GRAPH MODEL SELECTION IN GAUSSIAN GRAPHICAL MODELS

Thesis submitted by

Anshul Yadav

2017EE10565

under the guidance of

Prof. Sandeep Kumar

in partial fulfilment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY DELHI

JANUARY 2021

THESIS CERTIFICATE

This is to certify that the thesis titled **Graph Model Selection in Gaussian Graphical Models**, submitted by **Anshul Yadav**, to the Indian Institute of Technology Delhi, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: New Delhi
Date: 08-01-2021

Prof. Sandeep Kumar
Assistant Professor
Dept. of Electrical Engineering
IIT Delhi, 110016

ACKNOWLEDGEMENTS

I express my deepest gratitude to my advisor Prof. Sandeep Kumar who helped me in shaping up the entire project and provided constant encouragement, guidance and motivation at every step. I learned a lot from him during the process and greatly benefitted from his immense knowledge and insights.

I express my heartfelt gratitude to my Professors who always encouraged free flow of thoughts and inspired me to pursue Machine Learning as a career. I would specially like to thank Prof. Prathosh A.P. for introducing me to the field in such an elegant manner.

I would also like to thank all my friends who acted as a constant support of motivation and learning during my four years at IIT Delhi. We together had lots of fun and unforgettable memories. I would especially like to mention my wingmates Adarsh Shrivastava, Aman Tiwari, Anchit Tandon, Anurag Yadav, Ashwil Bhupesh, Ayush Jain, Bhargav Varshney, Mayank Bulkunde, Neel Patel and Nihar Patel with whom I have developed an everlasting bond.

Lastly, I am greatly indebted to my parents and my brother Aakash who encouraged me to pursue my dreams throughout my life and supported me through every thick and thin. This journey wouldn't have been possible without them.

ABSTRACT

KEYWORDS: Graph structure learning; Graph Model Selection; Gaussian Graphical Models, eBIC.

Learning the graph from data holds immense significance in graph-based applications. One of the major limitations of the existing graph learning methods is that they require some form of a priori information about the underlying structure present in the data, which may not be available in a variety of applications e.g., network medicine, gene regulatory networks. In such cases, structure identifiability from data becomes very crucial to create an end-to-end model. Our work proposes a novel method for graph model selection in gaussian graphical models. We solve this problem by integrating model selection and degrees of freedom with the structured graph learning algorithm (Kumar *et al.*, 2019). We conduct extensive experiments on synthetic and real datasets to demonstrate the effectiveness of our algorithm. All the codes and experiments are available at: https://github.com/anshul3899/Structured-Graph-Learning.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

GGMs	Gaussian Graphical Models
DOF	Degrees of Freedom
SGL	Structured Graph Algorithm
BIC	Bayesian Information Criterion
eBIC	extended Bayesian Information Criterion
GNNs	Graph Neural Networks

NOTATION

n	Number of observations/samples corresponding to each node of the graph
p	Number of nodes in the graph
$\mathcal{X}$	Data, $\in \mathbb{R}^{n \times p}$ (assumed to be coming from a gaussian graphical model)
Θ	Graph laplacian matrix, $\in \mathbb{R}^{p \times p}$
$\mathcal{L}$	Graph laplacian Operator

CHAPTER 1

INTRODUCTION

1.1 Motivation

We want to learn graph from data to encode relationships between entities. These entities and relationships can be represented using nodes and edges of the graph. The edge weight corresponds to the strength of connection between two nodes. The learned graphs have numerous applications in Graph Neural Networks (GNNs), graph signal processing, classification, prediction, smoothing, structural inference etc. With the advent of GNNs, learning graphs from data holds tremendous potential with applications such as node classification, pairwise node classification etc. Fig 1.1 shows the time series data samples observed from different parts in the brain. These signals can represent the neural activity of a part of the brain over time. If we can learn graph from data, we can capture the useful relationships between these nodes.

(a) (b)

Figure 1.1: Learning graph from brain data signals. Image src: Dong *et al.* (2019)

Our goal is to learn a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V} = 1, 2, ..., p$ is the node set consisting of p nodes corresponding to the entities and $\mathcal{E} = (1,2), (1,3), ...$ is the edge set with $|\mathcal{E}|$ number of edges representing the relationships between these

entities. The weight matrix W represents the edge weights denoting the strength of these relationships. For the sake of simplicity, we restrict ourselves to a class of Probabilistic Graphical Models (PGMs) called the Gaussian Graphical Models (GGMs). The assumption of gaussianity is not too restricting in the sense that if we have large number of samples, the distribution converges into gaussian distribution due to the central limit theorem. Fig 1.2 pictorially depicts our goal of learning graph from $\mathbb{R}^{n \times p}$ data.

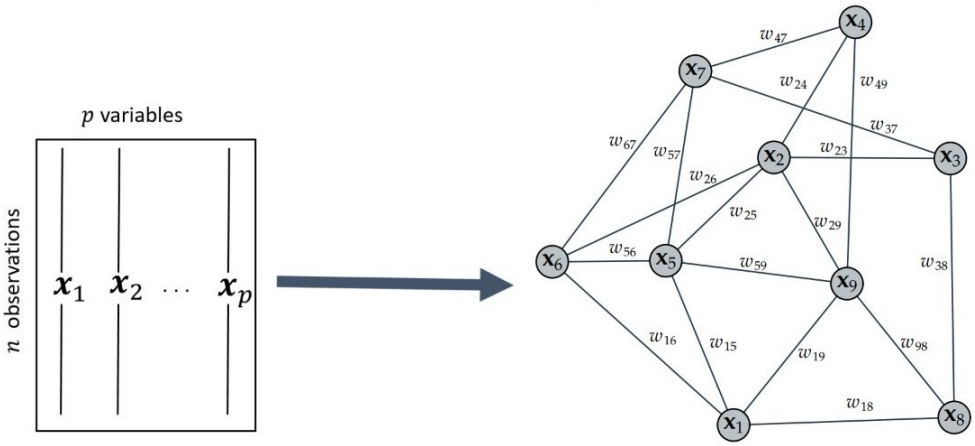


Figure 1.2: Learning graph structure from data

1.2 Gaussian Graphical Models

Gaussian Graphical Models are essentially multivariate gaussian distributions. Consider the following density of multivariate gaussian distribution:

$$p(\mathbf{x} \mid \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)\right) \tag{1.1}$$

Let $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$, also known as the precision matrix. We have n i.i.d. samples from this multivariate normal distribution such that $\mathbf{X} = \left\{\mathbf{x}^{(i)} \sim \mathcal{N}(\mathbf{0}, \Sigma)\right\}_{i=1}^{n}$. The graphical model aspect comes from the fact that given any pair of nodes $\mathbf{x_i}$ and $\mathbf{x_j}$ and a set of nodes $\mathbf{x_S}$, then $\mathbf{x_i}$ is conditionally independent of $\mathbf{x_j}$ given $\mathbf{x_S}$, thus following the Markov property. Mathematically,

$$\mathbf{x_i} \perp\!\!\!\perp \mathbf{x_j} \mid \mathbf{x_S}$$

There is an edge between nodes $\mathbf{x_i}$ and $\mathbf{x_j}$ if and only if $\mathbf{\Theta_{ij}} = (\mathbf{\Sigma^{-1}})_{\mathbf{ij}} \neq \mathbf{0}$. Therefore we can easily construct the graph if we know the entries of $\mathbf{\Theta}$ as it will directly give the edges between nodes. Therefore the problem of graph structure learning essentially boils down to learning the precision or the inverse covariance matrix $\mathbf{\Theta}$.

1.2.1 GGM optimization problem

GGMs are often formulated in the Maximum Likelihood Estimation (MLE) settings (Uhler, 2017). Given the data samples, $\mathbf{X} = \left\{\mathbf{x}^{(i)} \in \mathbb{R}^p\right\}_{i=1}^n$, the log likelihood for GGMs can be computed as follows:

$$\begin{aligned}
\ell(\mu, \Sigma) &\propto -\frac{n}{2}\log\det(\Sigma) - \frac{1}{2}\sum_{i=1}^{n}\left(\mathbf{x}^{(i)} - \mu\right)^T \Sigma^{-1}\left(\mathbf{x}^{(i)} - \mu\right) \\
&= -\frac{n}{2}\log\det(\Sigma) - \frac{1}{2}\operatorname{Tr}\left(\Sigma^{-1}\left(\sum_{i=1}^{n}\left(\mathbf{x}^{(i)} - \mu\right)\left(\mathbf{x}^{(i)} - \mu\right)^T\right)\right) \\
&= -\frac{n}{2}\log\det(\Sigma) - \frac{n}{2}\operatorname{Tr}\left(S\Sigma^{-1}\right) - \frac{n}{2}(\bar{\mathbf{x}} - \mu)^T\Sigma^{-1}(\bar{\mathbf{x}} - \mu)
\end{aligned} \tag{1.2}$$

where $\bar{\mathbf{x}}$ is the sample mean and $\mathbf{S}$ is the Sample Covariance Matrix (SCM) defined as:

$$\mathbf{S} = \frac{1}{n}\mathbf{X}\mathbf{X}^T = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x}^{(i)}\right)\left(\mathbf{x}^{(i)}\right)^\top$$

Maximizing the above log-likelihood to formulate an optimization problem over the precision matrix $\mathbf{\Theta}$ we get the following optimization objective:

$$\underset{\mathbf{\Theta}\succeq\mathbf{0}}{\text{maximize}}\ \log\det(\mathbf{\Theta}) - \operatorname{Tr}(\mathbf{\Theta}\mathbf{S}) - \alpha\mathbf{h}(\mathbf{\Theta}) \tag{1.3}$$

where an additional term $h(\Theta)$ is introduced to regularize Θ and α is the regularization parameter. Choice of $h(.)$ is dependent on the objective of formulation. One simple choice can be $l1$-regularization (Friedman *et al.*, 2008).

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

There is plenty of work in graph and optmization literature to learn graphs from data. Friedman *et al.* (2008) proposed Graphical Lasso (GLasso henceforth) to learn sparse graphs by imposing $l1$-norm penalty on the precision matrix. Solving the following optimization problem to obtain the positive semi-definite precision matrix Θ:

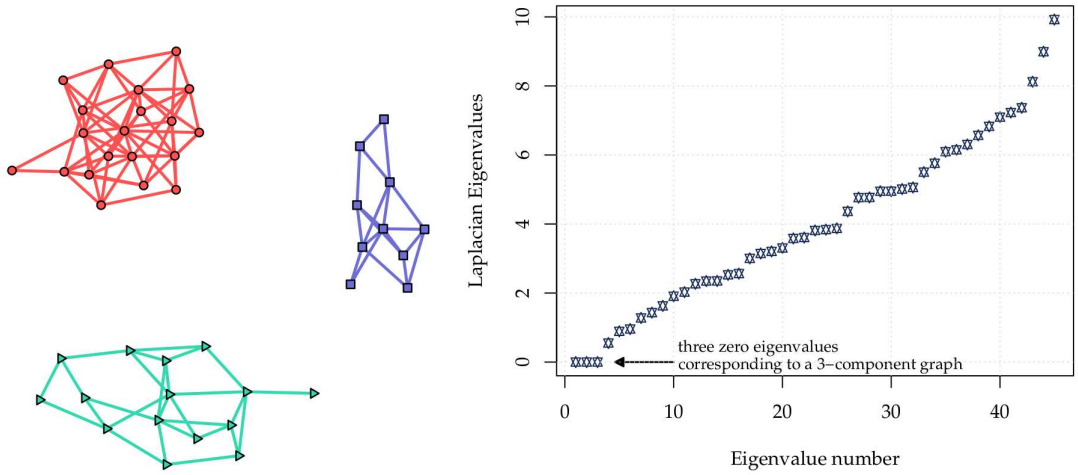
$$\min_{\Theta \succeq 0} \operatorname{tr}(S\Theta) - \log\det(\Theta) + \lambda\|\Theta\|_1$$

, where λ is the regularization parameter. Kumar *et al.* (2019, 2020) provided a unified approach for learning a large class of structured graph families, e.g., multi-component, bipartite, regular, etc. Their proposed algorithm learns graphs under the given structural constraints of the graph structure. These structural constraints are converted into the spectral constraints of graph and then imposed on either the laplacian matrix (SGL henceforth) or the adjacency matrix (SGA henceforth) or both of them (SGLA henceforth). The following optimization objective is solved to learn the graph:

$$\begin{aligned} \underset{\Theta}{\text{maximize}} \quad & \log \operatorname{gdet}(\Theta) - \operatorname{tr}(\Theta S) - \alpha h(\Theta) \\ \text{subject to} \quad & \Theta \in \mathcal{S}_\Theta, \lambda(\mathcal{T}(\Theta)) \in \mathcal{S}_\mathcal{T} \end{aligned}$$

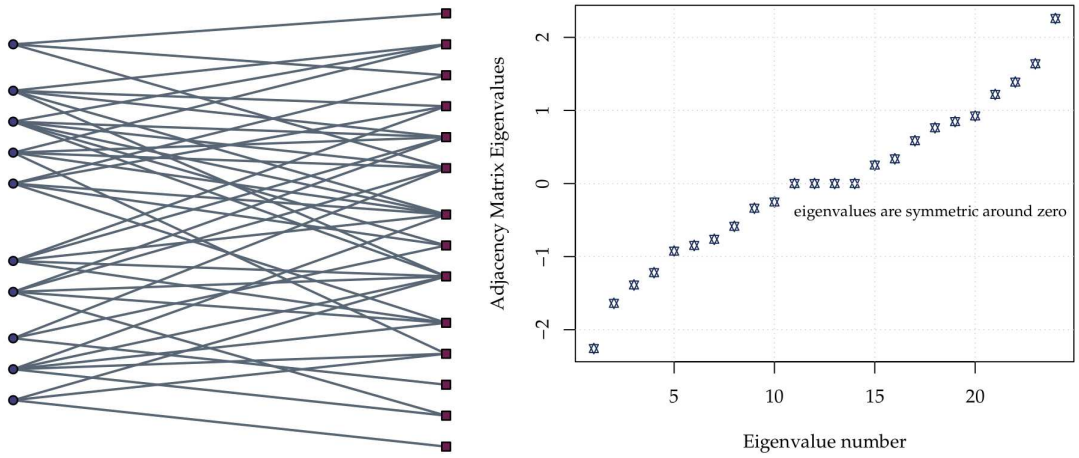
, where gdet $(\cdot)$ is the generalized determinant defined as the non-zero eigenvalues product, $\mathcal{S}_\Theta$ encodes the typical constraints of a Laplacian matrix , $\lambda(\mathcal{T}(\Theta))$ is the vector containing the eigenvalues of matrix $\mathcal{T}(\Theta)$, $\mathcal{T}(\cdot)$ is the transformation matrix to consider the eigenvalues of different graph matrices, and $\mathcal{S}_\mathcal{T}$ allows to include spectral constraints in the eigenvalues. The main idea of structured graph learning is that structural properties of graphs are encoded in their spectral properties. Fig 2.1 shows a 3-component graph and the distribution of corresponding eigenvalues of the Laplacian matrix with first three eigenvalues zero. For a k-component graph, first k of its eigenvalues are zero.

Similarly, Fig 2.2 shows a bipartite graph and its corresponding eigenvalues which are symmetric around origin.



(a) A 3-component graph. (b) Eigenvalue distribution of a 3-component graph.

Figure 2.1: 3-component graph and its eigenvalue distribution. Image src: Kumar *et al.* (2020)



(a) A bipartite graph. (b) Eigenvalue distribution for a bipartite graph.

Figure 2.2: Bipartite graph and its eigenvalue distribution. Image src: Kumar *et al.* (2020)

Therefore, the SGL/SGA/SGLA algorithms requires knowledge of the graph structure *a priori*, based on which it can learn the graph by incorporating the spectral constraints in the optimization objective. However, in many modern applications, e.g., network

medicine, gene regulatory networks, we do not have a prior information of the type of graph structure suitable for that particular data. In such cases structure identifiability from data becomes very crucial for building a graph-based application. Thus, before learning a graph it become imperative to identify best suitable type of graph structure for that data. This issue has not been explored in the literature, which will be the focus of our work. We envision that this problem can be tackled by integrating model selection, degree of freedom and conditional measure frameworks within the algorithmic framework developed by Kumar *et al.* (2020).

The previous attempts of graph model selection in GGMs tries to choose the best graph at the sparsity level (Lartigue *et al.*, 2020) or to enforce connectedness in the graph (Foygel and Drton, 2010). No prior work for model selection of graph type has been done earlier to the best of our knowledge. Our proposed formulation will allow us to directly learn the graph from data without requiring any structural (or any other) constraints.

CHAPTER 3

GRAPH MODEL SELECTION

3.1 Scope and Objectives

In order to tackle the problem of automatic graph structure learning, we use model selection along with the SGL/SGA/SGLA algorithm. If we can identify the structure of graph beforehand, we can directly learn the graphs from data. The underlying graph structure can be multi-component graph, bipartite graph, multi-component bipartite graph, grid graph etc. Given a data, we propose a generic criteria to identify the graph structure 'best' suitable to the data. Before we move forward, we need to ask what is 'best' here? Unlike the classical machine learning problems where our goal is to find the model closest to the truth (i.e. true data distribution) using KL Divergence, here we want to choose the model which is most likely to be the truth. We use model selection and degrees of freedom to tackle the structure identifiability problem. Chen and Chen (2008) introduced extended Bayesian Information Criterion (eBIC) to incorporate model complexity in the ordinary Bayesian Information Criterion (BIC) using degrees of freedom (DoF) of the model. BIC can be thought of as maximizing the posterior model probability conditioned on true data generating process. Foygel and Drton (2010) used eBIC for model selection in Gaussian Graphical models and found that it performs better than either cross-validation or the ordinary BIC. They also showed scalability of eBIC with number of nodes p and number of samples n and gave asymptotic bounds on the number of edges i.e. controlling the sparsity.

Foygel and Drton (2010) defined eBIC for GGMs by replacing the DoF with number of edges in the learned graph:

$$\text{eBIC}_\gamma(\hat{\mathbf{\Theta}}) = \mathbf{2}\ell_{\mathbf{n}}(\hat{\mathbf{\Theta}}) - |\mathcal{E}| \log \mathbf{n} - \mathbf{4}\gamma|\mathcal{E}| \log \mathbf{p} \tag{3.1}$$

where, $\hat{\mathbf{\Theta}}$ is the estimated precision matrix of the learned graph, $\ell_{\mathbf{n}}(\hat{\mathbf{\Theta}})$ is maximum likelihood estimate of the model, $|\mathcal{E}|$ is the number of edges counted using the unique

non-zero entries in $\hat{\mathbf{\Theta}}$ and $\gamma \in [0, 1]$ is the tuning parameter to control the sparsity of the learned graph. If,

- $\gamma = 0 \implies$ ordinary BIC
- $\gamma = 1 \implies$ additional sparsity
- $\gamma = 0.5 \implies$ good trade-off

Note: If $n >> p$, a good choice is $\gamma = 0$ and if $n \sim p$ choose $\gamma = 0.5$. Our main contribution is to employ eBIC for graph structure identification under some regularity conditions. This is the first such application of eBIC to the best of our knowledge. Our choice of eBIC is motivated due to several reasons:

- We want a parsimonious model with as few predictor variables as possible
- accounts for likelihood of different models
- $|\mathcal{E}|$ incorporates the degrees of freedom in the model
- Allows to learn sparse graph for high dimensional data $p >> n$

3.2 Proposed Algorithms

Our goal is to find the best graph structure for the given data, so that we can learn the graph using the SGL/SGA/SGLA algorithm. Any set of parameters of SGL/SGA/SGLA constitutes a model and we want to choose the model which is most likely to be the truth. Given a graph, finding the number of components in it is a well studied problem in the graph literature. But given the data whose underlying graph structure is multi-component, what is the best choice of k to learn the graph? This is the problem we would like to solve in our pursuit of structure identifiability. We propose Algorithm 1 to find the best choice of k to learn the graph. We loop over number of components upto K and estimate the precision matrix $\hat{\Theta}_k$ for k-component graphs using SGL algorithm. The degrees of freedom for the graph is determined through the non-zero entries in the precision matrix. A threshold can also be used to detemine the edges in some cases. Depending on the number of samples and number of nodes, we choose the value of γ, based on which eBIC is computed. Ideally the model with maximum eBIC score is most likely among the candidate models. Therefore the graph should be learned using this value of k using SGL algorithm, which is expected to be the most likely true graph. A detailed comparison of our proposed algorithm and k-means algorithm for clustering can be found in Section 4.

Algorithm 1: Component identification to learn best graph

Input: SCM S, number of nodes p, and the number of samples n.

for $k = 1, 2, ..., K$ **do**

  Estimate: $\hat{\mathbf{\Theta}}_{\mathbf{k}} \leftarrow \mathrm{SGL}\left(\mathbf{k}, \mathbf{c_1}, \mathbf{c_2}, ..\right)$ with $\mathbf{c_1}, \mathbf{c_2}$ as very small and very large values.

  Compute DoF: $|\mathcal{E}|_{\mathbf{k}} = \left\{\text{unique non-zeros in } \hat{\mathbf{\Theta}}_{\mathbf{k}}\right\}$, choose $\gamma_k \in [0, 1]$

  Compute:

  $$\mathrm{eBIC}\left(\hat{\mathbf{\Theta}}_{\mathbf{k}}\right) = \log \mathrm{gdet}\left(\hat{\mathbf{\Theta}}_{\mathbf{k}}\right) - \mathrm{Tr}\left(\mathbf{S}\hat{\mathbf{\Theta}}_{\mathbf{k}}\right) - |\mathcal{E}|_{\mathbf{k}} \log \mathbf{n} - \mathbf{4}\gamma_{\mathbf{k}}|\mathcal{E}|_{\mathbf{k}} \log \mathbf{p}$$

Choose k with maximum eBIC $\left(\hat{\Theta}_k\right)$

Our second goal is to identify whether the underlying graph structure of data is bipartite or not. For simplicity, we consider the connected bipartite case only. We use the SGA algorithm to estimate the bipartite precision matrix $\hat{\Theta}_{bip}$ and SGL algorithm to estimate precision matrix for single component graph (or simple connected graph) $\hat{\Theta}_1$. Then we compute the eBIC scores of the two models and conclude the structure which has higher eBIC score. Algorithm 2 describes this proposed formulation.

Algorithm 2: Bipartite identification: connected case

Input: Input: SCM S, number of nodes p, and the number of samples n.

Estimate: $\hat{\mathbf{\Theta}}_{\mathbf{bip}} \leftarrow \mathrm{SGA}\left(\mathbf{c_1}, \mathbf{c_2}, ..\right)$ with $\mathbf{c_1}, \mathbf{c_2}$ as very small and very large values.

Estimate: $\hat{\mathbf{\Theta}}_{\mathbf{1}} \leftarrow \mathrm{SGL}\left(\mathbf{k} = \mathbf{1}, \mathbf{c_1}, \mathbf{c_2}, ..\right)$ with $\mathbf{c_1}, \mathbf{c_2}$ as very small and very large values.

Compute DoF for bipartite structure: $|\mathcal{E}|_{\mathbf{bip}} = \left\{\text{unique non-zeros in } \hat{\mathbf{\Theta}}_{\mathbf{bip}}\right\}$, choose $\gamma_{bip} \in [0, 1]$

Compute DoF for simple connected structure: $|\mathcal{E}|_{\mathbf{1}} = \left\{\text{unique non-zeros in } \hat{\mathbf{\Theta}}_{\mathbf{1}}\right\}$, choose $\gamma_1 \in [0, 1]$

Compute:

$$\mathrm{eBIC}\left(\hat{\mathbf{\Theta}}_{\mathbf{bip}}\right) = \log \mathrm{gdet}\left(\hat{\mathbf{\Theta}}_{\mathbf{bip}}\right) - \mathrm{Tr}\left(\mathbf{S}\hat{\mathbf{\Theta}}_{\mathbf{bip}}\right) - |\mathcal{E}|_{\mathbf{bip}} \log \mathbf{n} - \mathbf{4}\gamma_{\mathbf{bip}}|\mathcal{E}|_{\mathbf{bip}} \log \mathbf{p}$$

Compute:

$$\mathrm{eBIC}\left(\hat{\mathbf{\Theta}}_{\mathbf{1}}\right) = \log \mathrm{gdet}\left(\hat{\mathbf{\Theta}}_{\mathbf{1}}\right) - \mathrm{Tr}\left(\mathbf{S}\hat{\mathbf{\Theta}}_{\mathbf{1}}\right) - |\mathcal{E}|_{\mathbf{1}} \log \mathbf{n} - \mathbf{4}\gamma_{\mathbf{1}}|\mathcal{E}|_{\mathbf{1}} \log \mathbf{p}$$

Chose the structure which has maximum eBIC $\left(\hat{\Theta}_{\mathrm{bip}}\right), \mathrm{eBIC}(\hat{\Theta}_1)$

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Experiments

We thoroughly investigate our proposed algorithms on synthetic as well as real datasets and conduct extensive experiments through systematic evaluation. We use k-means algorithm as baseline for number of components identification (see Algorithm 1) from data (NOT graph!). Whereas, it is not possible to compare the performance of bipartite identification (see Algorithm 2) for real datasets, we demonstrate its performance for synthetic dataset.

Synthetic datasets We first construct an adjacency matrix for multi-component graph and bipartite graphs such that the laplacian matrix is positive semi-definite i.e. $\mathbf{\Theta}^\dagger \succeq 0$. The edge weights are considered equal for simplicity. This is the true precision matrix of the graph. We compute the pseudo-inverse of precision matrix to get the covariance matrix. We sample n observations from the p-dimensional multivariate normal distribution $\mathbf{X} \sim \mathcal{N}\left(0, \Theta^\dagger\right)$ to obtain the synthetic data. We open source all our codes and experiments to further advance the research in graph learning and graph model selection on Github.

Real datasets We empirically evaluate the performance of our algorithms on following multivariate real datasets:

1. **Cancer RNA-Seq dataset** We consider the huge dataset (Weinstein *et al.*, 2013) consisting of 801 nodes and 20531 samples. It consists of genetic features of patients having 5 types of cancers: breast carcinoma (BRCA), kidney renal clear-cell carcinoma (KIRC), lung adenocarcinoma (LUAD), colon adenocarcinoma (COAD), and prostate adenocarcinoma (PRAD). Due to computational limitations, we consider a subset of the data consisting of first 70 nodes following Kumar *et al.* (2020).
2. **Iris dataset** It consists of sepal and petal dimensions of 3 different species of iris flowers: Iris Setosa, Iris Versicolour and Iris Virginica. The dataset (Fisher, 1936) comprises of 150 samples with 4 feature attributes.

Both the above datasets are available at UCI Machine Learning Repository.

4.1.1 Evaluation metrics

We use the Relative Error and F1-score metrics for evaluation of synthetic datasets where both the true precision matrix $\Theta^\dagger$ is known. $\hat{\Theta}$ is the estimated precision matrix through our algorithms. In case of real datasets, we use accuracy for the number of components identified in the data and visual inspection for the animals dataset where we don't know the true number of components. In F1-score, True Positive (tp) denotes the case when there is an edge in the actual graph and the estimated graph also predicts an edge; False Positive (fp) stands for the case when there is no edge in true graph but there is an edge in the estimated graph; and false negative (fn) denotes the case when there is an edge in the true graph but the estimated graph misses it. F1-score takes values between 0 and 1 where 1 denotes perfect structure recovery (Egilmez *et al.*, 2016).

$$\text{Relative Error } = \frac{\left\|\hat{\mathbf{\Theta}} - \mathbf{\Theta}^\dagger\right\|_{\mathbf{F}}}{\|\mathbf{\Theta}^\dagger\|_{\mathbf{F}}}, \text{ F1-Score } = \frac{\mathbf{2}\,\mathrm{tp}}{\mathbf{2}\,\mathrm{tp} + \mathrm{fp} + \mathrm{fn}}$$

4.2 Results

We present the results for Algorithm 1 and Algorithm 2 in this section.

4.2.1 Component identification

Synthetic Datasets

Data generation: samples $\mathbf{X} = \left\{\mathbf{x}^{(i)} \in \mathbb{R}^p\right\}_{i=1}^n$ from a 4-component GGM with 10 nodes in each component such that $n = 1200$ and $p = 40$, assuming equal number of nodes in each component and equal edge weights.

We test our proposed Algorithm 1 on the above data by feeding in the sample covariance matrix S to SGL algorithm for different number of components k. The learned k-component graphs are shown in Fig 4.1, where we can clearly see the learned graph for $k = 4$ is most likely to be the true graph which can be also be visually confirmed. Our algorithm predicts the number of components in the graph from which the data is sampled is 4 as it has obtained the maximum eBIC score in Fig 4.2. Infact,

our algorithm is not only accurately able to infer the true number of components from the data, but also able to clearly distinguish between bipartite and multi-component graph. Bipartite graph structure earns eBIC score of 789.834 which is much lower than the eBIC score of 4-component graph structure, thus distinguishing between the two. F1-scores, relative error and eBIC scores for all graph structures can be seen in Table 4.1.

Table 4.1: k-component graph structure identification results on 4-component GGM

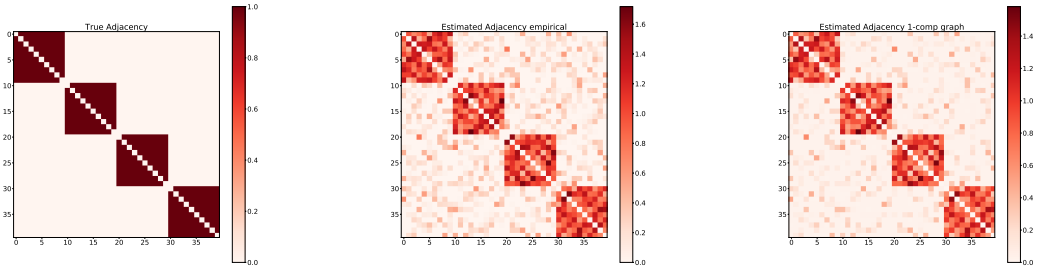
Metric	F1-Score	Relative Error	eBIC Score
True Adj	1.000	0	
Empirical Adj	0.375	0.375	
1-component Adj	0.375	0.341	-1946.963
2-component Adj	0.534	0.678	-17.131
3-component Adj	0.728	0.450	1056.563
4-component Adj	0.923	0.234	1532.479
5-component Adj	0.897	0.998	1496.080
6-component Adj	0.826	0.998	1309.700
7-component Adj	0.808	0.999	1282.476
8-component Adj	0.749	1.000	1157.726
9-component Adj	0.712	0.999	1095.420
10-component Adj	0.686	0.999	1036.030
Bipartite Adj	0.454	1.073	789.834

Real Datasets

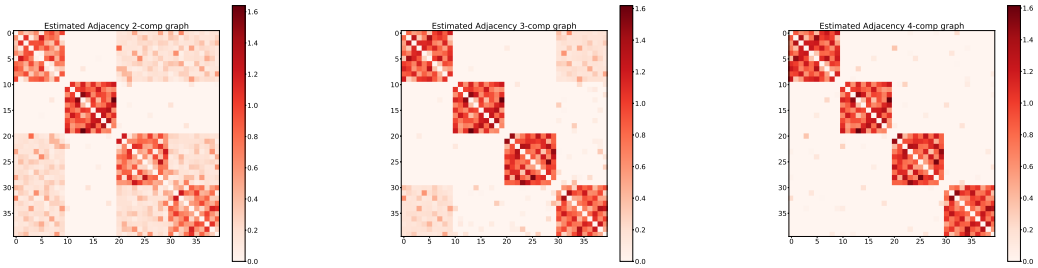
Baseline: We compare our component identification with the k-means algorithm to find the number of clusters in the data. We use the elbow method to find the optimum k for k-means. The elbow method uses plot of inertia or Within-cluster Sum of Squared distances (WSS) vs k and concludes the optimal k at which the plot forms an elbow. The WSS tends to zero as we increase the k since the points will become cluster centroids eventually making the square distances zero.

Cancer RNA dataset

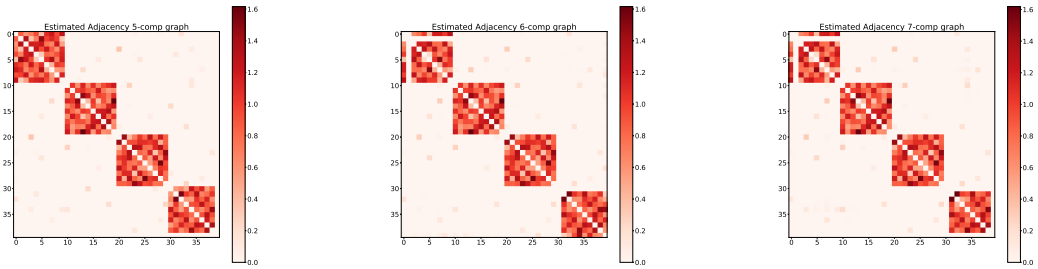
The true number of components in RNA dataset is 5. It is not very clear from Fig 4.3 that the elbow is at $k = 4$ or at $k = 5$. Whereas, the eBIC scores for our algorithm



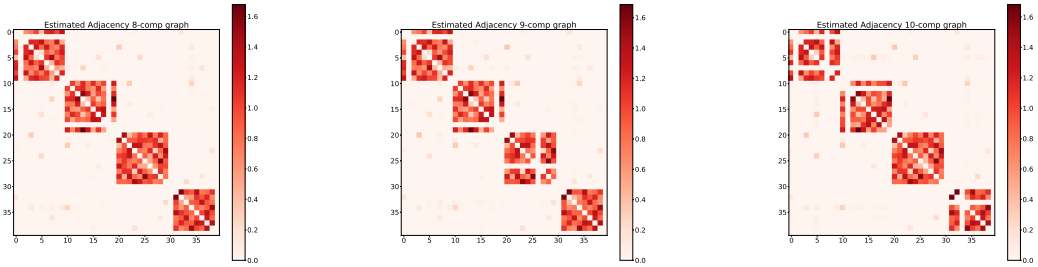
(a)



(b)



(c)



(d)

Figure 4.1: True, empirical and estimated adjacency matrix for $k = 1, 2, ..., 10$ component graphs

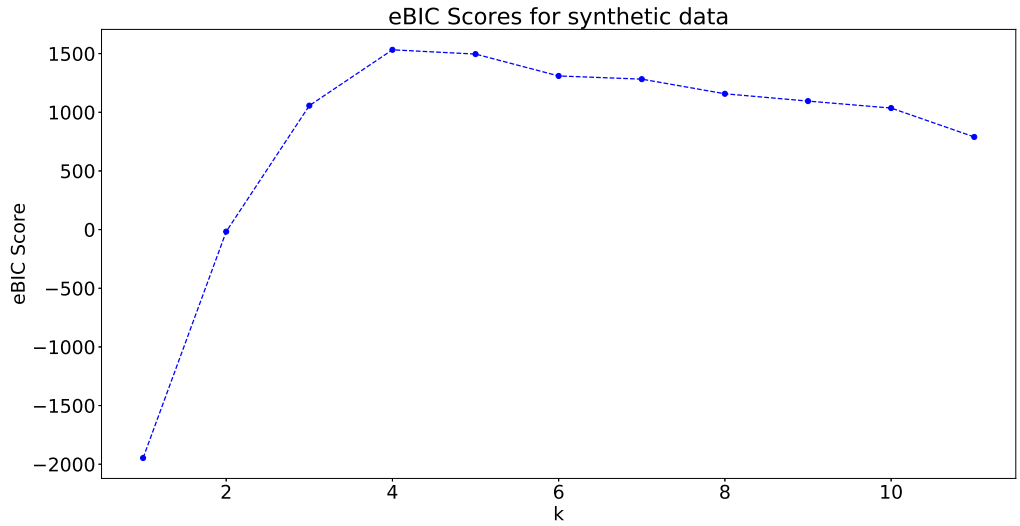


Figure 4.2: Plot of eBIC scores for 4-component GGM synthetic data

are shown in Fig 4.4. The reason for high eBIC scores for $k < 5$ can be attributed to the phenomenon of model mismatch. The SGL algorithm is able to perform well even when the graph is learned for mismatched value of k (Kumar *et al.*, 2020). We believe model mismatch is the cause for almost similar scores for different values of k.

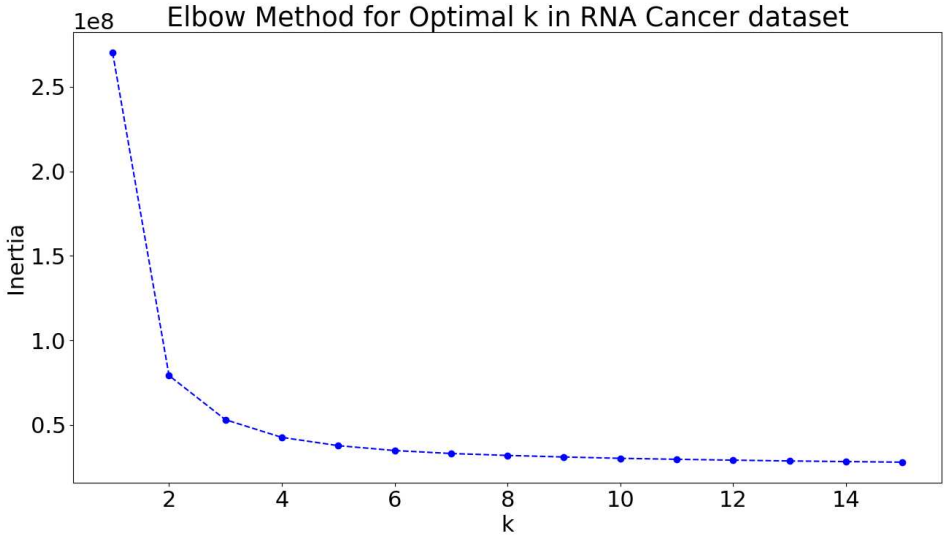


Figure 4.3: variation of k-means inertia with number of clusters k for RNA dataset

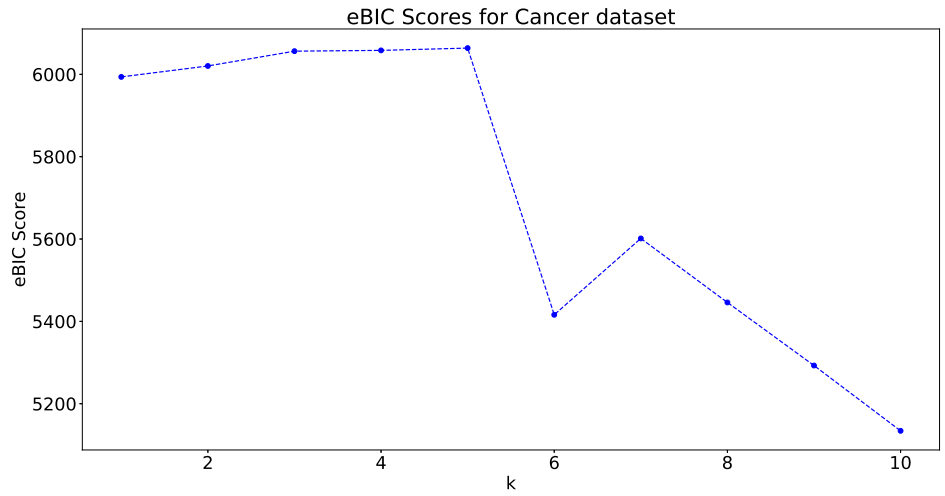


Figure 4.4: eBIC scores of estimated k-component graphs for RNA dataset

Iris dataset

The iris dataset has 3 classes corresponding to each species of iris flower. Fig 4.5 shows the WSS scores for different values of k indicating that $k = 3$ or $k = 4$ number of clusters must be there. Fig 4.6 shows the eBIC scores using our proposed component identification algorithm. It can be seen that after $k = 3$, eBIC scores fall down indicating that $k = 3$ should be the true number of clusters or components in the data. The eBIC scores for $k < 3$ are similar to $k = 3$. We again attribute this to the model mismatch of SGL algorithm.

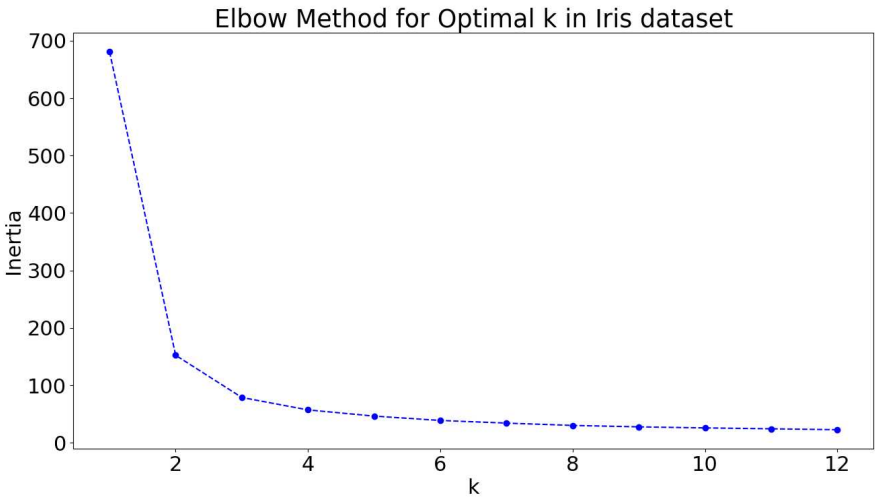


Figure 4.5: variation of k-means inertia with number of clusters k for Iris dataset

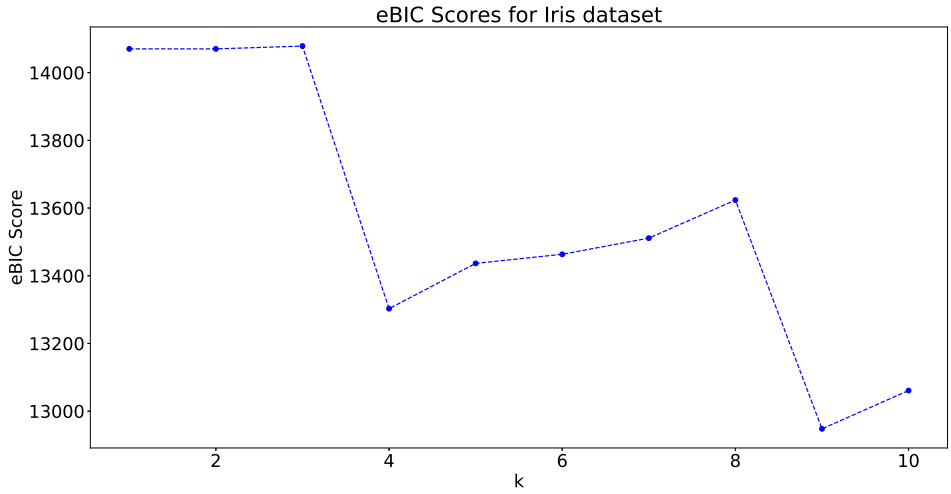


Figure 4.6: eBIC scores of estimated k-component graphs for Iris dataset

4.2.2 Bipartite identification

Synthetic datasets

Data: samples $\mathbf{X} = \left\{\mathbf{x}^{(i)} \in \mathbb{R}^p\right\}_{i=1}^n$ from bipartite GGM with 10 nodes in both sets making $p = 20$ and $n = 1600$ samples.

Fig 4.7 shows the true, empirical and learned adjacency matrix using 1-component SGL and SGA algorithm respectively. Both the SGL and SGA algorithms are only provided with the sample covariance matrix S. It can be visually confirmed that SGA has learned graph which is more likely to be the true graph. We compute eBIC scores (higher is better) for the two graph structures in Table 4.2. It is evident that our proposed Algorithm 2 selects the bipartite structure, which is indeed the underlying graph structure in this case. This upholds the validity of our proposed algorithm.

Table 4.2: Bipartite identification results on bipartite GGM with 10 nodes in both set

Metric	F1-Score	Relative Error	eBIC Score
True Adj	1.0	0	
Empirical Adj	0.689	0.123	
1-component Adj	0.766	0.997	-512.066
Bipartite Adj	0.959	0.134	-176.895

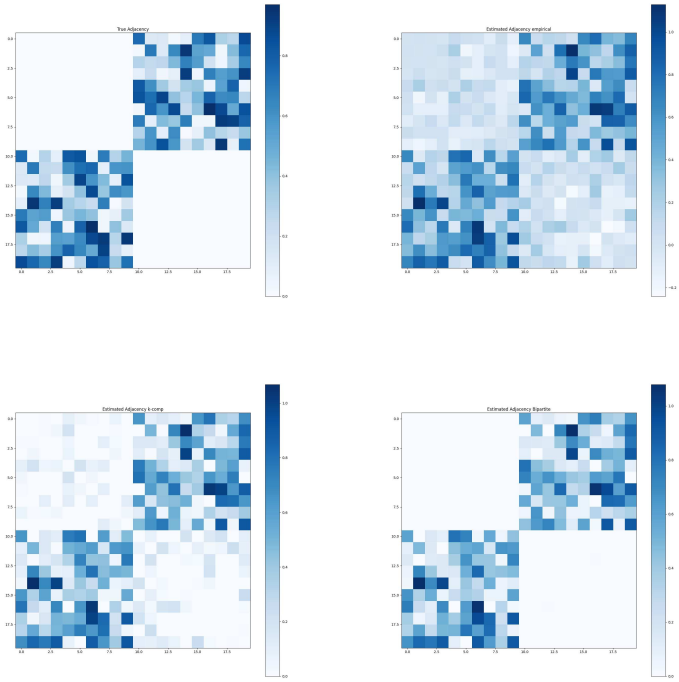


Figure 4.7: First row: True Adjacency matrix for bipartite GGM; empirical adjacency matrix respectively. Second row: adjacency matrix learned using 1-component SGL algorithm; adjacency matrix learned using bipartite SGA algorithm.

Data: samples $\mathbf{X} = \left\{\mathbf{x}^{(i)} \in \mathbb{R}^p\right\}_{i=1}^n$ from bipartite GGM with 10 and 6 nodes in each set making $p = 16$ and $n = 1600$ samples.

Given a data sampled from bipartite GGM, our algorithm always selects bipartite structure in contrast to the k-component structure. We have confirmed this observation with different numbers of nodes in each sets and different number of samples. Table 4.3 shows results for the above data, validating the algorithm. Though the scalability of nodes couldn't be tested due to $O(p^3)$ complexity of the algorithm.

Table 4.3: Bipartite identification results on bipartite GGM with 10 and 6 nodes in each set

Metric	F1-Score	Relative Error	eBIC Score
True Adj	1.0	0	
Empirical Adj	0.667	0.117	
1-component Adj	0.708	0.122	-458.748
Bipartite Adj	0.957	0.158	-198.36

CHAPTER 5

CONCLUSION AND FUTURE IDEAS

5.1 Conclusion

Graph learning is an interesting problem to solve at the intersection of graph theory and machine learning with lots of potential applications. Existing graph learning methods depend on the graph structure *a priori* and thus have little practical applications in the real world where we often have no prior information about the data. Our work integrates structured graph learning and graph model selection to solve this limitation through identifying the underlying graph structure of the data. As a starting point, we have proposed algorithms for component identification and bipartite identification and have empirically demonstrated their performance through exhaustive evaluation in Section 4. Our proposed Algorithm 1 for component identification correctly identifies true number of components for synthetic datasets as well as real datasets and provides better results than k-means algorithm. Our Algorithm 2 for bipartite identification is also able to distinguish between simple connected graph and bipartite graph for synthetic datasets. This structure identification will allow us to learn graphs which are most likely to be the true graph or gaussian graphical model from which the data is sampled.

Our algorithms uses SGL algorithm under the hood which required eigenvalue decomposition. Therefore the time complexity of our proposed algorithms is $O(p^3)$ where p is the number of nodes. This potentially limits the scalability of the algorithm and provides a scope of future work. Whereas the k-means algorithm takes $O(t*k*n*p)$ time for a fixed number t of iterations, n (p-dimensional) points and k number of centroids (or clusters).

5.2 Future Work

We believe the following directions must be a good starting point for further research:

Extension to other graph structures: Our work currently supports component and bipartite identification. The future work will focus on improving the criteria to incorporate different graph structures such as multi-component bipartite, Erdos-Renyi Graphs etc. We also envisage that more research and stronger baselines will come up for the problem of graph structure identification from data. Also, moving to non-$i.i.d.$ assumptions on data can further strengthen the research and practical applications, which is true for many machine learning problems.

Spectral density based methods: As spectral properties influence graph structure, we think it will be an interesting direction to develop spectral density based methods for structure identification. This can help in making the algorithm scalable to large number of nodes.

REFERENCES

1. **J. Chen** and **Z. Chen** (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, **95**(3), 759–771.

2. **X. Dong**, **D. Thanou**, **M. Rabbat**, and **P. Frossard** (2019). Learning graphs from data: A signal representation perspective. *IEEE Signal Processing Magazine*, **36**(3), 44–63.

3. **H. E. Egilmez**, **E. Pavez**, and **A. Ortega** (2016). Graph learning from data under structural and laplacian constraints. *arXiv preprint arXiv:1611.05181*.

4. **R. A. Fisher** (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, **7**(2), 179–188.

5. **R. Foygel** and **M. Drton**, Extended bayesian information criteria for gaussian graphical models. *In Advances in neural information processing systems*. 2010.

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

7. **S. Kumar**, **J. Ying**, **J. V. de Miranda Cardoso**, and **D. Palomar**, Structured graph learning via laplacian spectral constraints. *In Advances in Neural Information Processing Systems*. 2019.

8. **S. Kumar**, **J. Ying**, **J. V. de Miranda Cardoso**, and **D. P. Palomar** (2020). A unified framework for structured graph learning via spectral constraints. *Journal of Machine Learning Research*, **21**(22), 1–60.

9. **T. Lartigue**, **S. Bottani**, **S. Baron**, **O. Colliot**, **S. Durrleman**, and **S. Allassonnière** (2020). Gaussian graphical model exploration and selection in high dimension low sample size setting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

10. **K. Nakai** and **M. Kanehisa** (1991). Expert system for predicting protein localization sites in gram-negative bacteria. *Proteins: Structure, Function, and Bioinformatics*, **11**(2), 95–110.

11. **W. J. Nash**, **T. L. Sellers**, **S. R. Talbot**, **A. J. Cawthorn**, and **W. B. Ford** (1994). The population biology of abalone (haliotis species) in tasmania. i. blacklip abalone (h. rubra) from the north coast and islands of bass strait. *Sea Fisheries Division, Technical Report*, **48**, p411.

12. **E. E. Smith**, **E. Shafir**, and **D. Osherson** (1993). Similarity, plausibility, and judgments of probability. *Cognition*, **49**(1-2), 67–96.

13. **C. Uhler** (2017). Gaussian graphical models: an algebraic and geometric perspective. *arXiv preprint arXiv:1707.04345*.

14. **J. N. Weinstein**, **E. A. Collisson**, **G. B. Mills**, **K. R. M. Shaw**, **B. A. Ozenberger**, **K. Ellrott**, **I. Shmulevich**, **C. Sander**, **J. M. Stuart**, **C. G. A. R. Network**, *et al.* (2013). The cancer genome atlas pan-cancer analysis project. *Nature genetics*, **45**(10), 1113.