Overlapping Decomposition for Gaussian Graphical Modeling

Guojie Song, Lei Han, and Kunqing Xie

Abstract—Correlation based graphical models are developed to detect the dependence relationships among random variables and provide intuitive explanations for these relationships in complex systems. Most of the existing works focus on learning a single correlation based graphical model for all the random variables. However, it is difficult to understand and interpret the massive dependencies of the variables learned from a single graphical model at a global level especially when the graph is large. In order to provide a clearer understanding for the dependence relationships among a large number of random variables, in this paper, we propose the problem of estimating an overlapping decomposition for the Gaussian graphical model of a large scale to generate overlapping sub-graphical models, where strong and meaningful correlations remain in each subgraph with a small scale. Specifically, we propose a greedy algorithm to achieve the overlapping decomposition for the Gaussian graphical model. A key technique of the algorithm is that the problem of solving a $(k+1)$-node Gaussian graphical model can be approximately reduced to the problem of solving a one-step vector regularization problem based on a solved k-node Gaussian graphical model with theoretical guarantee. Based on this technique, a greedy expansion algorithm is proposed to generate the overlapping subgraphs. Moreover, we extend the proposed method to deal with dynamic graphs where the dependence relationships among random variables vary with the time. We evaluate the proposed methods on synthetic dataset and a real-life traffic dataset, and the experimental results show the superiority of the proposed methods.

Index Terms—Gaussian graphical model, correlation, overlapping decomposition, heterogeneity, dynamic

1 Introduction

Correlation based graphical models are established to meaningfully characterize the dependence or statistical relationships that exist among variables of interest and quantify them. The problem of characterizing the dependence relationships between variables in complex systems, such as economics, biological systems, traffic systems, climate change, etc., is important and fundamental. For example, economists want to know whether *burning natural gas* is a related factor with *global warming*.

The Gaussian graphical model (GGM) [1], which learns the dependence relationships among variables through the inverse of their covariance, is one of the most promising correlation based modeling methods, since the relationships revealed via the inverse covariance are important signals to tell which variables may interact with each other and find the dependence relationships existed among the variables. The Gaussian graphical model has been successfully employed in many applications, such as mining the interactions of climate attributes [2], gene regulatory network discovery [3], etc. In addition, several correlation based models on temporal evolving graphs have been proposed with applications in cross-species gene expression analysis [4], oil-production equipment stage capture [5] and climate research [6] as well.

The aforementioned methods construct a single graphical model (SGM) to capture all the dependence relationships among variables, treating all the variables together. These models are effective for understanding the relations among a small number of variables (usually in the order of tens as shown in their applications). However, when large number of variables are considered especially with only small number of available observations, interpreting a single graphical model becomes intractable. As a matter of fact, it has been shown that a correlation based analytical model with only 20 variables can be overwhelming and difficult to interpret at a global level [7], [8]. Worse still, it is much more challenging to construct and understand the complicated dependence relationships using graphical models for a relatively large graph, e.g., even with hundreds of variables, although many applications need to deal with large graphs with heterogenous and complicated relationships. For instance, a highway network in traffic systems often contains hundreds or thousands of variables, whose observations are counts of passing vehicles collected by sensors. In such traffic networks, complicated dependencies exist among the vehicle counts that the vehicles passed through one specific location may be from some other locations.

Therefore, it is essential to develop techniques to discover and understand such dependence relationships in a large network. To cope with the challenging problem, we propose to decompose a large graphical model into multiple overlapped sub-graphical models, where strong interactions exist in each subgraph with a small scale. For decomposing a graphical model, it is important to consider both the heterogeneity and homogeneity, where heterogeneity means the local correlations and homogeneity refers to the

- *G. Song and K. Xie are with the Key Laboratory of Machine Perception (Ministry of Education), EECS, Peking University, China. E-mail: gjsong@pku.edu.cn, kunqing@cis.pku.edu.cn.*
- *L. Han is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. E-mail: leihan@comp.hkbu.edu.hk.*

Manuscript received 12 Aug. 2014; revised 31 Dec. 2014; accepted 10 Feb. 2015. Date of publication 25 Feb. 2015; date of current version 2 July 2015. Recommended for acceptance by H. Xiong.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2015.2407358

overlaps between sub-graphical models. For example in traffic systems, some crucial traffic nodes may highly correlate with several different local regions, and thus these important nodes should be considered as overlap (homogeneity) by these local regions; meanwhile, we also need to find the interactions within a local region (heterogeneity).

Unfortunately, decomposing a graphical model is NP-hard even if overlaps are not allowed [7]. When we allow overlaps, the decomposition problem becomes more challenging because the search space becomes larger, which is due to more combinations of sub-graphical structures than those in the non-overlapping case.

In this paper, we address the challenging problem of estimating an overlapping decomposition for Gaussian graphical models of a large scale. We propose a novel approximation algorithm with theoretical guarantee, which is based on a local subgraph expansion strategy. The motivation begins by first decomposing the original problem of the single Gaussian graphical model, i.e., the optimization problem of the ℓ_1 penalized negative log-likelihood of the observations [9], [10], into sub-problems. Then we propose a greedy algorithm that starts with the initial small subgraphs and incrementally computes the new approximated sub-problem for each subgraph when a new node is involved.

One key technique in the decomposition of the original objective function of the single Gaussian graphical model is that the problem of solving a $(k+1)$-node Gaussian graphical model is approximated to the problem of solving a one-step vector regularization problem based on a solved k-node graphical model, referred to the additive expanding property, while the results obtained from the approximation step also enjoys good asymptotic properties. We provide detailed theoretical analysis for this key technique, which guarantees the feasibility of the overlapping decomposition method. Based on the above technique, we then propose a greedy expansion algorithm for generating the overlapping sub-graphical models.

The overlapping decomposition technique is so far developed for random variables from static graphs. However, in many real world applications, the graph evolves over time. For example, the dependence relationships among the vehicle flows in a highway network of traffic systems change frequently during one day, leading to the alternation of peak and leisure hours in the traffic. To address this dynamic problem, we then extend the proposed method to deal with sequence of graphs, where the dependence relationships among random variables vary over time. We assume the variable covariance changes smoothly over time, and then apply a weighted covariance matrix for the overlapping decomposition algorithm to capture the changes of the dependence relationships smoothly.

We evaluate the proposed method with two sets of experiments: (1) we empirically verify the properties of the proposed overlapping decomposition method on synthetic networks, and compare with the single graphical model [9] and the non-overlapping decomposition method (which is a special case of the proposed overlapping decomposition method). The experimental results demonstrate the advantages of our techniques; (2) we evaluate the proposed techniques on real-life traffic data by learning the dependence relationships among traffic observation points (e.g., on-ramps or off-ramps) and detecting the traffic regularity in large traffic networks. Both static and dynamic settings are evaluated in this traffic dataset.

In summary, our contributions are five-fold:

1) To our knowledge, our work is the first one that decomposes the single Gaussian graphical model into sub-graphical models according to overlapping subgraphs for both static and dynamic settings.
2) We propose an additive expanding technique to modify the original problem for the single Gaussian graphical model to one-step vector regularization sub-problems, and demonstrate its asymptotic properties with detailed theoretical analysis.
3) We propose a constrained greedy subgraph expansion algorithm for generating the overlapping subgraphs as well as learning the dependence relationships within each subgraph simultaneously.
4) We extend the proposed method to deal with dynamic graphs, where the dependence relationships among random variables change over time.
5) We evaluate our method on both synthetic and real-life traffic datasets. Experimental results show the effectiveness and superiority of our overlapping decomposition technique.

This paper is an improved version of the conference paper [11]. The rest of the contents is organized as follows. In Section 2, we briefly review closely related works. Section 3 presents the preliminaries and the problem statement. In Section 4, we present the proposed method. In Section 5, we introduce how to extend the proposed method to deal with dynamic graphs. Experimental studies are reported in Section 6. We conclude this paper and present future directions in Section 7.

2 Related Work

Most of the existing works on correlation based graphical models build a single graphical model. This renders them impractical to interpret and understand relatively large graphs. To cope with larger set of random variables, Ruan et al. [7] propose to cluster the variables into groups such that strong dependence relations appear only among the variables within a group while the relations between inter-group variables are ignored. The clustering problem is formulated as a regression coefficient sparsification problem for the decomposition of a graphical model. However, this approach only considers non-overlapping decompositions while ignoring the overlap between subgraphs, which, however, exists in many real-world applications. Moreover, the approach [7] is based on the Vector Autoregressive model, a type of temporal graphical model, rather than Gaussian graphical model as we consider in this work. A recent work [12] proposes to decompose a sparse Gaussian graphical model into disjoint connected components according to some threshold, while the solution obtained from the subgraphs is shown to be equivalent to the solution of the original problem. However, the structures of the subgraphs directly relay on the threshold parameter, and once at least one edge exists between two subgraphs under some threshold, the two subgraphs are indecomposable and have to be treated as an entire one. In other words, overlap is not

allowed among the subgraphs, otherwise the components with overlap should be considered as an entire component. Therefore, this approach is less applicable in real applications, since it is hard to find a threshold parameter to exactly decompose the graph into the desired disjoint components while preserving the global optimality. Another work [13] also proposes a local clustering algorithm for massive graphs, however, the nodes considered in their graphs are not random variables and their model does not aim to learn the dependence relationships among the variables.

For the Gaussian graphical model with ℓ_1 penalized negative log-likelihood as the objective function, some efficient algorithms have also been proposed to learn large scale graphs recently [14], [15], [16], [17], [18]. In these works, even a single graph with millions of variables is tractable to be solved, and their algorithms are guaranteed to converge to the optimal solution of the ℓ_1 penalized negative log-likelihood minimization problem. However, all these works focus on efficiently solving the original single Gaussian graphical model, while none of them tries to interpret and understand the obtained dependence relationships in the graph. As a matter of fact, it is impossible to understand a single graph at that scale with millions of variables for domain experts, and even only tens of variables are difficult to interpret at a global level [7], [8]. Instead of solving the single Gaussian graphical model, we propose to decompose it into small overlapping components, and interpret the entire graph with strong interacted variables existed in each subgraph.

Our work is closely related to the joint estimation methods for multiple graphical models that share common structures [19], [20]. The joint estimation methods [19], [20] are proposed to learn multiple graphical models on the data from different categories but with the same set of features (variables), considering both the underlying homogeneity and heterogeneity of networks. They estimate multiple graphical models for different categories of the features, but not decomposing the features themselves. We proceed to use the example application scenario in [19] to illustrate these methods. Consider a set of webpages collected from computer science departments of universities, and we want to find the correlations between selected keywords (e.g., 'book', 'model', 'problem', etc.) appearing in the collection. These keywords can be treated as features, and appear in webpages of different categories, such as 'student', 'faculty', 'project', etc. These features may display different dependence structures for different categories while sharing some common correlations across categories. The joint estimation methods cannot be applied to solve our problem, and they cannot be employed to discover the complicated dependence relationships in large feature networks. First, these methods do not consider the decomposition on features of a large graph. Second, these methods are developed for graphs with a small number of features (in the order of tens).

Our proposed approach is also related to detecting overlapping community structures [21]. Community structure detection aims to group similar nodes together based on known distance measurements or correlations of nodes themselves. In contrast, in our problem we aim to uncover the dependence relationships among the nodes, furthermore find subgraphs by the measurement that based on these relations but not the known properties of nodes themselves. Thus, our problem is essentially different from community structure detection.

3 Problem Setup

3.1 Preliminary: Gaussian Graphical Model

As a member of the correlation based graphical model, Gaussian graphical model assumes the joint distribution of the variables to be Gaussian. In GGM, the dependence structure is determined from the covariance matrix of the variables, and a natural way to evaluate the dependence relationships is to estimate the inverse of the covariance matrix [1], [22], [23]. Consider p random variables $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p)$, and each variable $\boldsymbol{x}_i$ has n observations $\boldsymbol{x}_i = (x_i^1, \ldots, x_i^n)^T$, where we usually have $n \gg p$. Without loss of generality, we assume $\boldsymbol{X}$ follows a multivariate Gaussian distribution $N(\boldsymbol{\mu}, \Sigma)$, where the mean vector $\boldsymbol{\mu}$ is p-dimensional and each element in covariance matrix Σ is the expected value $\Sigma_{ij} = E[(X_i - \mu_i)(X_j - \mu_j)]$. The precision matrix Ω is the inverse of the covariance matrix, i.e., $\Omega = \Sigma^{-1}$, which reveals the dependence relationships among the variables. There exists a dependence relationship between variables x_i and x_j iff $\Omega_{ij} \neq 0$ [1], [22]. Therefore, the key problem is to calculate Ω. The estimation of Ω can be obtained by minimizing the ℓ_1 penalized negative log-likelihood criterion [9], [10],

$$\widehat{\Omega} = \arg\min_{\Omega \succeq 0} \quad tr(\widehat{\Sigma}\Omega) - \log|\Omega| + \lambda \sum_{i \neq j} |\theta_{ij}|, \tag{1}$$

where θ_{ij} is the (i,j)th element in Ω; $\Omega \succeq 0$ means that Ω is positive semi-definite (PSD) matrix; $\widehat{\Sigma}$ is the sample covariance matrix obtained from the input $\boldsymbol{X}$; $|\cdot|$ and $tr(\cdot)$ are the determinant and the trace operators in matrix calculus, respectively; λ is a tuning parameter. The term $tr(\widehat{\Sigma}\Omega) - \log|\Omega|$ of Eq. (1) corresponds to the negative log-likelihood of the observations of a Gaussian graphical model. The term $\lambda \sum_{i \neq j} |\theta_{ij}|$ is called a ℓ_1 penalty, which is to shrink some of the off-diagonal elements in $\widehat{\Omega}$ to zero. The tuning parameter λ controls the sparsity of $\widehat{\Omega}$. This minimization problem can be efficiently solved by the algorithms proposed in [9], [10], [14], [15], [16], [17], [18].

3.2 Problem Definition

Problem Definition. *Given p random variables $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p)$, where p is large and each variable $\boldsymbol{x}_i$ has n observations, i.e. $\boldsymbol{x}_i = (x_i^1, \ldots, x_i^n)^T$, we aim to learn the dependence relationships among these variables by decomposing the random variables into overlapping subsets.*

In other words, we aim to encode the structure of $\boldsymbol{X}$ with an undirected graph $G = (V, E)$, where each node v in $V = \{v_1, \ldots, v_p\}$ corresponds to a variable in $\boldsymbol{X}$. The edge set E indicates the dependence relations between any two variables. More precisely, if $\boldsymbol{x}_i$ is correlated to $\boldsymbol{x}_j$, then edge e_{ij} is included in E. Thus, our objective is to obtain E. As introduced in Section 3.1, E can be constructed by

estimating the precision matrix Ω of $\boldsymbol{X}$. We add an edge e_{ij} to E iff $\Omega_{ij} \neq 0$.

Instead of creating a single Gaussian graphical model for G directly, we propose to construct K Gaussian sub-graphical models with overlaps to discover the dependence relationships among variables with enhanced interpretable ability. Each subgraph is denoted as $g_i = (SV_i, SE_i), 1 \leq i \leq K$. The dependence relationships reflected in E, $E = \bigcup_i SE_i$, are the output.

The challenge here is to generate the K sub-graphical models and allow some variables exist in more than one sub-graphical models, i.e. overlap exists, and then estimate Ω_i for each SE_i. To achieve this, we propose a novel algorithm for solving the overlapping decomposition problem, where a core step called *local subgraph expansion* is used. Our algorithm adopts a bottom-up strategy that expands the initial subgraphs by adding selected nodes gradually until the structure of overlapping subgraphs will become stable.

During this process, a *key* operation is to choose whether to include a new node in a subgraph. This operation is invoked many times, and calls for efficient techniques. Specifically, assume that there is a k-node subgraph whose inner dependence relationships have been detected, then we want to know whether a node v_{k+1} should be added to it. A straightforward method is creating a new Gaussian graphical model on all the $k+1$ nodes. However, this straightforward method ignores the known dependence relationships in the k-node subgraph and is a waste of computations. Thus, a natural question is whether we can reuse the known dependence relationships in a subgraph to detect the relationship between a new node and the subgraph. In the next section, we present the proposed approximation method with theoretical guarantees for this key operation, and then develop a greedy algorithm to construct the subgraphs.

4 The Proposed Method

In this section, we propose two techniques. The first technique is used to check whether a new variable (node) should be included in a subgraph. This technique modifies the penalized log-likelihood criterion in Eq. (1) so that it can be incrementally expanded to accommodate new nodes. We call it additive penalized log-likelihood expansion (APLE). In Section 4.2, we discuss the asymptotic properties for the APLE technique, which also motivates the necessity to decompose a large graphical model into sub-graphical models from a theoretical view.

The second technique is a local greedy approach presented in Section 4.3. We define a fitness function based on the APLE approach. Moreover, taking into account some structure constraints on the subgraphs, we develop the Constraint Greedy Subgraph Expansion (CGSE) algorithm, which can achieve the *local subgraph expansion* process.

4.1 Additive Penalized Log-Likelihood Expansion

Assume that the problem in Eq. (1) for a k-node graph has already been solved, and now a new variable x_{k+1} is added into the solved k-node graph. Instead of solving the original single Gaussian graphical model in Eq. (1) for a $(k+1)$-node graph, we propose to utilize the solution of the k-node graph to construct a new solution for the $(k+1)$-node graph, since the dependence relationships existed in the k-node graph should almost remain stable after incorporating a new node for a static graph. By denoting the solution of the k-node graph as $\widehat{\Omega}^{(k)}$, we propose to construct

$$\Omega^{(k+1)} = \begin{bmatrix} \widehat{\Omega}^{(k)} & \boldsymbol{\theta} \\ \boldsymbol{\theta}^T & \theta_{k+1} \end{bmatrix}, \tag{2}$$

where $\boldsymbol{\theta}$ is a $k \times 1$ vector that reveals the dependence relationship between x_{k+1} and $\{x_1, \ldots, x_k\}$ which we have to learn. In order to detect the relationship among $\widehat{\Omega}^{(k)}$, $\boldsymbol{\theta}$ and θ_{k+1}, we have to plug Eq. (2) into the original problem in Eq. (1).

Denote the penalized negative log-likelihood criterion for a k-node graph in Eq. (1) as $\ell(\Omega^{(k)})$. Now for a $(k+1)$-node graph, we have

$$\ell(\Omega^{(k+1)}) = tr(\widehat{\Sigma}^{(k+1)}\Omega^{(k+1)}) - \log|\Omega^{(k+1)}| + \lambda \sum_{i \neq j}^{k+1} |\theta_{ij}|,$$

where $\Omega^{(k+1)}$ has the form in Eq. (2). For notional simplicity, we introduce three symbols to denote the items:

$$I_1^{(k+1)} = tr(\widehat{\Sigma}^{(k+1)}\Omega^{(k+1)}),\ I_2^{(k+1)} = \log|\Omega^{(k+1)}|,$$
$$I_3^{(k+1)} = \lambda \sum_{i \neq j}^{k+1} |\theta_{ij}|.$$

We then unfold $\widehat{\Sigma}^{(k+1)}$ and $\Omega^{(k+1)}$ into block matrices and get

$$\begin{aligned} I_1^{(k+1)} &= tr\left(\begin{bmatrix} \widehat{\Sigma}^{(k)} & \widehat{\varepsilon} \\ \widehat{\varepsilon}^T & \widehat{\varepsilon}_{k+1} \end{bmatrix} \cdot \begin{bmatrix} \widehat{\Omega}^{(k)} & \boldsymbol{\theta} \\ \boldsymbol{\theta}^T & \theta_{k+1} \end{bmatrix} \right) \\ &= tr\left(\begin{bmatrix} \widehat{\Sigma}^{(k)}\widehat{\Omega}^{(k)} + \widehat{\varepsilon}\boldsymbol{\theta}^T & \widehat{\Sigma}^{(k)}\boldsymbol{\theta} + \theta_{k+1}\widehat{\varepsilon} \\ \widehat{\varepsilon}^T\widehat{\Omega}^{(k)} + \widehat{\varepsilon}_{k+1}\boldsymbol{\theta}^T & \widehat{\varepsilon}^T\boldsymbol{\theta} + \widehat{\varepsilon}_{k+1}\theta_{k+1} \end{bmatrix} \right) \\ &= tr(\widehat{\Sigma}^{(k)}\widehat{\Omega}^{(k)}) + tr(\widehat{\varepsilon}\boldsymbol{\theta}^T) + \widehat{\varepsilon}^T\boldsymbol{\theta} + \widehat{\varepsilon}_{k+1}\theta_{k+1} \\ &= I_1^{(k)} + 2\widehat{\varepsilon}^T\boldsymbol{\theta} + \widehat{\varepsilon}_{k+1}\theta_{k+1}, \end{aligned}$$

$$\begin{aligned} I_2^{(k+1)} &= \log \begin{vmatrix} \widehat{\Omega}^{(k)} & \boldsymbol{\theta} \\ \boldsymbol{\theta}^T & \theta_{k+1} \end{vmatrix} \\ &= \log\left(\left|\widehat{\Omega}^{(k)}\right| \cdot \left|\theta_{k+1} - \boldsymbol{\theta}^T \left(\widehat{\Omega}^{(k)}\right)^{-1}\boldsymbol{\theta}\right| \right) \\ &= I_2^{(k)} + \log\left(\theta_{k+1} - \boldsymbol{\theta}^T \left(\widehat{\Omega}^{(k)}\right)^{-1}\boldsymbol{\theta}\right), \end{aligned}$$

$$I_3^{(k+1)} = \lambda \sum_{i \neq j}^{k+1} |\theta_{ij}| = I_3^{(k)} + 2\lambda\|\boldsymbol{\theta}\|_1,$$

where $(\widehat{\varepsilon}, \widehat{\varepsilon}_{k+1})$ is the sample covariance vector between x_{k+1} and $\{x_1, \ldots, x_{k+1}\}$, and $I_1^{(k)} + I_2^{(k)} + I_3^{(k)}$ defines $\ell(\widehat{\Omega}^{(k)})$. The derivation of $I_2^{(k+1)}$ is obtained by the Leibniz formula. Finally, we can get

$$\begin{aligned} \ell(\Omega^{(k+1)}) = \ell(\widehat{\Omega}^{(k)}) &+ 2\widehat{\varepsilon}^T\boldsymbol{\theta} + \widehat{\varepsilon}_{k+1}\theta_{k+1} \\ &- \log\left(\theta_{k+1} - \boldsymbol{\theta}^T \left(\widehat{\Omega}^{(k)}\right)^{(-1)}\boldsymbol{\theta}\right) + 2\lambda\|\boldsymbol{\theta}\|_1. \end{aligned} \tag{3}$$

Note that in the problem of the $(k+1)$-node graph, $\widehat{\Omega}^{(k)}$ is known as a constant, therefore, we only have to solve the

following problem for $(\boldsymbol{\theta}, \theta_{k+1})$:

$$(\widehat{\boldsymbol{\theta}}, \widehat{\theta}_{k+1}) = \arg\min_{\boldsymbol{\theta}, \theta_{k+1}} \; 2\widehat{\boldsymbol{\varepsilon}}^T\boldsymbol{\theta} + \widehat{\varepsilon}_{k+1}\theta_{k+1} - \log\big(\theta_{k+1} - \boldsymbol{\theta}^T\big(\widehat{\Omega}^{(k)}\big)^{-1}\boldsymbol{\theta}\big) + 2\lambda_\theta\|\boldsymbol{\theta}\|_1. \tag{4}$$

In problem (4), it potentially requires that $\theta_{k+1} > \boldsymbol{\theta}^T(\widehat{\Omega}^{(k)})^{-1}\boldsymbol{\theta} > 0$ to make $\Omega^{(k+1)}$ PSD since $\widehat{\Omega}^{(k)}$ is assumed to be a solved PSD matrix. Similarly, we use $\ell(\boldsymbol{\theta}, \theta_{k+1})$ to represent the objective in problem (4). Now, we have

$$\ell(\Omega^{(k+1)}) = \ell(\widehat{\Omega}^{(k)}) + \ell(\boldsymbol{\theta}, \theta_{k+1}). \tag{5}$$

So far, it is understandable that to obtain $\widehat{\Omega}^{(k+1)}$ with the formulation in Eq. (2) based on a known $\widehat{\Omega}^{(k)}$, we just need to solve $\ell(\boldsymbol{\theta}, \theta_{k+1})$ of problem (5) for an additional *vector* optimization problem. We next proceed to explorer the properties of the vector regularization problem (5), which is summarized in the following theorem.

Theorem 1. *With given PSD $\widehat{\Omega}^{(k)}$, problem (4) is convex w.r.t. $\boldsymbol{\theta}$, θ_{k+1} and $(\boldsymbol{\theta}, \theta_{k+1})$, respectively.*

Proof. Let $A = (\widehat{\Omega}^{(k)})^{-1}$. The convexity of problem (4) can be achieved by directly calculating the Hessian matrix H of the objective $l(\boldsymbol{\theta}, \theta_{k+1})$:

$$H = \begin{bmatrix} \frac{\partial^2 l}{\partial \boldsymbol{\theta}^2} & \frac{\partial^2 l}{\partial \boldsymbol{\theta}\partial\theta_{k+1}} \\ \frac{\partial^2 l}{\partial\theta_{k+1}\partial\boldsymbol{\theta}} & \frac{\partial^2 l}{\partial\theta_{k+1}^2} \end{bmatrix} = \begin{bmatrix} \frac{2(\theta_{k+1} + \boldsymbol{\theta}^T A\boldsymbol{\theta})A}{(\theta_{k+1} - \boldsymbol{\theta}^T A\boldsymbol{\theta})^2} & \frac{-2A\boldsymbol{\theta}}{(\theta_{k+1} - \boldsymbol{\theta}^T A\boldsymbol{\theta})^2} \\ \frac{-2\boldsymbol{\theta}^T A}{(\theta_{k+1} - \boldsymbol{\theta}^T A\boldsymbol{\theta})^2} & \frac{1}{(\theta_{k+1} - \boldsymbol{\theta}^T A\boldsymbol{\theta})^2} \end{bmatrix}.$$

Since $\widehat{\Omega}^{(k)}$ is a given PSD matrix, and so is A, it is easy to verify that $l(\boldsymbol{\theta}, \theta_{k+1})$ is convex w.r.t. $\boldsymbol{\theta}$ and θ_{k+1} respectively, because $\frac{\partial^2 l}{\partial \boldsymbol{\theta}^2}$ is PSD and $\frac{\partial^2 l}{\partial \theta_{k+1}^2}$ is positive. Now, we have to show H is PSD matrix to see the joint convexity of $l(\boldsymbol{\theta}, \theta_{k+1})$. For any vector $[\boldsymbol{\alpha}; \beta] \in \mathbb{R}^{k+1}$, where $\boldsymbol{\alpha} \in \mathbb{R}^k$, we have

$$[\boldsymbol{\alpha} \quad \beta] H \begin{bmatrix} \boldsymbol{\alpha} \\ \beta \end{bmatrix} = \frac{2(\theta_{k+1} - \boldsymbol{\theta}^T A\boldsymbol{\theta})\boldsymbol{\alpha}^T A\boldsymbol{\alpha} + (2\boldsymbol{\theta}^T A\boldsymbol{\alpha} - \beta)^2}{(\theta_{k+1} - \boldsymbol{\theta}^T A\boldsymbol{\theta})^2} \geq 0.$$

Therefore, H is PSD matrix. □

From Theorem 1, we know that problem (4) is convex w.r.t. $\boldsymbol{\theta}$, θ_{k+1} and $(\boldsymbol{\theta}, \theta_{k+1})$ respectively, and therefore this problem can be solved efficiently using the generalized ℓ_1 solver in [24]. To be concise in the following analysis, we denote $\bar{\boldsymbol{\theta}} = (\boldsymbol{\theta}, \theta_{k+1})$.

The solution constructed from Eq. (2) is an approximation of the optimal solution of the original Gaussian graphical model defined in Eq. (1) because we restrict the form of $\Omega^{(k+1)}$ in Eq. (2). However, we show in the next section that the solution in Eq. (2) can be asymptotically consistent with the true precision matrix when the number of samples n is sufficiently large. Such theoretical results provide important guarantee for the feasibility of the APLE technique.

4.2 Asymptotic Analysis of APLE

In this part, we discuss the asymptotic property of the APLE technique, which guarantees that $\widehat{\Omega}^{(k+1)}$ can be asymptotically consistent with the true precision matrix by choosing an appropriate λ_θ when n is sufficiently large. A detailed asymptotic analysis of Eq. (1) has been discussed in [25]. Inspired by it, we establish the analysis to our APLE approach as follows.

Let the true precision matrix be Ω_0, and let the true covariance matrix be Σ_0 ($\Omega_0 = (\Sigma_0)^{-1}$) for any corresponding size, as well as true precision vector $\bar{\boldsymbol{\theta}}_0 = (\boldsymbol{\theta}_0, \theta_{k+1}^0)$ and true covariance vector $\bar{\varepsilon}_0 = (\varepsilon_0, \varepsilon_{k+1}^0)$ for size $k+1$. We make the following assumptions according to [25]:

- A1: There exists a constant η such that $0 < \varphi_{max}(\Omega_0^{(k)})$, $\varphi_{max}(\Omega_0^{(k+1)}) \leq \eta$, where $\varphi_{max}(\cdot)$ denotes the maximum eigenvalue;
- A2: There exist constants σ_1 and σ_2 such that $\sigma_1 \leq \widehat{\theta}_{k+1} \leq \sigma_2$ will guarantee $\widehat{\Omega}^{(k+1)}$ positive semidefinite;
- A3: $\widehat{\Omega}^{(k)}$ in problem (4) is a root-n consistent solution [25] for the k-node graph.

Now we have the following theorem.

Theorem 2. *Let $\bar{\varepsilon} = (\widehat{\varepsilon}, \widehat{\varepsilon}_{k+1})$ and let $\bar{\boldsymbol{\theta}} = (\widehat{\boldsymbol{\theta}}, \widehat{\theta}_{k+1})$ be the optimal solution for problem (4). Under A1-A3, if $\lambda_\theta = C_0\sqrt{\frac{\log(k+1)}{n}}$, where C_0 is a positive constant, then*

$$\|\bar{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}_0\|_2 = O_P\left(\sqrt{\frac{(k+1)\log(k+1)}{n}}\right), \tag{6}$$

where $O_P(\cdot)$ is the order in probability.

Proof. Let $G(\Delta_{\bar{\theta}}) = \ell(\bar{\boldsymbol{\theta}}_0 + \Delta_{\bar{\theta}}) - \ell(\bar{\boldsymbol{\theta}}_0)$. Assume that there exists a bounded convex set

$$\mathcal{G} = \{\Delta_{\bar{\theta}} : \|\Delta_{\bar{\theta}}\|_2 \leq Mr_n\},$$

where M is a positive constant and

$$r_n = \sqrt{\frac{(k+1)\log(k+1)}{n}} \to 0 \quad (n \to \infty).$$

Note that $G(\Delta_{\bar{\theta}})$ is a convex function, if we demonstrate that G is positive everywhere on the boundary $\partial\mathcal{G}$ ($\|\Delta_{\bar{\theta}}\|_2 = Mr_n$), then G has a minimum inside $\mathcal{G}$. Actually,

$$\begin{aligned} G(\Delta_{\bar{\theta}}) = \ell(\bar{\boldsymbol{\theta}}_0 + \Delta_{\bar{\theta}}) - \ell(\bar{\boldsymbol{\theta}}_0) &= 2\widehat{\boldsymbol{\varepsilon}}^T\Delta_\theta + \widehat{\varepsilon}_{k+1}\Delta_{\theta_{k+1}} \\ &- \Big(\log\Big(\theta_{k+1}^0 - (\boldsymbol{\theta}_0^T + \Delta_\theta)(\widehat{\Omega}^{(k)})^{-1}(\boldsymbol{\theta}_0 + \Delta_\theta)\Big) \\ &- \log\Big(\theta_{k+1}^0 - \boldsymbol{\theta}_0^T(\widehat{\Omega}^{(k)})^{-1}\boldsymbol{\theta}_0\Big)\Big) \\ &+ 2\lambda_\theta(\|\boldsymbol{\theta}_0 + \Delta_\theta\|_1 - \|\boldsymbol{\theta}_0\|_1). \end{aligned} \tag{7}$$

For the subtraction of the logarithm terms in Eq. (7), denote $f(\bar{\boldsymbol{\theta}}) = \log(\theta_{k+1} - \boldsymbol{\theta}^T(\widehat{\Omega}^{(k)})^{-1}\boldsymbol{\theta})$. Because $f(\bar{\boldsymbol{\theta}}) = I_2^{(k+1)} - I_2^{(k)}$, we have

$$\begin{aligned} &f(\bar{\boldsymbol{\theta}}_0 + \Delta_{\bar{\theta}}) - f(\bar{\boldsymbol{\theta}}_0) \\ &= \Big(\log|\Omega_0^{(k+1)} + \Delta^{(k+1)}| - \log|\Omega_0^{(k+1)}|\Big) \\ &\quad - \Big(\log|\Omega_0^{(k)} + \Delta^{(k)}| - \log|\Omega_0^{(k)}|\Big). \end{aligned}$$

As has been proved by [25], for any Ω that satisfies A1, we have

$$\log|\Omega+\Delta|-\log|\Omega| = tr(\Sigma\Delta)-\widetilde{\Delta}^T\left[\int_0^1(1-v)(\Omega+v\Delta)^{-1}\otimes(\Omega+v\Delta)^{-1}dv\right]\widetilde{\Delta},$$

where

$$F=\widetilde{\Delta}^T\left[\int_0^1(1-v)(\Omega+v\Delta)^{-1}\otimes(\Omega+v\Delta)^{-1}dv\right]\widetilde{\Delta} \geq \frac{1}{4\eta^2}\|\Delta\|_F^2.$$

Thus we have

$$f(\bar{\boldsymbol{\theta}}_0+\Delta_{\bar{\theta}})-f(\bar{\boldsymbol{\theta}}_0) = \left(tr\left(\Sigma_0^{(k+1)}\Delta^{(k+1)}\right)-tr\left(\Sigma_0^{(k)}\Delta^{(k)}\right)\right)-(F^{(k+1)}-F^{(k)}) = 2\varepsilon_0^T\Delta_\theta+\varepsilon_{k+1}^0\Delta_{\theta_{k+1}}-(F^{(k+1)}-F^{(k)}).$$

Then we can get

$$G(\Delta_{\bar{\theta}})=\left(\bar{\varepsilon}^T-\bar{\varepsilon}_0^T\right)\Delta_{\bar{\theta}}+\left(\widehat{\varepsilon}^T-\varepsilon_0^T\right)\Delta_\theta+(F^{(k+1)}-F^{(k)}) +2\lambda_\theta(\|\boldsymbol{\theta}_0+\Delta_\theta\|_1-\|\boldsymbol{\theta}_0\|_1).$$

For each item in $G(\Delta_{\bar{\theta}})$, we have the following boundaries

$$B1:|\left(\bar{\varepsilon}^T-\bar{\varepsilon}_0^T\right)\Delta_{\bar{\theta}}|\leq C_1\sqrt{\frac{\log(k+1)}{n}}\|\Delta_{\bar{\theta}}\|_1 \leq C_1\sqrt{\frac{(k+1)\log(k+1)}{n}}\|\Delta_{\bar{\theta}}\|_2,$$

$$|\left(\widehat{\varepsilon}^T-\varepsilon_0^T\right)\Delta_\theta|\leq C_1\sqrt{\frac{\log k}{n}}\|\Delta_\theta\|_1\leq C_1\sqrt{\frac{k\log k}{n}}\|\Delta_\theta\|_2,$$

$$B2:F^{(k+1)}-F^{(k)}\geq\frac{1}{4\eta^2}\left(\|\Delta^{(k+1)}\|_2^2-\|\Delta^{(k)}\|_2^2\right)\geq\frac{1}{4\eta^2}\|\Delta_{\bar{\theta}}\|_2^2,$$

$$B3:\lambda_\theta(\|\boldsymbol{\theta}_0+\Delta_\theta\|_1-\|\boldsymbol{\theta}_0\|_1)\leq\lambda_\theta\|\Delta_\theta\|_1\leq\lambda_\theta\sqrt{k}\|\Delta_\theta\|_2 \leq\lambda_\theta\sqrt{(k+1)}\|\Delta_{\bar{\theta}}\|_2,$$

where the inequalities in B1 are boundaries from [25], and B3 can be obtained by the mean inequalities. Combine all the above items and finally we can get

$$G(\Delta_{\bar{\theta}})\geq\frac{1}{4\eta^2}\|\Delta_{\bar{\theta}}\|_2^2-2C_1\sqrt{\frac{(k+1)\log(k+1)}{n}}\|\Delta_{\bar{\theta}}\|_2 -2\lambda_\theta\sqrt{(k+1)}\|\Delta_{\bar{\theta}}\|_2 =\|\Delta_{\bar{\theta}}\|_2^2\left(\frac{1}{4\eta^2}-\frac{2C_1\sqrt{\frac{(k+1)\log(k+1)}{n}}+2\sqrt{k+1}\lambda_\theta}{\|\Delta_{\bar{\theta}}\|_2}\right).$$

Take $\lambda_\theta=C_0\sqrt{\frac{\log(k+1)}{n}}$,

$$G(\Delta_{\bar{\theta}})\geq\|\Delta_{\bar{\theta}}\|_2^2\left(\frac{1}{2\eta^2}-\frac{2C_1+2C_0}{M}\right)$$

for M sufficiently large we can get $G(\Delta_{\bar{\theta}})>0$. □

According to the theorems in [19], [25], a root-n consistent solution $\widehat{\Omega}^{(k)}$ for the k-node graph satisfies

$$\left\|\widehat{\Omega}^{(k)}-\Omega_0^{(k)}\right\|_F=O_P\left(\sqrt{\frac{(k+s_k)\log k}{n}}\right), \tag{8}$$

where s_k is the number of non-zero off-diagonal elements in $\Omega_0^{(k)}$. For $\widehat{\Omega}^{(k+1)}$, we have

$$\left\|\widehat{\Omega}^{(k+1)}-\Omega_0^{(k+1)}\right\|_F^2\leq\left\|\widehat{\Omega}^{(k)}-\Omega_0^{(k)}\right\|_F^2+2\left\|\bar{\boldsymbol{\theta}}-\bar{\boldsymbol{\theta}}_0\right\|_2^2.$$

Combine with our Theorem 2 and note that $0\leq s_{k+1}-s_k\leq 2k$, then it can be found that

$$\left\|\widehat{\Omega}^{(k+1)}-\Omega_0^{(k+1)}\right\|_F^2 \leq O_P\left(\frac{(k+s_k)\log k}{n}\right)+O_P\left(\frac{2(k+1)\log(k+1)}{n}\right) =O_P\left(\frac{(k+1+s_{k+1})\log(k+1)}{n}\right),$$

which is in line with the asymptotic property of the optimal solution obtained from the original problem (1) for the $(k+1)$-node graph [19], [25]. Such results verifies that the approximated solution constructed from the APLE technique also enjoys good asymptotic properties.

It is worth mentioning that both Eq. (8) and our Theorem 2 are in line with the motivation of the decomposition on large scale graphical model. Note that both of them show that when the number of variables k is relatively large or exceeds the number of observations n of each variable, the error on the estimation will increase dramatically and the asymptotic properties may not hold any more since the assumptions that a sufficiently large n is not satisfied. This suggests that, in addition to the interpretable ability, when we consider only a single graphical model on a large network, the result will be inaccurate especially when there are not enough observations to support such a large graphical model. An example is that traffic systems often contain hundreds of ramps (variables), and the number of the observations for each ramp is limited by the sampling quantity. The periodicity of traffic behaviors is often measured by days. Thus, if we want to know the dependence relationships between observations at a specific time period in a day, we can just get one value for each ramp one day. Therefore, the decomposition of a large graphical model is necessary.

One disadvantage of solving problem (4) is that it requires calculating the inverse of $\widehat{\Omega}^{(k)}$. As mentioned previously, problem (4) is involved many times to achieve the *local subgraph expansion*, and it has to be solved efficiently. To accelerate the procedure, we propose to substitute $(\widehat{\Omega}^{(k)})^{-1}$ with the sample covariance $\widehat{\Sigma}^{(k)}$, where problem (4) becomes

$$(\widehat{\boldsymbol{\theta}},\widehat{\theta}_{k+1})=\arg\min_{\boldsymbol{\theta},\theta_{k+1}}\ 2\widehat{\varepsilon}^T\boldsymbol{\theta}+\widehat{\varepsilon}_{k+1}\theta_{k+1}-\log\left(\theta_{k+1}-\boldsymbol{\theta}^T\widehat{\Sigma}^{(k)}\boldsymbol{\theta}\right) +2\lambda_\theta\|\boldsymbol{\theta}\|_1. \tag{9}$$

The advantage of solving problem (9) instead of problem (4) is that: (1) $\widehat{\Sigma}^{(k)}$ can be obtained directly from the input

and no matrix inversion needs to be calculated; (2) when n is sufficiently large, the optimal $\widehat{\Omega}^{(k)}$ is asymptotically consistent with the true precision matrix $\Omega_0^{(k)}$, and the sample covariance matrix $\widehat{\Sigma}^{(k)} \to \Sigma_0^{(k)} = (\Omega_0^{(k)})^{-1}$, then the properties in Theorem 2 can still hold. Therefore, in our implementations, we will solve problem (9) instead of problem (4).

4.3 Constraint Greedy Subgraph Expansion

We present the algorithm for the *local subgraph expansion* process based on our APLE approach. We consider some constraints corresponding to the structures of the subgraphs, and apply them to the *local subgraph expansion* process in this section.

Constraints. When a new node (variable) x_{new} joins in a solved k-node subgraph g to do expansion, based on APLE we can obtain a new dependence vector between x_{new} and g, i.e. $\widehat{\boldsymbol{\theta}}_{new} = APLE(g, x_{new}, \lambda_\theta) \in \mathbb{R}^k$ (we do not consider $\widehat{\theta}_{k+1}$). Let $E_{\widehat{\theta}} = \{i : \widehat{\theta}_i \neq 0, 1 \leq i \leq k\}$, we define the fitness as

$$Fitness(\widehat{\boldsymbol{\theta}}_{new}) = e^{-\gamma o} \frac{|E_{\widehat{\theta}}|}{k}, \tag{10}$$

where o is the number of subgraphs to which node v_{new} has been mapped, and thus γ controls the degree of overlaps, which can be regarded as a constraint. With γ, we have that the correlation contributions of v_{new} to other subgraphs reduce as the number of subgraphs to which v_{new} has already been mapped increases.

Because the fitness in Eq. (10) is always nonnegative, a threshold ϵ_f should be given as the minimum accepted fitness, which is actually a constraint on the size of each subgraph. After each iteration of the expansion, we check whether there are near-duplicated subgraphs based on the following equation:

$$\max\left\{\frac{|SV_i \cap SV_j|}{|SV_i|}, \frac{|SV_i \cap SV_j|}{|SV_j|}\right\} > \epsilon_o, \tag{11}$$

where SV_i is the set of nodes in subgraph g_i and ϵ_o is the combination threshold. We combine subgraphs g_i and g_j into a new subgraph if the above equation is satisfied. Here ϵ_o balances the sizes of overlaps.

Adaption of λ_θ. It has been mentioned above that as the subgraph expands, λ_θ has to be adapted to make APLE satisfy Theorem 2. According to Theorem 2, we know that $\lambda_\theta^{(k)} = C_0\sqrt{\frac{\log(k+1)}{n}}$, and thus when k expands to $k+1$, we have

$$\lambda_\theta^{(k+1)} = C_0\sqrt{\frac{\log(k+2)}{n}} = \sqrt{\log_{k+1}(k+2)}\lambda_\theta^{(k)}. \tag{12}$$

In the following algorithm we will update λ_θ based on Eq. (12).

The proposed Constraint Greedy Subgraph Expansion algorithm is then outlined in Algorithm 1.

Algorithm explanation. Without loss of generality, S can be selected randomly as long as the seeds in S are disjoint with each other. Lines 3-5 initialize the tuning parameter λ_0 for each seed. Lines 7-15 give one step expansion for each subgraph. We expand all the subgraphs together, which can achieve a balance for the size of each subgraph. Lines 16-22 check if two subgraphs should be combined. In the expansion step 8, the order of the unvisited nodes that will be added may influence the structure of the subgraphs. For example, given a subgraph with 10 nodes, assume that there exist two new nodes that each of the two connects nine nodes in the subgraph. Then if we select $\epsilon_f = 0.9$, the first considered new node will always be incorporated into the subgraph while the second one will be rejected. In traffic analysis, the order of the nodes can be obtained according to some domain knowledge, e.g. the spatial locations of the nodes. If no prior information is available, the order of the nodes can be arranged randomly.

Complexity analysis. Assume that the final average size of the subgraphs is R and denote the time of the solver to solve the APLE step 11 as $L(R)$, lines 7-15 can be computed in $O(\mathcal{K} \cdot L(R))$ time. Lines 16-22 take at most $O(\mathcal{K}^2 p)$ time with auxiliary $O(p)$ space. The number of iteration of line 6 reaches p at most. Thus our CGSE algorithm takes $O(\mathcal{K}^2 p^2 + \mathcal{K} p \cdot L(R))$ time in the worst case. In this paper, $L(R)$ is the complexity of the ℓ_1 regularization solver in [24], which is logarithmic complexity with R [24].

Algorithm 1. CGSE Algorithm

Input: (1) p random variables $\boldsymbol{X} = \{x_1, \ldots, x_p\}$ where x_i contains n observations; (2) $\mathcal{K}$ initial seeds $S = \{S_1, \ldots, S_\mathcal{K}\}$ where $|S_1| = \cdots = |S_\mathcal{K}|$;
Parameters: (1) fitness threshold ϵ_f; (2) combination threshold ϵ_0; (3) initial tuning parameter λ_0;
Output: Dependence relationship among p variables and the overlapping subgraphs;

```
1:  g = S;
2:  K = 𝒦;
3:  for i = 1 to K do
4:      λ_i = λ_0;
5:  end for
6:  repeat
7:      for i = 1 to K do
8:          Find an unvisited variable x_j from the nodes that are
        not in subgraph g_i;
9:          k = |g_i|;
10:         λ_i = sqrt(log_k(k+1)) λ_i;
11:         θ_new = APLE(g_i, x_j, λ_i);
12:         if Fitness(θ_new) ≥ ε_f then
13:             Add x_j into g_i;
14:         end if
15:     end for
16:     for each g_i in g do
17:         for each g_j in g (j ≠ i) do
18:             if |g_i ∩ g_j|/|g_i| > ε_o or |g_i ∩ g_j|/|g_j| > ε_o then
19:                 Combine g_j into g_i;
20:             end if
21:         end for
22:     end for
23:     K = |g|;
24: until Each subgraph stays unchangeable
25: Output g;
```

5 Deal with Dynamic Graphs

In this section, we extend the proposed overlapping decomposition technique to deal with sequence of graphs, where

the dependence relationships among the random variables in the graph change over time. A number of works have been proposed to learn dynamic structures of the graphs over time [26], [2], [27], [28], [29], [30], however, none of them considers to learn the dynamic structures and decomposes the graph into overlapping subgraphs simultaneously as we do. In this paper, we focus on comparing the extended dynamic overlapping decomposition technique with our static counterpart. The comparison between the proposed dynamic method and the previous works concerned with dynamic graphs will be considered in our further work.

Assume the graph evolves over time $t = 1, \ldots, T$, then at any time t, the target of a single graphical model is to estimate $\Omega(t)$ by minimizing the corresponding penalized negative log-likelihood criterion [26], [31]:

$$\widehat{\Omega}(t) = \arg\min_{\Omega \succeq 0} \quad tr(\widehat{\Sigma}(t)\Omega) - \log|\Omega| + \lambda \sum_{i \neq j} |\theta_{ij}|, \tag{13}$$

where $\widehat{\Sigma}(t)$ is the kernel estimator of the sample covariance at time t estimated from input $X(t)$:

$$\widehat{\Sigma}(t) = \frac{\sum_{t'} w_{t't} \boldsymbol{X}(t') \boldsymbol{X}(t')^T}{\sum_{t'} w_{t't}}, \tag{14}$$

which is an averaged weighted covariance matrix. The weights $w_{t't}$ is defined as $w_{t't} = K(\frac{|t'-t|}{h})$, where t' takes the value from some adjacent time points of time t, $K(\cdot)$ is a symmetric nonnegative kernel function, and h is a bandwidth parameter that controls the smoothness over time of the estimated covariance matrix. Here, we use the Gaussian kernel $K(x) = \exp(-x^2)$.

Algorithm 2. Dynamic Overlapping Decomposition Procedure

Input: Random variables $\{\boldsymbol{X}(1), \ldots, \boldsymbol{X}(T)\}$ over time $1, \ldots, T$;
Parameters: Bandwidth parameter h;
Output: Dependence relationships among the variables and the overlapping subgraphs for each graph at time $t = 1, \ldots, T$;

```
for t = 1 to T do
    Calculate Σ̂(t) according to Eq. (14);
    Learn the graph at time t by CGSE Algorithm;
end for
Output the subgraphs at time t = 1, ..., T;
```

Based on the overlapping decomposition technique introduced previously, we propose to estimate $\Omega(t)^{(k+1)}$ by solving the following problem:

$$\begin{aligned} &\min_{(\boldsymbol{\theta}(t), \theta_{k+1}(t))} 2\widehat{\varepsilon}(t)^T \boldsymbol{\theta}(t) + \widehat{\varepsilon}_{k+1}(t)\theta_{k+1}(t) \\ &- \log\big(\theta_{k+1}(t) - \boldsymbol{\theta}(t)^T \cdot \widehat{\Sigma}^{(k)}(t) \cdot \boldsymbol{\theta}(t)\big) + 2\lambda_\theta \|\boldsymbol{\theta}(t)\|_1, \end{aligned} \tag{15}$$

where $(\boldsymbol{\theta}(t), \theta_{k+1}(t))$ is defined in Eq. (2) according to $\Omega(t)^{(k+1)}$, and $(\widehat{\varepsilon}(t), \widehat{\varepsilon}_{k+1}(t))$ is the kernel estimator of the sample covariance vector between $x_{k+1}(t)$ and $\{x_1(t), \ldots, x_{k+1}(t)\}$, which can be obtained directly from Eq. (14). Then the dynamic overlapping decomposition procedure is stated in Algorithm 2. The time complexity of Algorithm 2 is T times of that of the CGSE algorithm for one static graph.

6 Experimental Study

We evaluate the proposed Overlapping Decomposition method for Gaussian Graphical Model (ODGM). We compare with the Single Graphical Model, which is solved by the graphical lasso [9]. To further study the advantage of the overlapping decomposition, we adapt the proposed CGSE algorithm to support the Non-Overlapping Decomposition for Gaussian Graphical Model (NODGM) by setting $\gamma = +\infty$ in Eq. (10) to forbid overlaps.

We report results on synthetic datasets in Section 6.1. In Section 6.2, we report the performance study on real-life traffic dataset, and show the usefulness of the results for traffic analysis.

6.1 Synthetic Data

6.1.1 Setting

Since we focus on graphical models of a relatively large scale, we generate a set of networks whose number of nodes, p, ranges from 100 to 900. Note that previous correlation based analytical models normally use networks with tens of nodes. We set the number of observations $n = 800$ for all the settings. We follow the approach [20] to generate the synthetic data. To simulate the heterogeneity in large networks, we generate local centered network by K local Erdös-Rényi random graphs $\{g_1, g_2, \ldots, g_K\}$, $g_i = (SV_i, SE_i)$, and for homogeneity, we add edges between any g_i and g_j randomly. Specifically, we generate the data as follows.

1) We generate K Erdös-Rényi graphs, each with a random size in $[20, 80]$, such that $\sum_i^K |SV_i| = p$. Let E_{cross} be the set of cross links between the K graphs, and let $E_{inner} = \bigcup_i^K SE_i$ be the set of total inner links. Let $\rho = |E_{cross}|/|E_{inner}|$ be a factor to control the homogeneity. We randomly add $\rho|E_{inner}|$ cross edges. Finally, we can get a network $G = (V, E)$, where $V = \bigcup_i^K SV_i$ and $E = E_{inner} \bigcup E_{cross}$.
2) Based on the above network, we create a covariance matrix by following [32]. Define a $p \times p$ matrix A as

$$A_{ij} = \begin{cases} 1, & i = j, \\ U([-1, -0.5] \cup [0.5, 1]), & (i, j) \in E, \\ 0, & else, \end{cases}$$

where $U(\cdot)$ represents uniform distribution. We scale the diagonal elements to ensure positive definiteness and average the matrix with its transpose to get a symmetric A. Then the covariance matrix Σ is calculated as

$$\Sigma_{ij} = (A^{-1})_{ij} / \sqrt{(A^{-1})_{ii}(A^{-1})_{jj}}.$$

3) We generate p-dimensional samples from $\mathcal{N}(0, \Sigma)$.

We define Precision, Recall and F1-score to measure the effectiveness of different models in finding the dependence relationships. Note that the true dependence relationships in E are known in the generated data.

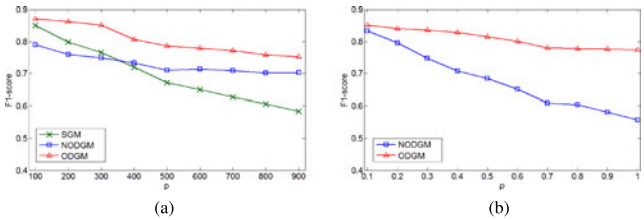


Fig. 1. F1-scores for SGM, NODGM and ODGM when varying p and ρ. (a) Varying p. (b) Varying ρ.

Given an estimated $\widehat{E}$ returned by a method, we define these metrics as follows:

$$Pre = \frac{|\{(i,j) : (i,j) \in E, (i,j) \in \widehat{E}\}|}{|\{(i,j) : (i,j) \in \widehat{E}\}|},$$
$$Rec = \frac{|\{(i,j) : (i,j) \in E, (i,j) \in \widehat{E}\}|}{|\{(i,j) : (i,j) \in E\}|},$$
$$F1 = \frac{2 \cdot Pre \cdot Rec}{Pre + Rec}.$$

Moreover, we fix the fitness threshold $\epsilon_f = 0.1$, combination threshold $\epsilon_o = 0.6$, $\gamma = 0.1$ and $\lambda_0 = 0.2$ in the synthetic data to evaluate our proposed model on different settings where the number of nodes p and the parameter ρ are varying. In reverse, we study and visualize the effect of some of these hyper-parameters on a real-life traffic network in the next section where the number of nodes is given without parameter ρ. We set the number of seeds as $|S| = K$ and each S_i is selected randomly from the K Erdös-Rényi graphs with size $|S_i| = 3$.

6.1.2 Results

Varying p. To evaluate these methods on networks of various sizes, we vary p from 100 to 900. We fix $\rho = 0.3$ in this setting. The performances of all the methods are shown in Fig. 1a. We can see that when p is small, SGM performs as well as ODGM, because a single graphical model can work well. However, as p increases, the accuracy of SGM falls rapidly. As explained in Section 4.2, a relatively large p will lead to worse performance with small n. However, both decomposition methods still work well with the increase of p. ODGM achieves a high accuracy and outperforms NODGM consistently, because non-overlapping decomposition cannot capture the overlap information.

Varying ρ. The parameter ρ plays an important role on controlling the homogeneity of the network. When $\rho = 0$, it means the network is essentially heterogeneous and is actually composed of several separate sub-networks, while a large ρ indicates that the edges in the network tend to distribute homogeneously. Fig. 1b shows the F1-score of ODGM and NODGM while ρ is varying, and in this setting we fix $p = 500$. As expected, when ρ approaches zero, NODGM performs as well as ODGM because the network can be divided completely into sub-networks. But as ρ increases, the disparity between ODGM and NODGM becomes larger since ODGM can discover the overlaps while NODGM losses more information.

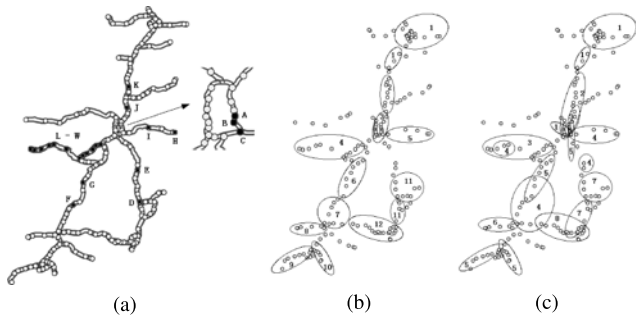


Fig. 2. (a) The real-life traffic network; (b) the non-overlapping decomposition structure learned by NODGM; (c) the overlapping decomposition structure learned by ODGM.

6.2 Traffic Data

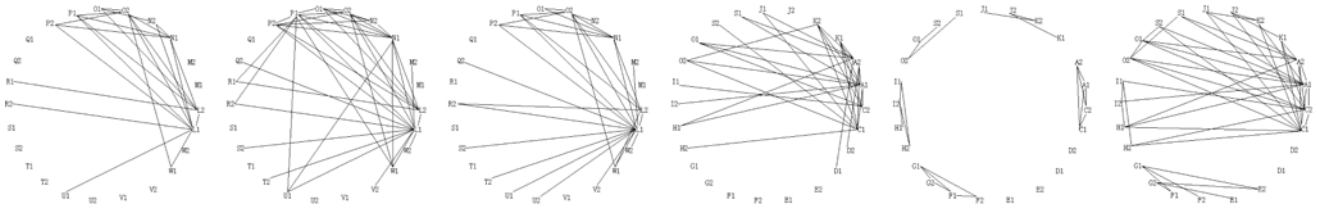
6.2.1 Description and Setting

In this section, we evaluate our methods on real-life traffic data. The features in this traffic dataset are observations collected from sensors located on ramps in a highway traffic network. Each observation is the vehicle count during a time interval. Fig. 2a shows the structure of the highway traffic network from a province in China, in which each circle represents a traffic station consisting of an on-ramp and an off-ramp, and the line between any connected traffic stations is the bidirectional highway. There is an important ring in the network which is amplified on the right hand side of Fig. 2a—the city in the center of this ring is a big city and plays a central role in the entire traffic network. We study both the static and dynamic traffic network in this dataset.

There are total 180 traffic stations (circles), which correspond to 360 ramps, i.e., $p = 360$. We first study the static network, where the observations are collected at time interval 9:00-9:15 from 2011/1/1 to 2011/2/28 (59 days). Therefore, $n = 59$ for each feature. Due to the stability and periodicity of traffic behaviors, the observations collected from the same time duration in each day (e.g. the considered interval 9:00-9:15 here) are assumed to follow a Gaussian distribution.

In addition to the static setting above, we also study the daily evolvement of this network, where each day is divided into 96 time intervals: 0:00-0:15, 0:15-0:30, . . ., 23:45-24:00. We assume that the dependence relationships among variables during each time interval is changeless.

If no specific settings are declared, we use $\epsilon_f = 0.1$, $\epsilon_o = 0.6$, $\gamma = 0.1$, $\lambda_0 = 50$, and set the number of seeds $|S| = 12$ with each $|S_i| = 6$ as default in both static and dynamic studies. We will discuss how to select some of the hyper-parameters in specific learning tasks later. Since there is no ground truth for the precision matrix in real-life traffic data, F1-score cannot be measured. Moreover, since the NODGM and ODGM methods utilize an incremental strategy by solving the problem (4) or problem (9) at each step, it does not optimize the value of the negative log-likelihood defined in Eq. (1) directly. Therefore comparing the value of the negative log-likelihood may not be a good criterion to evaluate the model performance. Nevertheless, we can utilize the discovered dependence relationships to construct predictive models for predicting the traffic flows, and evaluate different models in view of the prediction performance.



(a) SGM (b) NODGM (c) ODGM (d) SGM (e) NODGM (f) ODGM

Fig. 3. Detailed dependence relationships among the selected features: (a), (b) and (c) are correlations discovered by SGM, NODGM and ODGM among the local concentrated features respectively; (d), (e) and (f) are correlations discovered by SGM, NODGM and ODGM among the scattered features respectively.

Moreover, the dependence relationships detected are the most important information for traffic analysis, and our domain experts can help with their knowledge on the interactive relationships in the traffic network, which can also provide a measurement for different models.

6.2.2 Results and Analysis in Static Setting

Figs. 2b and 2c give the subgraph structures returned by NODGM and ODGM, respectively. For clear representation, we draw the results based on the initial traffic network with 180 traffic stations instead of 360 features, and a subgraph contains a traffic station node iff at least one feature (ramp) of this traffic station belongs to it. In the figures, the ellipses with the same label denote an indexed subgraph. Since non-overlapping subgraphs have no intersections, the subgraphs cannot be combined together, and thus the number of final subgraphs equals to the number of seeds in Fig. 2b. For the overlapping structure, when two subgraphs overlap at a certain threshold ϵ_o, they are combined together. Thus, we end up with eight subgraphs in Fig. 2c.

From the two figures, we can observe: (1) both NODGM and ODGM show that the dependence relations between the vehicle flows follow the spatial distribution in general—the nearer two features locate spatially in the traffic network, the more correlated they tend to be; (2) ODGM highlights some crucial traffic nodes that are highly overlapped, such as the nodes on the central ring. As mentioned earlier, the central ring is around the central city and plays an important role in the entire traffic network. Additionally, traffic station C on the ring is the passageway connecting the unique airport of the entire network, and traffic stations A and B are the top 2 traffic stations with the highest vehicle flows measured on both on-ramp and off-ramp. These domain information matches well with our ODGM result and gives an reasonable explanation; (3) ODGM is able to detect long distance dependence relations in addition to the local relations within distances. For example, the components (ellipses) of subgraph 1, 3, 4 and 5 are distributed spatially, but they are highly correlated within the vehicle flows. In other words, there also exists long distance origin-destination demand in the traffic network. However, NODGM cannot mine such information described in both conclusions (2) and (3); (4) some of the sparsely located traffic stations are not included in any subgraphs in both the figures. We find that the vehicle flows in most of these traffic station ramps are nearly 0 during the observation periods, and almost 80 percent of the ramps and their located highways are newly built. Thus they are seldom used and have no interactive relationships with other ramps.

Fig. 3 gives the detailed dependence relationships among a set of selected features. For each selected traffic station i, $i1$ and $i2$ denote the on-ramp and off-ramp features, respectively. In Figs. 3a, 3b and 3c, the features are selected from traffic stations L-W in Fig. 2a, and these traffic stations are selected locally concentrated. We can see that SGM detects fewer correlation information than do NODGM and ODGM, because a single graphical model treats the entire network globally, and can only interpret the correlations from a global view. In this setting, NODGM detects more dependence relationships than do ODGM, which also claims that NODGM focuses more on a local view while ODGM is a compromise of SGM and NODGM. Figs. 3d, 3e and 3f provide the detailed dependence relationships among the features selected from A, C-K, O and S, which are scattered in the network. From the results, ODGM discovers more meaningful correlation information than do SGM and NODGM, e.g., the relations among $\{E1, E2, F1, F2, G1, G2\}$. Both ODGM and SGM are able to discover the important long distance relations for the important traffic stations A and C. However, NODGM is restricted by its non-overlapping structure and only detects the inner relationship within subgraphs, even if A and C are highly correlated with others.

These results obtained by ODGM are important for the analysis of traffic systems. First, the traffic stations in the same subgraph are highly correlated and should be considered together by traffic systems. For example, it is possible that vehicle flows rush into each other within the same subgraph. Second, the dependence relationships are very helpful for traffic flow prediction and anomaly detection which are hot concerns of traffic operators and managers. Third, it is important to find the highly overlapped traffic stations. These crucial traffic stations are correlated with a number of regions, based on which the regions with heavy traffic can be detected. On the other hand, the regions with light traffic can also be reflected by independent traffic stations. These information can be used by highway construction planners to design new roads. To be more specific, in the next section we will show how to import these discovered dependence relationships in learning specific traffic tasks, and we provide a measurement for the discovered dependence relationships obtained from different models to evaluate their performance.

6.2.3 Application in Traffic Flow Prediction

We utilize the discovered dependence relationships obtained from different models to predict the traffic flows. Specifically, we use the learned overlapping subgraphs to

TABLE 1
Prediction Performance Based on Different Dependence Relationships Obtained from the NODGM and ODGM Models

	STL	NODGM	ODGM
TMSE (%)	23.94	21.85	**19.46**

construct multiple multi-task learning (MTL) models [33], since the strong interactions existed in the subgraph can be viewed as information sharing in the MTL paradigm. The nodes corresponding to the off-ramps are viewed as the tasks that need to be predicted, the nodes corresponding to all the on-ramps are treated as features, and the tasks in the same subgraph are viewed as related tasks that share some common information. The tasks that do not belong to any subgraphs are predicted via single task learning (STL) models. Denoting by $\mathbf{y}$ and $\mathbf{X}$ the responses of the tasks and the features respectively, we consider a linear model $\mathbf{y} = \mathbf{XW} + \boldsymbol{\delta}$ ($\boldsymbol{\delta}$ is the noise vector) as the predictive model for both the MTL and STL settings. We use the MTL model described in [33].[1] We define the Total Mean Square Error (TMSE) as $\text{TMSE} = \frac{\sum_i |y_i^* - \hat{y}_i|}{\sum_i y_i^*} \times 100\%$, where y_i^* is the true value of the traffic flow of the ith off-ramp, and $\hat{y}_i$ is the corresponding predicted value. We use 49 samples for training and the rest for testing. Table 1 shows the prediction performance based on the dependence relationships obtained from NODGM and ODGM. We also provide the prediction performance of the STL model for all the off-ramps as the baselines. From the results, we see that the dependence relationships obtained from both the NODGM and ODGM models are helpful for improving the prediction accuracy compared with STL, and the dependence relationships obtained from the ODGM models are more useful than that obtained from the NODGM model, which provides a measurement for the performance of the NODGM and ODGM methods in the view of traffic flow prediction.

6.2.4 Varying Hyper-Parameters in Static Setting

We study the effect of the hyper-parameters ϵ_f, ϵ_o and γ for CGSE. Fig. 4 shows some information about the decomposition results obtained from ODGM when these hyper-parameters are varying. When we vary each hyper-parameter, we use the aforementioned default values for the other hyper-parameters.

Parameter ϵ_f controls the minimum fitness, and restricts the size of each subgraph. As shown in Fig. 4a, when ϵ_f decreases, more features are added into subgraphs and the size of each subgraph becomes larger. Fig. 4b shows the effect of ϵ_o. When ϵ_o is reduced, the subgraphs are more likely to be combined together under ϵ_o. Fig. 4c shows the relationship between γ, which controls overlaps, and the number of features with different overlap degrees. We observe that when γ increases, fewer overlaps exist in the decomposition structure, and so does the number of the overlapped features.

Fig. 5 visualizes the generated subgraphs for selected hyper-parameter values to show the details. From these figures, we can see that the property of each hyper-parameter is in line with the results in Fig. 4, and these figures give a more intuitive and understandable description for our method.

Moreover, we also explorer how the obtained subgraphs from different settings of the hyper-parameters influence the prediction performance when we use these results to predict traffic flows as we do previously. Fig. 6 shows the prediction performance in terms of TMSE when we apply different hyper-parameters ϵ_f, ϵ_o and γ for ODGM and utilize the corresponding results to construct MTL models. From Fig. 6, we observe that an appropriate setting (e.g. the default setting) for the hyper-parameters will obtain better prediction performance. This implies that when we apply the obtained subgraphs as well as the dependence relationships to specific learning tasks, we can use a grid search method to select these hyper-parameters from some candidate sets.

6.2.5 Results and Analysis in Dynamic Setting

In this experiment, we learn the graph structures at the 96 different time intervals during one day, and report the results for the NODGM and ODGM methods in Fig. 7 by selecting the graphs from four time intervals: 3:00-3:15, 9:00-9:15, 15:00-15:15, 21:00-21:15. The settings keep default except that for the time interval 3:00-3:15, the number of seeds is set as $|S| = 6$ with each $|S_i| = 4$. The reason will be explained in the following analysis.

Figs. 7a, 7b, 7c, and 7d are learned by the NODGM method, while Figs. 7e, 7f, 7g, and 7h are obtained by the ODGM method. We can observe that: (1) the subgraphs learned at different time intervals from both NODGM and ODGM methods show different structures, implying that the dependence relationship among the traffic network evolves over time; (2) both the number and the size of the subgraphs are small, since there is merely dependence relationship existed in the graph at the time interval 3:00-3:15. Actually, during the time interval 3:00-3:15 in each day, i.e. the late night, there are very few vehicles traveling on this highway network (most of the values are zeros), and it is even hard to find some initial small seeds such that some correlations exist there. This explains the results in Fig. 7e and why we use a different setting with smaller number and size of the initial seeds for this time interval; (3) NODGM still fails to detect long distance relations and maintain the subgraphs independent, while ODGM is effective to discover long distance dependence relationships. For example, at time interval 3:00-3:15 in Fig. 7a again, the result obtained from NODGM is almost the same with its initial six seeds (subgraph-3 and subgraph-4 are exactly the initial seeds, where one cycle contain two nodes, an on-ramp and an off-ramp), and no meaningful insights can be obtained. However, ODGM combines the subgraph-2, subgraph-3 and subgraph-4 in Fig. 7a into one subgraph-2 in Fig. 7e, which implies that there exist long distance dependence relationships among these distant ramps. As being made aware by our domain experts, during the late night of each day such as 3:00-3:15, lots of trunks loading with mineral travel from east to west or west to east across this highway network,

1. http://www.cs.ucl.ac.uk/staff/A.Argyriou/code/

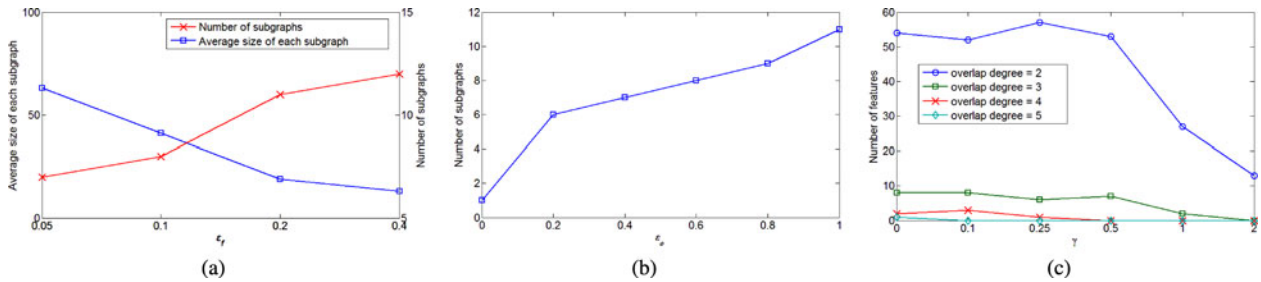


Fig. 4. (a) The relationship among the average size of the subgraphs, the number of subgraphs and ϵ_f; (b) the relationship between the number of subgraphs and ϵ_o; (c) the relationship between the overlap degree and γ.

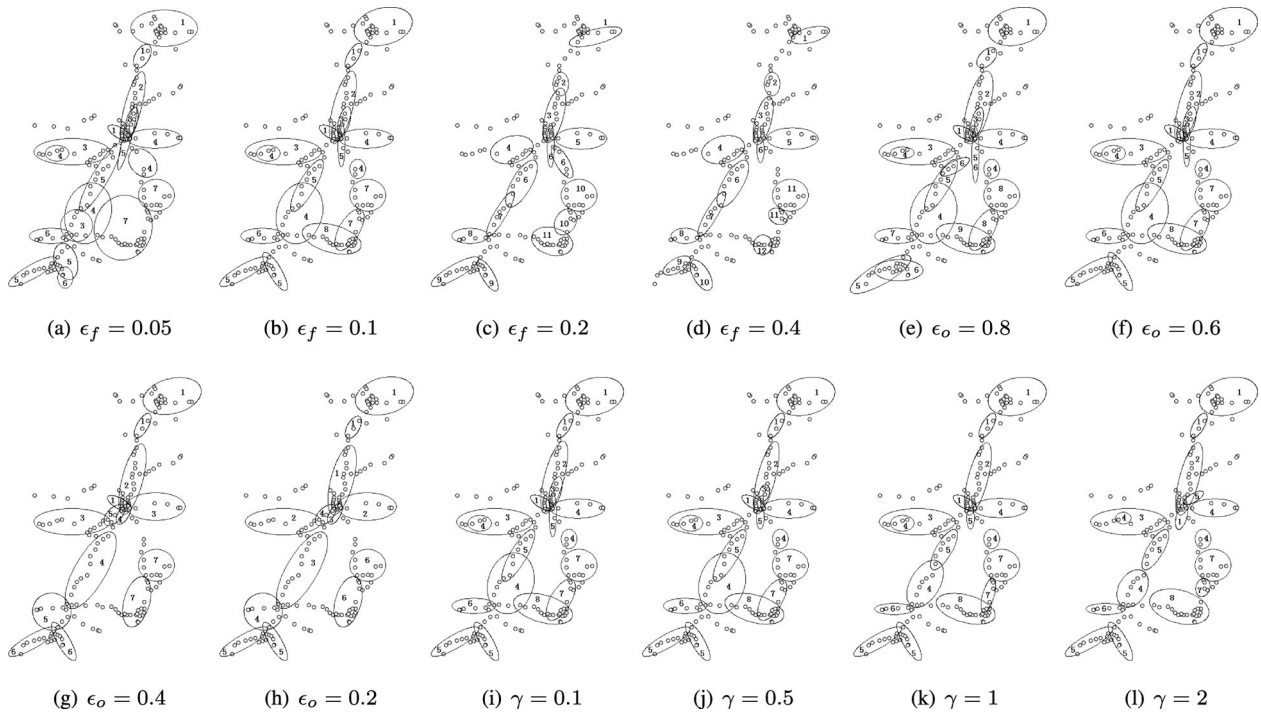


Fig. 5. The obtained decomposition structures by ODGM when varying different hyper-parameters. Each hyper-parameter varies with other hyper-parameters fixed. Default setting: $\epsilon_f = 0.1$, $\epsilon_o = 0.6$ and $\gamma = 0.1$.

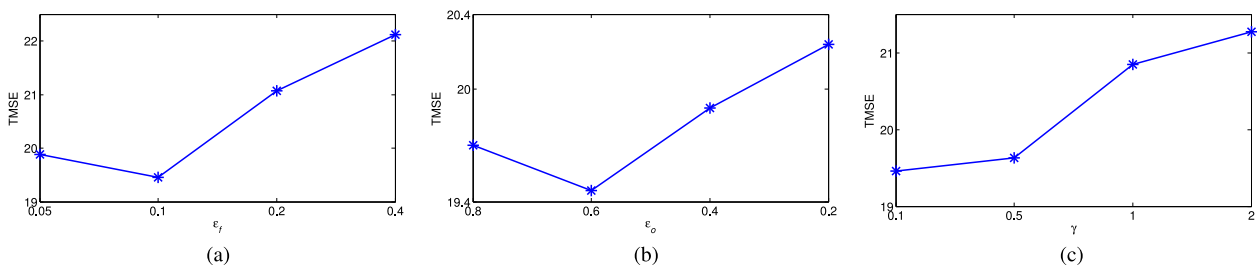


Fig. 6. The prediction performance in terms of TMSE (percent) when varying different hyper-parameters, where the obtained dependence relationships are applied to the MTL predictive models.

which leads to frequent connections between the subgraph-3 and subgraph-4 in Fig. 7a. This is just captured by the subgraph-2 in Fig. 7e. Such domain information verifies the results learned by the ODGM method reasonably; (4) the overlapping parts in different time intervals are quite different. It has been discussed that the overlaps highlighted by ODGM in the traffic network play crucial roles. Therefore, the evolvement of the overlaps indicates the evolvement of the busy areas or the locations with heavy traffic. Such information is essentially important for learning the dynamics of the traffic behaviors and is helpful for traffic predication.

7 Conclusion and Future Work

In this paper, we proposed an overlapping decomposition technique for the Gaussian graphical model of a large scale. The technique utilizes an additive expanding property and reduce the problem of solving a $(k+1)$-node Gaussian graphical model to the problem of solving a one-step vector

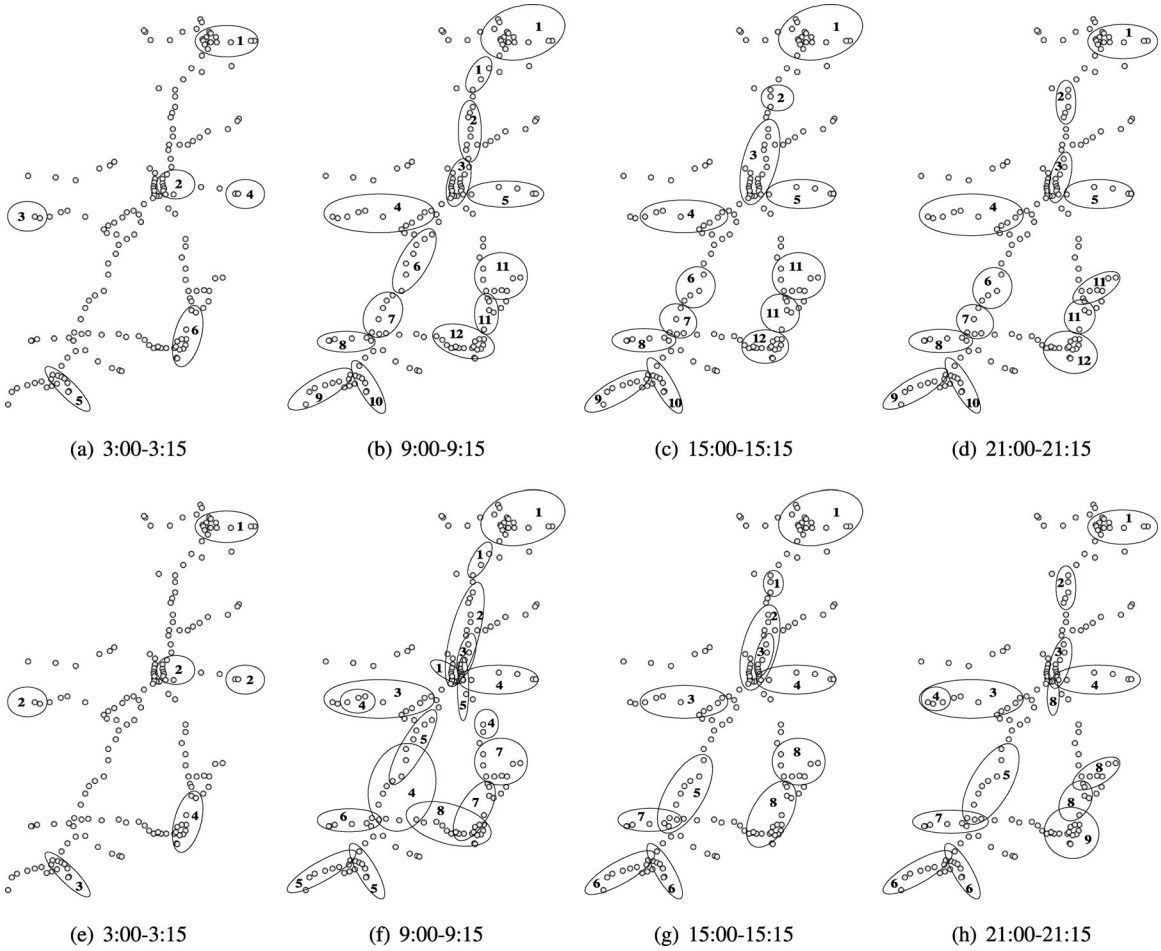


Fig. 7. Decomposition structures learned by NODGM and ODGM at different time intervals. (a)-(d) Structures learned by NODGM. (e)-(h) Structures learned by ODGM.

regularization problem based on a solved k-node Gaussian graphical model. Detailed asymptotic analysis of this technique was discussed. Based on the additive expanding property, we developed a constraint greedy subgraph expansion algorithm to generate overlapped subgraphs. We demonstrated on both synthetic data and real-life traffic data that the overlapping decomposition method is more powerful than the single graphical model and its non-overlapping decomposition counterpart. Moreover, in the application of the traffic data analysis, we study both the static and dynamic cases, and the results show that our models can provide rich information for traffic analysis.

In the current paper, we focus on the Gaussian graphical model, which deals with undirected graph structures. As one of the future direction, it is interesting to apply the overlapping decomposition technique to deal with directed graphical models with other formulations. In highway systems, vehicles travel through the traffic network with time costs, therefore, some time lags exist among the dependencies of different ramps. As another future direction, we are interested in studying the decomposition problem by considering lag intervals for temporal dependency analysis.

Acknowledgments

The authors thank the anonymous reviewers for their valuable comments. They also thank Prof. Yu Zhang for the valuable discussions. This work was supported by the National High Technology Research and Development Program of China (2014AA015103), Beijing Natural Science Foundation (4152023), the National Natural Science Foundation of China under Grant No. 61473006, and the National Science and Technology Support Plan (2014BAG01B02). Guojie Song and Lei Han contributed equally. Lei Han is the corresponding author.

References

[1] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[2] X. Chen, Y. Liu, H. Liu, and J. G. Carbonell, "Learning spatial-temporal varying graphs with applications to climate data analysis," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 425–430.

[3] N. Friedman, "Inferring cellular networks using probabilistic graphical models," *Science*, vol. 303, no. 5659, pp. 799–805, 2004.

[4] Y. Liu, A. Niculescu-Mizil, A. Lozano, and Y. Lu, "Temporal graphical models for cross-species gene regulatory network discovery," *J. Bioinform. Comput. Biol.*, vol. 9, no. 02, pp. 231–250, 2011.

[5] Y. Liu, J. R. Kalagnanam, and O. Johnsen, "Learning dynamic temporal graphs for oil-production equipment monitoring system," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 1225–1234.

[6] A. C. Lozano, H. Li, A. Niculescu-Mizil, Y. Liu, C. Perlich, J. Hosking, and N. Abe, "Spatial-temporal causal modeling for climate change attribution," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 587–596.

[7] N. Ruan, R. Jin, V. E. Lee, and K. Huang, "A sparsification approach for temporal graphical model decomposition," in *Proc. 9th IEEE Int. Conf. Data Mining*, 2009, pp. 447–456.
[8] A. Arnold, Y. Liu, and N. Abe, "Temporal causal modeling with graphical granger methods," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 66–75.
[9] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.
[10] M. Yuan and Y. Lin, "Model selection and estimation in the gaussian graphical model," *Biometrika*, vol. 94, no. 1, pp. 19–35, 2007.
[11] L. Han, G. Song, G. Cong, and K. Xie, "Overlapping decomposition for causal graphical modeling," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 114–122.
[12] R. Mazumder and T. Hastie, "Exact covariance thresholding into connected components for large-scale graphical lasso," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 781–794, 2012.
[13] D. A. Spielman and S.-H. Teng, "A local clustering algorithm for massive graphs and its application to nearly-linear time graph partitioning," *arXiv preprint arXiv: 0809.3232*, 2008.
[14] C.-J. Hsieh, A. Banerjee, I. S. Dhillon, and P. K. Ravikumar, "A divide-and-conquer method for sparse inverse covariance estimation," in *Proc. Adv. Neural Inform. Process. Syst.*, 2012, pp. 2330–2338.
[15] C.-J. Hsieh, M. A. Sustik, I. Dhillon, P. Ravikumar, and R. Poldrack, "BIG & QUIC: Sparse inverse covariance estimation for a million variables," in *Proc. Adv. Neural Inform. Process. Syst.*, 2013, pp. 3165–3173.
[16] C.-J. Hsieh, I. S. Dhillon, P. K. Ravikumar, and M. A. Sustik, "Sparse inverse covariance matrix estimation using quadratic approximation," in *Proc. Adv. Neural Inform. Process. Syst.*, 2011, pp. 2330–2338.
[17] F. Oztoprak, J. Nocedal, S. Rennie, and P. A. Olsen, "Newton-like methods for sparse inverse covariance estimation," in *Proc. Adv. Neural Inform. Process. Syst.*, 2012, pp. 755–763.
[18] E. Treister and J. S. Turek, "A block-coordinate descent approach for large-scale sparse inverse covariance estimation," in *Proc. Adv. Neural Inform. Process. Syst.*, 2014, pp. 927–935.
[19] P. Danaher, P. Wang, and D. M. Witten, "The joint graphical lasso for inverse covariance estimation across multiple classes," *J. Roy. Statistical Soc.: Series B (Statistical Methodol.)*, vol. 76, no. 2, pp. 373–397, 2014.
[20] J. Guo, E. Levina, G. Michailidis, and J. Zhu, "Joint estimation of multiple graphical models," *Biometrika*, vol. 98, pp. 1–15, 2011.
[21] C. Lee, F. Reid, A. McDaid, and N. Hurley, "Detecting highly overlapping community structure by greedy clique expansion," in *Proc. 4th Int. Workshop Soc. Netw. Mining Anal.*, 2010, pp. 33–42.
[22] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*. New York, NY, USA: Wiley, 2009.
[23] S. L. Lauritzen, *Graphical Models*. London, U.K: Oxford Univ. Press, 1996.
[24] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for L1 regularization: A comparative study and two new approaches," in *Proc. 18th Eur. Conf. Mach. Learn.*, 2007, pp. 286–297.
[25] A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu, "Sparse permutation invariant covariance estimation," *Electron. J. Statist.*, vol. 2, pp. 494–515, 2008.
[26] M. Kolar and E. P. Xing, "On time varying undirected graphs," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2011, pp. 407–415.
[27] M. Kolar, L. Song, and E. P. Xing, "Sparsistent learning of varying-coefficient models with structural changes," in *Proc. Adv. Neural Inform. Process. Syst.*, 2009, pp. 1006–1014.
[28] L. Song, M. Kolar, and E. P. Xing, "Time-varying dynamic bayesian networks," in *Proc. Adv. Neural Inform. Process. Syst.*, 2009, pp. 1732–1740.
[29] M. Kolar, L. Song, A. Ahmed, and E. P. Xing, "Estimating time-varying networks," *Ann. Appl. Statist.*, vol. 4, no. 1, pp. 94–123, 2010.
[30] J. Honorio and D. Samaras, "Multi-task learning of gaussian graphical models," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 447–454.
[31] S. Zhou, J. Lafferty, and L. Wasserman, "Time varying undirected graphs," *Mach. Learn.*, vol. 80, no. 2/3, pp. 295–319, 2010.
[32] J. Peng, P. Wang, N. Zhou, and J. Zhu, "Partial correlation estimation by joint sparse regression models," *J. Amer. Statist. Assoc.*, vol. 104, no. 486, pp. 735–746, 2009.
[33] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Mach. Learn.*, vol. 73, no. 3, pp. 243–272, 2008.

Guojie Song received the BS and MS degrees from Zhengzhou University, Zhengzhou, China, in 1998 and 2001, respectively, and the PhD degree from Peking University, Beijing, China, in 2004. From 2004 to 2005, he was a research fellow with the Singapore Management University, Singapore. He is currently an associate professor with the School of Electronic Engineering and Computing Science and the vice director in the Research Center of Intelligent Information Processing, Peking University. His research interests include various techniques of data mining, machine learning and their applications in intelligent transportation systems, and social networks.

Lei Han received the PhD degree from the School of Electronics Engineering and Computer Science (EECS), Peking University, in 2014. Currently, he is a postdoctoral researcher in the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His current research interests include artificial intelligence, machine learning, pattern recognition, and data mining. He is especially interested in multitask learning, transfer learning, graphical modeling, sparse learning, convex optimization, and dimensionality reduction.

Kunqing Xie received the BS degree from Shanxi Normal University in 1982 and the MS degree from Peking University in 1987. He received the PhD degree from Beijing Normal University in 1998. He is a professor with the School of Electronic Engineering and Computer Science, where he is the dean of the Department of Intelligent Science and the Director of the Research Center of Intelligent Traffic System, Peking University, Beijing, China. He has presided over several research programs in the provincial, ministerial, and national levels, and in international cooperation. He is the author or coauthor of more than 80 papers. His research interests include spatial databases and data warehouses, spatiotemporal information analysis and data mining, remote sensing and geographic information systems, and intelligent traffic systems. He received several provincial-level and ministerial-level awards in research and teaching.