{\sum_{k=1}^{K} \frac{n_k}{\hat{\sigma}^2_{ab,k}}} \times \sum_{k=1}^{K} \frac{n_k}{\hat{\sigma}^2_{ab,k}} \hat{\mathbf{L}}_{ab,k}. \tag{6}$$

Note that this new estimator behaves like a weighted average as such, its weights are a function of the sample size n_k and of the variance $\hat{\sigma}^2_{ab,k}$ in each sub-sample. In contrast, if one were to act by taking the simple average, one might miss information that is derived from the variances. The advantage of this estimator, as stated in Theorem 3, is that it is asymptotically unbiased and normally distributed. We can thus construct confidence intervals and statistical tests based on this estimator.

Theorem 3 *Under assumptions* (A1) *and* (A2)*, suppose* $\mathbf{L}_k(k = 1, \cdots, K)$ *is the solution to optimization problem* (1) *with tuning parameter* $\lambda_k \asymp \sqrt{\log p/n_k}$ *and* $\hat{\mathbf{\Delta}}_{ab}$ *is the pooled estimator from* (6)*. Let* $n_k/n \to c_k \in (0, 1]$ *as* $n_k \to \infty$ *such that* $\sum_{k=1}^{K} c_k = 1$*. Under the sparsity condition*

$$d = o\big(\frac{\sqrt{n_\dagger}}{C \log p}\big), \tag{7}$$

where C *is defined in* (3)*, it holds that*

$$\sqrt{\sum_{k=1}^{K} \frac{n_k}{\hat{\sigma}^2_{ab,k}}}\Big(\hat{\mathbf{\Delta}}_{ab} - \mathbf{\Theta}_{ab}\Big) = \mathbf{W}_{ab,\dagger} + o_p(1),$$

where $n_\dagger = \min_{k\in\{1,\cdots,K\}} n_k$ *and* $\mathbf{W}_{ab,\dagger}$ *converges weakly to* $\mathcal{N}(0, 1)$ *as* $n_\dagger \to \infty$*.*

Proof Since $\frac{\sqrt{n_\dagger}}{C \log p} \le \frac{\sqrt{n_k}}{C \log p}$ for each $k = 1, \cdots, K$, the sparsity assumption (7) implies that

$$d = o\big(\frac{\sqrt{n_k}}{C \log p}\big), \quad k = 1, \cdots, K.$$

Furthermore, $n_\dagger \to \infty$ implies $n_k \to \infty$ for each k. So, using (4) we get

$$\frac{\sqrt{n_k}}{\sigma_{ab}}(\hat{\mathbf{L}}_{ab,k} - \mathbf{\Theta}_{ab}) = \mathbf{Z}_{ab,k} + \mathbf{H}_{ab,k}, \quad k = 1, \cdots, K,$$

where $\mathbf{Z}_{ab,k}$ converges weakly to $\mathcal{N}(0, 1)$ and $\mathbf{H}_{ab,k} = o_p(1)$ as $n_k \to \infty$. Multiplying above expression by $\frac{\sqrt{n_k}}{\sigma_{ab}}$, we have

$$\frac{n_k}{\sigma^2_{ab}}(\hat{\mathbf{L}}_{ab,k} - \mathbf{\Theta}_{ab}) = \frac{\sqrt{n_k}}{\sigma_{ab}}\mathbf{Z}_{ab,k} + \frac{\sqrt{n_k}}{\sigma_{ab}}\mathbf{H}_{ab,k}. \tag{8}$$

Using the continuous mapping $g(x) = 1/x$ for a consistent estimator $\hat{\sigma}^2_{ab,k}$, we get $\sigma^2_{ab}/\hat{\sigma}^2_{ab,k} \xrightarrow{p} 1$, as $n_k \to \infty$. By multiplying (8) with $\sigma^2_{ab}/\hat{\sigma}^2_{ab,k}$, we have

$$\frac{n_k}{\hat{\sigma}^2_{ab,k}}(\hat{\mathbf{L}}_{ab,k} - \mathbf{\Theta}_{ab}) = \frac{\sigma^2_{ab}}{\hat{\sigma}^2_{ab,k}} \times \frac{\sqrt{n_k}}{\sigma_{ab}}\mathbf{Z}_{ab,k} + \frac{\sigma^2_{ab}}{\hat{\sigma}^2_{ab,k}} \times \frac{\sqrt{n_k}}{\sigma_{ab}}\mathbf{H}_{ab,k}. \tag{9}$$

Since the K estimators $\hat{\mathbf{L}}_{ab,1}, \cdots, \hat{\mathbf{L}}_{ab,K}$ are independent, by summing up (9) from $k = 1$ to K and multiplying by $1/\sqrt{\sum_{k=1}^{K} \frac{n_k}{\sigma^2_{ab}}}$, we get

$$\frac{1}{\sqrt{\sum_{k=1}^{K} \frac{n_k}{\sigma^2_{ab}}}} \sum_{k=1}^{K} \frac{n_k}{\hat{\sigma}^2_{ab,k}}(\hat{\mathbf{L}}_{ab,k} - \mathbf{\Theta}_{ab}) = \mathbf{W}^{'}_{ab,\dagger} + \mathbf{H}^{'}_{ab,\dagger}, \tag{10}$$

where

$$\mathbf{W}^{'}_{ab,\dagger} = \frac{1}{\sqrt{\sum_{k=1}^{K} \frac{n_k}{\sigma^2_{ab}}}} \sum_{k=1}^{K} \frac{\sigma^2_{ab}}{\hat{\sigma}^2_{ab,k}} \times \frac{\sqrt{n_k}}{\sigma_{ab}}\mathbf{Z}_{ab,k}, \quad \mathbf{H}^{'}_{ab,\dagger} = \frac{1}{\sqrt{\sum_{k=1}^{K} \frac{n_k}{\sigma^2_{ab}}}} \sum_{k=1}^{K} \frac{\sigma^2_{ab}}{\hat{\sigma}^2_{ab,k}} \times \frac{\sqrt{n_k}}{\sigma_{ab}}\mathbf{H}_{ab,k}.$$

Denoting $\mathbf{V}_{ab,k} = \frac{\sigma^2_{ab}}{\hat{\sigma}^2_{ab,k}}\mathbf{Z}_{ab,k}$ and using Slutsky's theorem, it holds that $\mathbf{V}_{ab,k} \xrightarrow[n_k\to\infty]{d} \mathcal{N}(0,1)$, where $\xrightarrow{d}$ means convergence in distribution. As such, $\mathbf{W}'_{ab,\dagger} = \frac{1}{\sqrt{\sum_{k=1}^K \frac{n_k}{\sigma^2_{ab}}}} \sum_{k=1}^K \frac{\sqrt{n_k}}{\sigma_{ab}}\mathbf{V}_{ab,k}$ also converges in distribution to $\mathcal{N}(0,1)$ as $n_\dagger \to \infty$. On the other hand,

$$\frac{\sum_{k=1}^K \frac{n_k}{\hat{\sigma}^2_{ab,k}}}{\sum_{k=1}^K \frac{n_k}{\sigma^2_{ab}}} = \sum_{k=1}^K \left\{ \frac{\frac{n_k}{\hat{\sigma}^2_{ab,k}}}{\sum_{k=1}^K \frac{n_k}{\sigma^2_{ab}}} \right\} = \sum_{k=1}^K \left\{ \frac{n_k}{n} \times \frac{\sigma^2_{ab}}{\hat{\sigma}^2_{ab,k}} \right\}. \tag{11}$$

Since $\frac{n_k}{n} \xrightarrow[n_k\to\infty]{} c_k$, then $\frac{n_k}{n} \times \frac{\sigma^2_{ab}}{\hat{\sigma}^2_{ab,k}} \xrightarrow[n_k\to\infty]{p} c_k$ and (11) converges in probability as

$$\frac{\sum_{k=1}^K \frac{n_k}{\hat{\sigma}^2_{ab,k}}}{\sum_{k=1}^K \frac{n_k}{\sigma^2_{ab}}} \xrightarrow[n_\dagger\to\infty]{p} \sum_{k=1}^K c_k = 1.$$

Considering the continuous mapping $g(x) = 1/\sqrt{x}$, we get $\sqrt{\frac{\sum_{k=1}^K \frac{n_k}{\sigma^2_{ab}}}{\sum_{k=1}^K \frac{n_k}{\hat{\sigma}^2_{ab,k}}}} \xrightarrow[n_\dagger\to\infty]{p} 1$. By multiplying (10) in this expression, we have

$$\begin{aligned}\sqrt{\sum_{k=1}^K \frac{n_k}{\hat{\sigma}^2_{ab,k}}}\left(\hat{\mathbf{\Delta}}_{ab} - \mathbf{\Theta}_{ab}\right) &= \frac{1}{\sqrt{\sum_{k=1}^K \frac{n_k}{\hat{\sigma}^2_{ab,k}}}} \sum_{k=1}^K \frac{n_k}{\hat{\sigma}^2_{ab,k}}(\hat{\mathbf{L}}_{ab,k} - \mathbf{\Theta}_{ab}) \\ &= \sqrt{\frac{\sum_{k=1}^K \frac{n_k}{\sigma^2_{ab}}}{\sum_{k=1}^K \frac{n_k}{\hat{\sigma}^2_{ab,k}}}}\mathbf{W}'_{ab,\dagger} + \sqrt{\frac{\sum_{k=1}^K \frac{n_k}{\sigma^2_{ab}}}{\sum_{k=1}^K \frac{n_k}{\hat{\sigma}^2_{ab,k}}}}\mathbf{H}'_{ab,\dagger}.\end{aligned}$$

Using Slutsky's theorem again it holds that $\mathbf{W}_{ab,\dagger} = \sqrt{\frac{\sum_{k=1}^K \frac{n_k}{\sigma^2_{ab}}}{\sum_{k=1}^K \frac{n_k}{\hat{\sigma}^2_{ab,k}}}}\mathbf{W}'_{ab,\dagger} \xrightarrow[n_\dagger\to\infty]{d} \mathcal{N}(0,1)$.

Hence, to prove the theorem, it is enough to show that $\sqrt{\frac{\sum_{k=1}^K \frac{n_k}{\sigma^2_{ab}}}{\sum_{k=1}^K \frac{n_k}{\hat{\sigma}^2_{ab,k}}}}\mathbf{H}'_{ab,\dagger} = o_p(1)$. Since $\sqrt{\frac{\sum_{k=1}^K \frac{n_k}{\sigma^2_{ab}}}{\sum_{k=1}^K \frac{n_k}{\hat{\sigma}^2_{ab,k}}}} \xrightarrow[n_\dagger\to\infty]{p} 1$, it is just needed to prove $\mathbf{H}'_{ab,\dagger} = o_p(1)$.

As $\frac{\sigma^2_{ab}}{\hat{\sigma}^2_{ab,k}} \xrightarrow[n_k\to\infty]{p} 1$ and $\mathbf{H}_{ab,k} = o_p(1)$, then $\mathbf{J}_{ab,k} = \frac{\sigma^2_{ab}}{\hat{\sigma}^2_{ab,k}}\mathbf{H}_{ab,k} = o_p(1)$. Furthermore,

$$\mathbf{H}'_{ab,\dagger} = \frac{1}{\sqrt{\sum_{k=1}^K \frac{n_k}{\sigma^2_{ab}}}} \sum_{k=1}^K \frac{\sqrt{n_k}}{\sigma_{ab}}\mathbf{J}_{ab,k} \leq \max_k \left\{\mathbf{J}_{ab,k}\right\} \sum_{k=1}^K \left\{ \frac{\frac{\sqrt{n_k}}{\sigma_{ab}}}{\sqrt{\sum_{k=1}^K \frac{n_k}{\sigma^2_{ab}}}} \right\}.$$

Since $0 < \frac{n_k}{\sigma^2_{ab}} \leq \sum_{k=1}^K \frac{n_k}{\sigma^2_{ab}}$, we have that $\frac{\frac{\sqrt{n_k}}{\sigma_{ab}}}{\sqrt{\sum_{k=1}^K \frac{n_k}{\sigma^2_{ab}}}} \leq 1$ which implies that

$$\mathbf{H}'_{ab,\dagger} \leq K \max_k \left\{\mathbf{J}_{ab,k}\right\} = o_p(1),$$

and the proof is completed. ■

In the following remark, the convergence rate of the de-biased distributed estimator $\hat{\mathbf{\Delta}}_{ab}$ is investigated.

Remark 4 To measure the distance between the estimator $\hat{\mathbf{\Delta}}$ and the true precision matrix $\mathbf{\Theta}$, consider the difference between $\hat{\mathbf{\Delta}}_{ab}$ and $\mathbf{\Theta}_{ab}$. We have

$$\begin{aligned}
\mid \hat{\mathbf{\Delta}}_{ab} - \mathbf{\Theta}_{ab} \mid &= \mid \frac{1}{\sum_{k=1}^{K} \frac{n_k}{\hat{\sigma}^2_{ab,k}}} \times \sum_{k=1}^{K} \frac{n_k}{\hat{\sigma}^2_{ab,k}} (\hat{\mathbf{L}}_{ab,k} - \mathbf{\Theta}_{ab}) \mid \\
&\leq \frac{1}{\sum_{k=1}^{K} \frac{n_k}{\hat{\sigma}^2_{ab,k}}} \times \sum_{k=1}^{K} \frac{n_k}{\hat{\sigma}^2_{ab,k}} \mid \hat{\mathbf{L}}_{ab,k} - \mathbf{\Theta}_{ab} \mid \\
&\leq \frac{1}{\sum_{k=1}^{K} \frac{n_k}{\hat{\sigma}^2_{ab,k}}} \times \sum_{k=1}^{K} \frac{n_k}{\hat{\sigma}^2_{ab,k}} \max_k \mid \hat{\mathbf{L}}_{ab,k} - \mathbf{\Theta}_{ab} \mid \\
&= \max_{k \in \{1,\cdots,K\}} \mid \hat{\mathbf{L}}_{ab,k} - \mathbf{\Theta}_{ab} \mid \\
&\leq \max_{k \in \{1,\cdots,K\}} \|\hat{\mathbf{L}}_k - \mathbf{\Theta}\|_{\max}.
\end{aligned}$$

Under the assumptions of Remark 1, we get

$$\mid \hat{\mathbf{\Delta}}_{ab} - \mathbf{\Theta}_{ab} \mid = \mathcal{O}_p\Bigg(\max_{k \in \{1,\cdots,K\}} \Bigg\{ \max\Big\{ \sqrt{\log p / n_k}, 1/\alpha^2 \kappa^2_{\mathbf{\Gamma}} \kappa_{\mathbf{\Sigma}} d \log p / n_k \Big\} \Bigg\} \Bigg). \tag{12}$$

If $\kappa_{\mathbf{\Gamma}} = \mathcal{O}(1), \kappa_{\mathbf{\Sigma}} = \mathcal{O}(1)$ and $1/\alpha = \mathcal{O}(1)$ then the convergence rate in (12) simplifies to

$$\mid \hat{\mathbf{\Delta}}_{ab} - \mathbf{\Theta}_{ab} \mid = \mathcal{O}_p\Bigg(\max_{k \in \{1,\cdots,K\}} \Bigg\{ \max\Big\{ \sqrt{\log p / n_k}, d \log p / n_k \Big\} \Bigg\} \Bigg),$$

and under the sparsity condition $d = o(\frac{\sqrt{n_\dagger}}{\log p})$, it is deduced that $\mid \hat{\mathbf{\Delta}}_{ab} - \mathbf{\Theta}_{ab} \mid = \mathcal{O}_p\big(\sqrt{\log p / n_\dagger}\big)$.

Algorithm 1 contains a pseudo-code to implement the proposed procedure for estimating $\mathbf{\Theta}$. The first stage consists in splitting the dataset, the second performs estimation via the sub-samples and the third stage corresponds to combining estimators from the sub-samples.

In the next section we also consider splitting on the feature space in the case of a block diagonal covariance matrix.

Algorithm 1: Distributed estimator of the precision matrix $\mathbf{\Theta}$

1) Split the sample data into K sub-samples. Denote the k-th sub-sample with $\mathbf{X}_k, k = 1, \cdots, K$.
2) **Estimation in the k-th sub-sample, $k = 1, \cdots, K$:**
   2-1) Using $\mathbf{X}_k$ and optimization problem (1), obtain $\mathbf{L}_k$ as an estimator of the precision matrix $\mathbf{\Theta}$;
   2-2) Compute the de-biased estimator $\hat{\mathbf{L}}_k = 2\mathbf{L}_k - \mathbf{L}_k\hat{\mathbf{\Sigma}}_k\mathbf{L}_k$, where $\hat{\mathbf{\Sigma}}_k = \frac{1}{n_k}(\mathbf{X}_k)^\top\mathbf{X}_k$;
   2-3) Obtain $\hat{\sigma}^2_{ab,k} = \mathbf{L}_{aa,k}\mathbf{L}_{bb,k} + (\mathbf{L}_{ab,k})^2$ for $(a, b) \in \mathcal{V} \times \mathcal{V}$;
3) **Combine estimators from the sub-samples:** Using the estimators in steps 2-2) and 2-3), construct the final estimator $\hat{\mathbf{\Delta}}_{ab} = \frac{1}{\sum_{k=1}^K \frac{n_k}{\hat{\sigma}^2_{ab,k}}} \times \sum_{k=1}^K \frac{n_k}{\hat{\sigma}^2_{ab,k}}\hat{\mathbf{L}}_{ab,k}$.

5. Block diagonal covariance matrix

Suppose that the true covariance matrix $\mathbf{\Sigma}$ has a block diagonal form with M blocks. Then, the true precision matrix of interest has also a block diagonal form as

$$\mathbf{\Theta} = \begin{pmatrix} \mathbf{\Theta}^1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\Theta}^M \end{pmatrix}. \tag{13}$$

Consider the vertex set and the size of block $\mathbf{\Theta}^m (m = 1, \cdots, M)$ as $\mathcal{V}_m$ and d_m, respectively. Define the maximum row degree of $\mathbf{\Theta}$ as $d := \max_{m=1,\cdots,M} d_m$. As such, the maximum row degree of $\mathbf{\Theta}$ and $\mathbf{\Sigma}$ are both equal as d. In this case, $\kappa_{\mathbf{\Sigma}}$ and $\kappa_{\mathbf{\Gamma}}$ are tractable and we can find upper bounds for them. For instance, using the definition of $\kappa_{\mathbf{\Sigma}}$, the relation between ℓ_∞ and spectral norm, and the fact that $d_m \leq d$, it holds that

$$\kappa_{\mathbf{\Sigma}} = \max_{m=1,\cdots,M} \parallel (\mathbf{\Theta}^m)^{-1} \parallel_\infty \leq \sqrt{d} \max_{m=1,\cdots,M} \parallel (\mathbf{\Theta}^m)^{-1} \parallel_2,$$

where $\parallel \cdot \parallel_2$ is the the spectral norm defined as $\parallel \mathbf{A} \parallel_2 = \sqrt{\Lambda_{\max}(\mathbf{A}^\top\mathbf{A})}$ for an arbitrary real valued matrix $\mathbf{A}$. Based on the fact that the eigenvalues of $\mathbf{\Sigma}$ are the inverse of the eigenvalues of $\mathbf{\Theta}$ and using assumption (A2), it is deduced that $\kappa_{\mathbf{\Sigma}} = \mathcal{O}(\sqrt{d})$. The same argument can be made for $\kappa_{\mathbf{\Gamma}}$ and it holds that $\kappa_{\mathbf{\Gamma}} = \mathcal{O}(d)$.

When the precision matrix has the block diagonal form, at the sample level, one can split features into M subsets, place the subsets on different machines, and so reduce the computational complexity. The complexity of the graphical Lasso algorithm, which uses a row by row block coordinate descent method, is $\mathcal{O}(p^3)$, while by splitting the variables into M subsets it decreases to $\mathcal{O}(Mp_*^3)$, where $p_* \ll p$ is the number of parameters in the largest subset among M subsets. Suppose that the features in the k-th sub-sample are further

divided into M subsets as the splitting pattern at the population level. Denote the sample data in the k-th sub-sample and the m-th subset by $\mathbf{X}_k^m = (\dot{\mathbf{X}}_{1,k}^m, \dot{\mathbf{X}}_{2,k}^m, \cdots, \dot{\mathbf{X}}_{n_k,k}^m)^\top$ which is a matrix of dimension $n_k \times p_m$ with the i-th row denoted by the vector $(X_{i,k}^{1,m}, \cdots, X_{i,k}^{p_m,m})^\top$ and the j-th column denoted by the vector $(X_{1,k}^{j,m}, \cdots, X_{n_k,k}^{j,m})^\top$, where p_m is the number of parameters in the m-th ($m = 1, \cdots, M$) subset. For each sub-matrix $\mathbf{X}_k^m$, the graphical Lasso estimator of $\mathbf{\Theta}^m$, say $\mathbf{L}_k^m$, is defined as the solution to the optimization problem

$$\mathbf{L}_k^m = \arg \min_{\mathbf{\Theta}^m \in S_{++}^{p_m}} \left\{ \operatorname{trace}\big(\hat{\mathbf{\Sigma}}_k^m \mathbf{\Theta}^m\big) - \log\det(\mathbf{\Theta}^m) + \lambda_k \|\mathbf{\Theta}^m\|_{1,\text{off}} \right\}, \tag{14}$$

where $\hat{\mathbf{\Sigma}}_k^m = \frac{1}{n_k}(\mathbf{X}_k^m)^\top \mathbf{X}_k^m$ is the sample covariance in the k-th sub-sample and the m-th subset. After solving the optimization problem (14) for all subsets $m = 1, \cdots, M$, one can construct $\bar{\mathbf{L}}_k$ as

$$\bar{\mathbf{L}}_k = \begin{pmatrix} \mathbf{L}_k^1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{L}_k^M \end{pmatrix},$$

which is the graphical Lasso estimator of $\mathbf{\Theta}$ in (13). Under the condition that the tuning parameter λ_k is the same for all subsets $m = 1, \cdots, M$, $\bar{\mathbf{L}}_k$ satisfies the KKT condition and one can use the sample covariance $\hat{\mathbf{\Sigma}}_k$ to obtain the de-biased estimator $\hat{\mathbf{L}}_k = 2\bar{\mathbf{L}}_k - \bar{\mathbf{L}}_k \hat{\mathbf{\Sigma}}_k \bar{\mathbf{L}}_k$. It is noteworthy that if the tuning parameter λ_k is not considered the same in all M subsets, the estimators $\mathbf{L}_k^1, \cdots, \mathbf{L}_k^M$ are obtained with different tuning parameters and the KKT condition does not hold for the aggregated estimator $\bar{\mathbf{L}}_k$ and as a result, the de-biasing is not applicable.

Due to the boundedness property of $\kappa_{\mathbf{\Sigma}}$ and $\kappa_{\mathbf{\Gamma}}$ in the block diagonal structure, i.e., $\kappa_{\mathbf{\Sigma}} = \mathcal{O}(\sqrt{d})$ and $\kappa_{\mathbf{\Gamma}} = \mathcal{O}(d)$, by assuming $1/\alpha = \mathcal{O}(1)$, the convergence rate (12) is simplified as

$$| \hat{\mathbf{\Delta}}_{ab} - \mathbf{\Theta}_{ab} | = \mathcal{O}_p\bigg(\max_{k \in \{1, \cdots, K\}} \Big\{ \max\big\{ \sqrt{\log p / n_k}, d^{7/2} \log p / n_k \big\} \Big\} \bigg).$$

Considering the sparsity condition $d^{7/2} = o(\frac{\sqrt{n_\dagger}}{\log p})$, it is again deduced that $| \hat{\mathbf{\Delta}}_{ab} - \mathbf{\Theta}_{ab} | = \mathcal{O}_p(\sqrt{\log p / n_\dagger})$.

6. Asymptotic statistical inference of the precision matrix

Based on the asymptotic distribution of $\hat{\mathbf{\Delta}}_{ab}$, one can construct inferential procedures. In this section we provide a confidence interval and hypothesis testing framework for the elements of the precision matrix using the asymptotic results of Theorem 3. According to it, the $(1-\alpha)100\%$ asymptotic confidence interval for $\mathbf{\Theta}_{ab}$ based on the de-biased distributed estimator $\hat{\mathbf{\Delta}}_{ab}$ is constructed as

$$\text{CI}_{ab} = \hat{\mathbf{\Delta}}_{ab} \pm \Phi^{-1}(1-\alpha/2) / \sqrt{\sum_{k=1}^{K} n_k / \hat{\sigma}_{ab,k}^2}, \tag{15}$$

where $\Phi^{-1}(1-\alpha/2)$ is the $(1-\alpha/2)$-th quantile of the standard normal distribution. Moreover, using the asymptotic distribution of $\hat{\mathbf{\Delta}}_{ab}$, the rejection region at the level of $\frac{\alpha}{p(p-1)/2}$ (with the Bonferroni correction) for the hypothesis test $H_0 : \mathbf{\Theta}_{ab} = 0$ (which indicates that there is no edge between nodes X^a and X^b) vs $H_1 : \mathbf{\Theta}_{ab} \neq 0$ can be constructed as follows

$$| \hat{\mathbf{\Delta}}_{ab} | > \Phi^{-1}(1 - \frac{\alpha}{p(p-1)}) / \sqrt{\sum_{k=1}^{K} n_k / \hat{\sigma}^2_{ab,k}}. \tag{16}$$

The coverage probability and the length of the confidence intervals are investigated via a simulation study in the next section. Moreover, testing the zero edges between variables is performed in Section 8 using a real data example.

7. Simulation study

In this section, we demonstrate the performance of the proposed distributed estimator with a simulation study. To this end, we compare the performance of the distributed estimator with the following estimators:

1) (Full) A de-biased estimator based on the full non-distributed data given by the de-biased graphical Lasso $\hat{\mathbf{\Theta}}_{\text{Full}}$ (the estimator of Jankova and Van De Geer, 2015).

2) (Naive) An estimator based on splitting data on the rows and averaging the de-biased graphical Lasso estimators from each machine $\hat{\mathbf{\Theta}}_{\text{Naive}} = \frac{1}{K}\sum_{k=1}^{K} \hat{\mathbf{L}}_k$, where $\hat{\mathbf{L}}_k$ is defined in (2).

To conduct the simulation, we set the total sample size $n = 50000$ as fixed and changed $K = 5, 10, 20$. To show the performance of the distributed estimator in the unbalanced setting, we considered the following structure during the simulation study.

Suppose among all available machines, two of them are powerful. The first one is the most powerful and 55% of dataset is distributed on this computer. The second one is less powerful than the first, and 60% of remaining dataset is distributed on this one. The other remaining dataset is distributed roughly equally on the remaining machines.

For the data generating process, we fixed the number of variables to $p = 500, 1000, 2000$ and the samples are generated from a multivariate normal distribution $\mathcal{N}_p(\mathbf{0}, \mathbf{\Sigma})$ such that $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ has the following structures:

- block diagonal with 10 blocks and 0.2 as the probability of connection in each block;
- random with probability of connection 0.05;
- hub with 10 hubs;
- chain or tridiagonal with 1 on the diagonal and 0.2 on the upper and lower diagonals.

The corresponding graph structures are shown in Figure 1 for $p = 500$. In this study, the tuning parameter λ_k in the graphical Lasso algorithm has been chosen as $\lambda_k = \lambda = \sqrt{\log p/n}$ for all simulation settings like in Jankova and Van De Geer (2015). Alternatively, it can be chosen by K-fold cross validation in each sub-sample. Moreover, all simulation results are

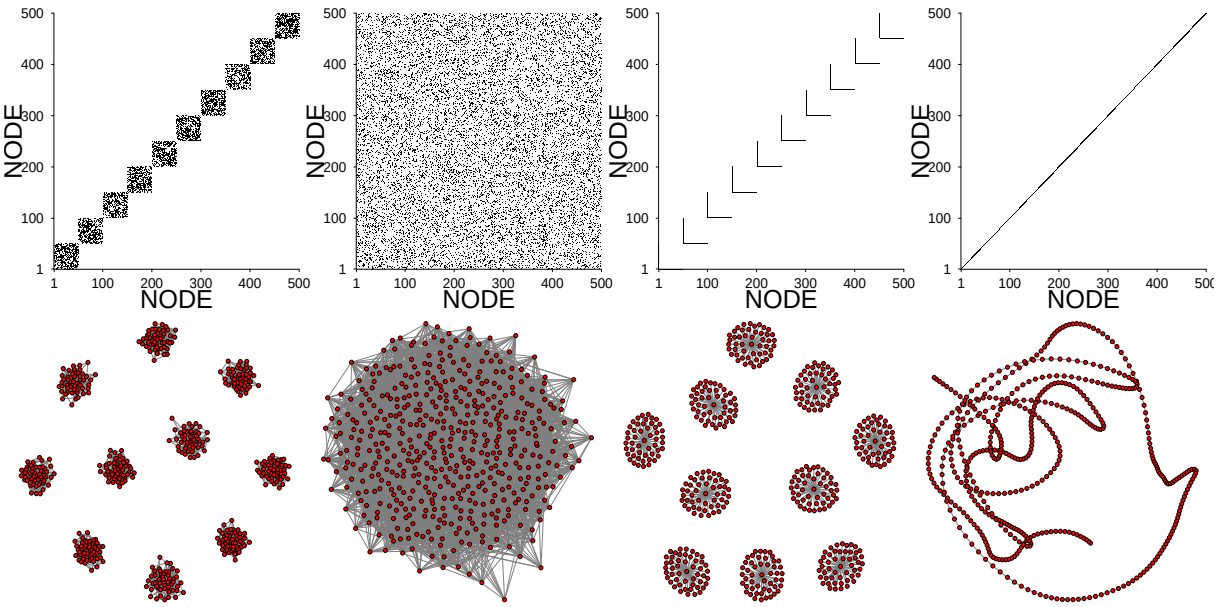

Figure 1: Visual representation of four $\mathbf{\Theta}$ matrices and their graph structures for $p = 500$ nodes. From left to right: block diagonal with 10 blocks, random, hub with 10 hubs and tridiagonal.

calculated as averages over 100 different repetitions. To compare the performance of the estimators, we used the Frobenius norm, $\ell_{\max}$ and ℓ_∞ norms between the estimated precision matrix for each competitor and the true precision matrix. Since the results of $p = 500$ and $p = 1000$ were rather similar, we report here the case of $p = 1000$ for illustration. In the block diagonal structure, the dataset was split on both the observations and features, so the computations are much faster than the other structures. Therefore, for the block diagonal structure, p is also considered as $p = 2000$.

The results are presented in Tables 1 and 2 for the Frobenius norm via the simulation setting explained above and it is observed that the performance of the proposed estimator is as good as that of the non-distributed full estimator in terms of Frobenius norm. The results for the two other norms are presented in Appendix A. The proposed norm is computed for the active set S, which corresponds to the edges of graph, and the non-active set S^c, separately. By increasing K from 5 to 20, the norm does not increase much and the differences are negligible. This suggests that by splitting the dataset on the sample size, one might not lose much information. On the other hand, for the naive estimator, by increasing K, the norm increases substantially, which suggests that it is quite sensitive to K, as opposed to the distributed estimator we propose. This behavior holds on both the active and the non-active sets for all four structures of $\mathbf{\Theta}$. Especially, in the random graph on the active set there are large differences between the naive estimator and our procedure. Comparing Tables 1 and 2, it is observed that by increasing p the performance of the naive estimator deteriorates much more, while the proposed distributed method works well and its performance is close to the centralized estimator that uses the full data.

Table 1: Frobenius norm of the proposed estimators when $n = 50000$ and $p = 1000$.

		Active set			Non-active set		
		K			K		
		5	10	20	5	10	20
Block	Full	2.20	2.20	2.20	6.09	6.09	6.09
	Distributed	2.23	2.25	2.29	6.08	6.09	6.13
	Naive	3.29	4.46	6.13	9.04	11.61	13.38
Random	Full	6.63	6.63	6.63	0.04	0.04	0.04
	Distributed	6.68	6.90	7.14	0.04	0.04	0.22
	Naive	10.30	17.15	32.93	0.06	0.10	0.90
Hub	Full	2.10	2.10	2.10	5.91	5.91	5.91
	Distributed	2.73	2.77	2.51	5.76	5.95	6.13
	Naive	4.36	6.94	12.77	8.72	14.68	24.99
Tridiag	Full	0.20	0.20	0.20	4.39	4.39	4.39
	Distributed	0.29	0.53	0.52	4.48	4.67	4.77
	Naive	0.83	4.13	13.55	7.35	13.78	25.82

Table 2: Frobenius norm of the proposed estimators when $n = 50000$ and $p = 2000$ and the block diagonal structure is used to simulate data.

	Active set			Non-active set		
	K			K		
	5	10	20	5	10	20
Full	4.22	4.22	4.22	11.26	11.26	11.26
Distributed	4.33	4.41	4.50	11.24	11.27	11.40
Naive	6.38	9.08	13.84	16.72	21.59	25.25

In addition to error norms, the coverage and the length of the confidence intervals are also obtained for the estimator using the full data and our proposed estimator at significance level $\alpha = 0.05$. To this end, we estimate the coverage probability of the confidence interval (15) by computing its empirical version. Similar to Jankova and Van De Geer (2015), the empirical probability that the true parameter $\mathbf{\Theta}_{ab}$ in included in the confidence interval is defined as $\hat{P}_{ab} = \frac{\#\{\mathbf{\Theta}_{ab} \in \text{CI}_{ab,r}\}}{R}$, where R is the number of simulation repetitions, $\text{CI}_{ab,r}$ is the confidence interval defined in (15) for the r-th repetition, and # denotes the number of times that $\mathbf{\Theta}_{ab}$ is included in the confidence interval. After obtaining $\hat{P}_{ab}$ for all $(a, b) \in \mathcal{V} \times \mathcal{V}$, the average coverage probability on the active set S can be obtained as $\text{Avg.Cov}_S = \frac{1}{s}\sum_{(a,b)\in S} \hat{P}_{ab}$, where s is the number of active components. Similar computations can be implemented for obtaining the coverage probability over the non-active set S^c. The results are reported in Tables 3 and 4, when $p = 1000$. It is observed that our estimator performs similarly to the non-distributed estimator on both the active and the non-active sets, as the coverage probabilities are close to the nominal level 95%. Moreover, the average lengths are relatively narrow and they don't change much by increasing K. There is one challenging case: the tridiagonal structure. In this case, increasing K seems to reduce the average coverage probability on the active set. Such a phenomenon can probably be explained by the small signal to noise ratio in this example. Table 4 present the results for $p = 2000$ and it shows the good performance of distributed method when both n and p are large.

Table 3: Average coverage probability and average length of the confidence interval when $n = 50000$ and $p = 1000$.

		Avg.Cov (Distributed)			Avg.Len (Distributed)			Avg.Cov (Full)	Avg.Len (Full)
		K			K				
		5	10	20	5	10	20		
Active set	Block	.94	.94	.94	.03	.03	.03	.95	.03
	Random	.88	.88	.89	.03	.03	.03	.89	.03
	Hub	.94	.93	.94	.02	.02	.03	.95	.02
	Tridiag	.83	.38	.43	.02	.02	.02	.95	.02
Non-active set	Block	.96	.97	.97	.03	.03	.02	.96	.03
	Random	.95	.94	.94	.03	.03	.03	.96	.03
	Hub	.95	.95	.95	.02	.02	.03	.95	.02
	Tridiag	.95	.95	.95	.02	.02	.02	.95	.02

Table 4: Average coverage probability and average length of the confidence interval when $n = 50000$, $p = 2000$ and the block structure is used to simulate data.

	Avg.Cov (Distributed)			Avg.Len (Distributed)			Avg.Cov (Full)	Avg.Len (Full)
	K			K				
	5	10	20	5	10	20		
Active set	.94	.94	.94	.03	.03	.03	.94	.03
Non-active set	.97	.97	.97	.03	.03	.03	.97	.03

8. Real data example

To explore the performance of our proposed estimator on real data, we used the publicly available "4 university web-pages" dataset which was collected in January 1997 by the World Wide Knowledge Base project of the CMU text learning group. The original dataset had been pre-processed by standard text mining methods and it is available in Cardoso-Cachopo (2007). This dataset contains web-pages collected from computer science departments of four universities Cornell, Texas, Washington and Wisconsin in seven categories: student, faculty, staff, department, course, project and other. The 'other' category is a collection of web-pages that were not considered in the main 6 categories. In this study, we considered the four largest categories student, faculty, course and project containing 1641, 1124, 930 and 504 web-pages, respectively. To calculate the term-document matrix $\mathbf{X}$, we used the log-entropy weighting method of Dumais (1991) which was also implemented in Guo et al. (2011). The final dataset contains $n = 4199$ web-pages as the observations and 7686 distinct terms as features. We considered for the analysis $p = 500$ terms with the highest log-entropy weights.

We have compared next the proposed distributed estimator with $K = 3, 4, 5$ and the de-biased estimator that uses all of the data. For both procedures $\lambda = \sqrt{\log p/n}$, $\alpha = 5\%$ and a Bonferroni correction is applied for multiple testing. The rejection region of the distributed estimator is obtained using (16) and for the non-distributed estimator (that uses the full data) of Jankova and Van De Geer (2015) it is defined as $| \hat{\mathbf{T}}_{ab} | > \Phi^{-1}(1 - \frac{\alpha}{p(p-1)})\hat{\sigma}_{ab}/\sqrt{n}$. Table 5 below present the obtained results.

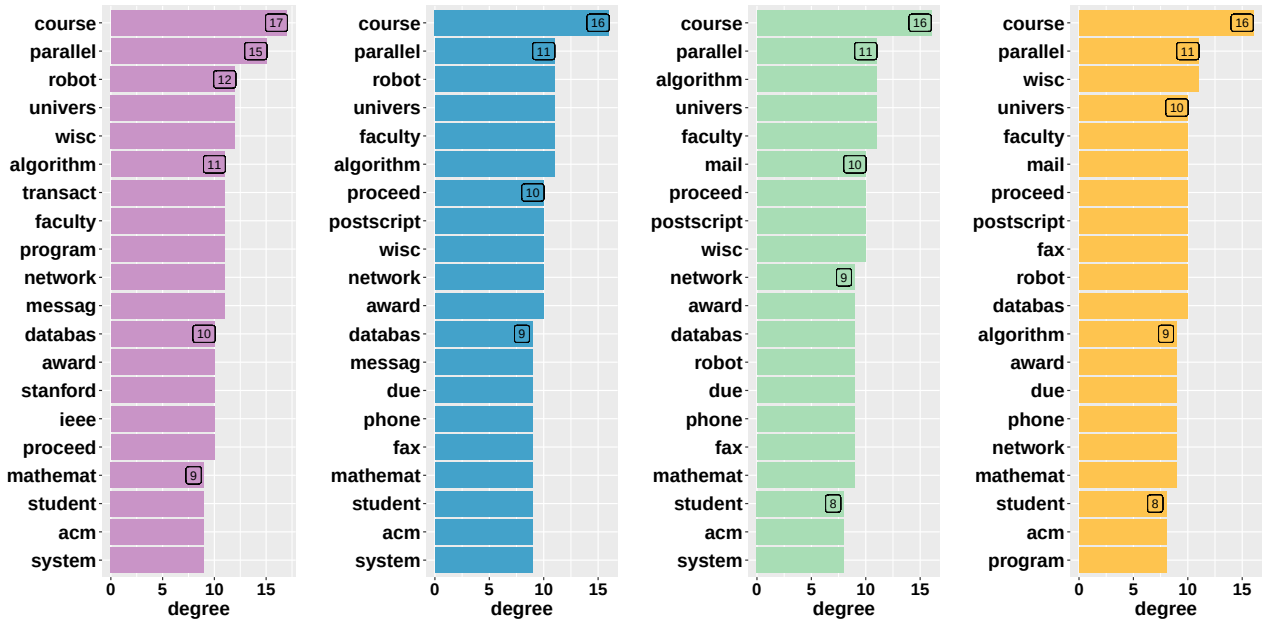

Figure 2: The first 20 terms with the highest node degrees. From left to right: non-distributed centralized estimator, distributed with $K = 3, 4$ and 5 machines.

Table 5: Meaningful edges based on multiple testing and common edges among distributed and full estimators.

	Full	$K = 3$	$K = 4$	$K = 5$
Meaningful edges	1044	842	832	831
Common edges		801	768	754

It is observed that there are many common edges between the graphs estimated by the distributed estimator and the full one. By increasing K, the number of common edges decreases due to the loss of information by splitting on more machines. The first twenty terms with the highest node degrees based on the four considered estimators are presented in Figure 2. It is observed that the term 'course' is identified as the term with the highest node degree, with degree 17 and 16 by the full and distributed method, respectively. Moreover, some terms like 'parallel', 'robot', 'univers', etc. are high degree nodes among the four estimators. The sparse graphical Lasso estimator is also considered as competitor and by this method we identified 17877 edges suggesting that many estimated edges might in fact be false positive edges.

Afterwards, to investigate the performance of the estimators for completely independent variables, we permuted sample data for each feature to construct a dataset with mutually independent variables. As such, all the off-diagonal elements of the precision matrix should be estimated as zero. By implementing separately the de-biased estimator using the full data and the distributed estimator with $K = 3, 4, 5$ machines on this independently permuted dataset, we identified zero meaningful edges which confirms this assertation. However, by implementing the sparse graphical lasso estimator on this independent dataset we identified

1643 edges which confirms the large amount of false positive edges that are identified by this procedure.

9. Conclusion

In this paper, we proposed a new distributed estimator of the precision matrix in the Gaussian graphical models framework. The estimator is constructed to tackle an unbalanced split of the sample data as input. To improve misaligned selection of non-zero estimated entries on different sub-samples, a bias correction was performed in each sub-sample. Finally, all estimators were pooled together into a composite estimator using a confidence distribution. Then, statistical guarantees of the distributed estimator were provided. Its tractable limit distribution was derived and it was shown that under regularity assumptions it follows an asymptotic normal distribution. Moreover, the convergence rate of this estimator was investigated and it was shown that under mild conditions it is of order $\sqrt{\log p/n_\dagger}$. In addition to unbalanced splitting of observations, splitting on the features was also performed in the case of a block diagonal covariance matrix. The finite sample performance of the proposed estimator was evaluated with a simulation study where it has been observed that the estimator produced competitive results relative to a non-distributed estimator that uses the entire data. Moreover, it performs substantially better than the naive estimator obtained by taking a simple average over the estimators from the sub-samples. It was also observed that the coverage probability of the distributed estimator is close to that of the non-distributed estimator. This points to the fact that in practice, performing distributed estimation across multiple machines in our unbalanced framework induces a minimal loss in performance relative to models using all the data in a centralized location. For future research, we plan to study an extension of this approach in the setting of splitting on the features in a general covariance matrix framework.

References

J. Arroyo and E. Hou. Efficient distributed estimation of inverse covariance matrices. In *2016 IEEE Statistical Signal Processing Workshop (SSP)*, pages 1–5. IEEE, 2016.

O. Banerjee, L. E. Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *The Journal of Machine Learning Research*, 9(3):485–516, 2008.

H. Battey, J. Fan, H. Liu, J. Lu, and Z. Zhu. Distributed testing and estimation under sparse high dimensional models. *The Annals of Statistics*, 46(3):1352, 2018.

T. Cai, W. Liu, and X. Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.

T. Cai, W. Liu, and H. Zhou. Estimating sparse precision matrix: Optimal rates of convergence and adaptive estimation. *The Annals of Statistics*, 44(2):455–488, 2016.

A. Cardoso-Cachopo. Improving methods for single-label text categorization. PhD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, 2007.

S. T. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments, & Computers*, 23(2):229–236, 1991.

B. Efron. Bayes and likelihood calculations from confidence intervals. *Biometrika*, 80(1): 3–26, 1993.

R. A. Fisher. *Statistical methods and scientific inference*. Hafner Publishing Co., 1956.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

J. Guo, E. Levina, G. Michailidis, and J. Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15, 2011.

Z. C. Guo, L. Shi, and Q. Wu. Learning theory of distributed regression with bias corrected regularization kernel network. *The Journal of Machine Learning Research*, 18(1):4237–4261, 2017.

C. J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar. Quic: quadratic approximation for sparse inverse covariance estimation. *The Journal of Machine Learning Research*, 15 (1):2911–2947, 2014.

J. Jankova and S. Van De Geer. Confidence intervals for high-dimensional inverse covariance estimation. *Electronic Journal of Statistics*, 9(1):1205–1229, 2015.

J. D. Lee, Q. Liu, Y. Sun, and J. E. Taylor. Communication-efficient sparse regression. *The Journal of Machine Learning Research*, 18(1):115–144, 2017.

S. Lin, X. Guo, and D. Zhou. Distributed learning with least square regularization. *The Journal of Machine Learning Research*, 18(92):1–31, 2017.

D. Liu, R. Y. Liu, and M. Xie. Multivariate meta-analysis of heterogeneous studies using only summary statistics: efficiency and robustness. *Journal of the American Statistical Association*, 110(509):326–340, 2015.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.

P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

J. D. Rosenblatt and B. Nadler. On the optimality of averaging in distributed statistical learning. *Information and Inference: A Journal of the IMA*, 5(4):379–404, 2016.

T. Schweder and N. L. Hjort. Confidence and likelihood. *Scandinavian Journal of Statistics*, 29(2):309–332, 2002.

K. Singh, M. Xie, and W. E. Strawderman. Combining information from independent sources through confidence distributions. *The Annals of Statistics*, 33(1):159–183, 2005.

L. Tang, L. Zhou, and P. X. K. Song. Distributed simultaneous inference in generalized linear models via confidence distribution. *Journal of Multivariate Analysis*, 176:104567, 2020.

L. Wang, X. Ren, and Q. Gu. Precision matrix estimation in high dimensional Gaussian graphical models with faster rates. In *Artificial Intelligence and Statistics*, pages 177–185, 2016.

M. Xie, K. Singh, and W. E. Strawderman. Confidence distributions and a unifying framework for meta-analysis. *Journal of the American Statistical Association*, 106(493):320–333, 2011.

M. Yuan and Y. Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

Y. Zhang, J. Duchi, and M. Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16(1):3299–3340, 2015.

Appendix A.

Some simulation results are presented in this section.

Table 6: The $\ell_{\max}$ norm of the proposed estimators when $n = 50000$ and $p = 1000$.

		Active set			Non-active set		
		K			K		
		5	10	20	5	10	20
Block	Full	.05	.05	.05	.04	.04	.04
	Distributed	.05	.05	.05	.04	.04	.04
	Naive	.06	.08	.11	.06	.07	.08
Random	Full	.04	.04	.04	.02	.02	.02
	Distributed	.04	.04	.04	.02	.02	.02
	Naive	.06	.10	.23	.02	.04	.10
Hub	Full	.09	.09	.09	.20	.20	.20
	Distributed	.11	.11	.10	.15	.14	.14
	Naive	.16	.21	.20	.15	.15	.15
Tridiag	Full	.02	.02	.02	.02	.02	.02
	Distributed	.02	.03	.03	.02	.02	.02
	Naive	.04	.14	.41	.04	.07	.13

Table 7: The $\ell_{\max}$ norm of the proposed estimators when $n = 50000$, $p = 2000$ and the block structure is used to simulate data.

	Active set			Non-active set		
	K			K		
	5	10	20	5	10	20
Full	.05	.05	.05	.03	.03	.03
Distributed	.05	.05	.05	.03	.03	.04
Naive	.06	.09	.12	.05	.07	.08

Table 8: The ℓ_∞ norm of the proposed estimators when $n = 50000$ and $p = 1000$.

		Active set			Non-active set		
		K			K		
		5	10	20	5	10	20
Block	Full	1.15	1.15	1.15	7.11	7.11	7.11
	Distributed	1.19	1.21	1.22	7.08	7.07	7.11
	Naive	1.60	2.07	2.67	10.42	13.25	15.14
Random	Full	6.73	6.73	6.73	0.02	0.02	0.02
	Distributed	6.72	6.88	7.13	0.02	0.02	0.16
	Naive	10.10	15.69	27.78	0.03	0.04	0.63
Hub	Full	2.46	2.46	2.46	16.44	16.44	16.44
	Distributed	4.49	4.39	3.31	14.06	14.11	14.78
	Naive	7.11	8.76	5.22	16.68	19.21	20.48
Tridiag	Full	0.02	0.02	0.02	3.78	3.78	3.78
	Distributed	0.03	0.04	0.04	3.85	4.03	4.11
	Naive	0.07	0.26	0.76	6.33	11.86	22.13

Table 9: The ℓ_∞ norm of the proposed estimators when $n = 50000$, $p = 2000$ and the block structure is used to simulate data.

	Active set			Non-active set		
	K			K		
	5	10	20	5	10	20
Full	1.88	1.88	1.88	12.81	12.81	12.81
Distributed	1.94	1.99	2.01	12.76	12.78	12.88
Naive	2.72	3.71	5.24	18.83	24.09	27.61