Estimation of Sparse Graph with Lifecycle

Shuo Liu*
Undergraduate Thesis
Information and Computing Science
Fudan University

Abstract Due to its capability of recovering the structure of the Gaussian graphical model (GGM), the sparse estimation of inverse covarience matrix has become an attractive topic of research. The *graphical lasso* approach intends to learn the sturcture of a GGM by minimizing the log-likelihood of the data with a l_1 penalty on the elements of the inverse covarience matrix. Yet the result of the graphical lasso in real-world application can be difficult to interpret because no prior knowledge is taken into consideration. To overcome the challenge, some researches apply domain knowledge to the graphical lasso. In this paper, we consider a specific kind of graphical model, which subject to a constrain derived from the *lifecycles*, the period of time in which the nodes are active. With the constrain the graphical lasso problem can be solved with significant improvements on both efficiency and effectiveness.

1 Introduction

Undirected graphical model with capability of showing the complex relevance among a set of random variables has been gaining more attention recently. For instance, a group of scholars coauthor with each other indicates close cooperation among them; a group of genes with similar functions are more likely to work together. We then can have a graph to represent the academic circles or genome by regarding scholars or genes as nodes and the relationship between them as edges. Yet in many applications, we can only reveal the sturcture of such graphs from limited samples of observation.

Gaussian graphical model (GGM)[4] is one of the probablistic graphical models that are frequently used. The model assumes that random variables (or nodes) follow a multivariate Gaussian distribution. The learning of the graphical model is thus equivalent to the estimation of the inverse covirance matrix, or precision matrix, because the non-zero elements in precision matrix correspond to the edges in the graphical model. To keep a low complexity of the model, and have the result interpretable, it is natrual to try to keep a sparse precision matrix. For this purpose, researchers proposed the problem of the sparse estimation of the inverse covariance matix, or *Graphial Lasso* problem[1].

Nevertheless, with the scale of data sets growing larger, many algorithms for learning GGMs become uninterpretable since the number of connections is too large, or even impractical when the number of nodes exceeds tens of thousands. Based on such consideration, additional information should be taken into account. In a biosystem, a pair of gene can only be connected if they are in a same cellular process, by which M. Grechkin *et al.* are inspired to propose the method namely Pathway Graphical Lasso[3]. With more information, the method comes up with results more meaningful and solving process accelerated.

Similarly, we consider another scenario in real-life. Graduate students are more likely to co-operate with those whose year of admission is near or close to them when conducting academic research, since they are more familiar with and eazier to reach each other. Besides, as graduate *students*, they can only conducting such research after the commencement and before the end of their graduate life. Or the proteins in the proteome of a cell can only be active while they exists in the cell, and many proteins only exist in some certain periods of time. So the proteins connected with each other should at least have the chance of co-exist.

*Translated from the original Chinese version.

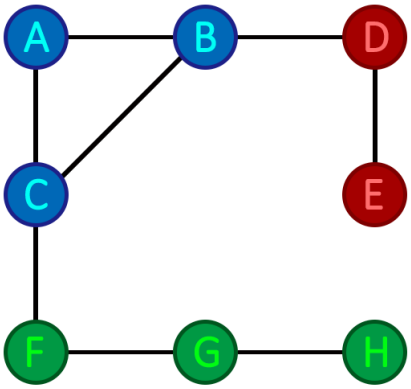


Figure 1: A simple example of the graphical model with lifecycle, in which nodes in $\{D, E\}$, $\{A, B, C\}$ and $\{F, G, H\}$ respectively start their lifecycle in the 1st, 2nd and 3rd stage.

By treating each student (or protein) as a node, we can build a network of relationship by the cooperations among them. The nodes have these natures: every nodes has a specific *'lifecycle'*, and the node can only have activites during its lifecycle. To simplify the problem, we divide the whole time of obsevation into T *stages* of time. Lifecycles of nodes start in some stages, and last for limited stages. There is a constrain implied by the lifecycle, requiring no connection between nodes whose lifecycles have completely no overlap. Besides, in practical applications, there is also a constrain regarding the distance of the beginning of the lifecycles between nodes. If the interval of the lifecycle beginning is too large, they would not have connections even if their lifecycle are overlapping. We name the later constrain 'proximity constrain'. Noted that the constrain implied by lifecycle is implicited in the proximity constrain.

A simple example is shown by Figure 1. Because of the proximity constrain (in this case we require the difference between their start stages is no more than 1), there should be no connections between one node from $\{D, E\}$ and the other from $\{F, G, H\}$. With information of the constrain mentioned above given *a priori*, the result of learning the GGMs will be more accurate. We will adapt the Pathway Graphical Lasso method to solve the graphical lasso problem with lifecycle. Experiments show that this method will have better performance on both synthetic and real-world datasets than the traditional graphial lasso method.

2 Preliminaries

The mathematical notations used in this paper can be find in Table 1.

Table 1: Mathematical Notations

Notation	Description
$\mathbb{R}$	The set of real numbers
$\mathcal{S}^n_{++}$	The set of $n \times n$ symmetric positive definite matrices
$\mathcal{S}^n_{+}$	The set of $n \times n$ symmetric positive semi-definite matrices
$A \succ 0$	Matrix A is positive definite
$A \succeq 0$	Matrix A is positive semi-definite
$\|\cdot\|_{1,\Lambda}$	The element-wise ℓ_1 norm of a matrix, *i.e.*, $\|A\|_{1,\Lambda} = \Sigma_{ij}\Lambda_{ij}\|A_{ij}\|$
$\biguplus$	Disjoint union
$\mathbf{card}\,(\cdot)$	The cardinal number of a set
$\mathbf{enumerate}\,(\cdot)$	The `enumerate` function in Python. For $A = \{a_{i_1}, ..., a_{i_k}\}$, $\mathbf{enumerate}\,(A) = \{[1, a_{i_1}], ..., [k, a_{i_k}]\}$

2.1 Gaussian Graphical Model

Probablistic graph model use graph to represent the relevance among random variables, one of which is Gaussian graphical model[4]. The nodes in the graph represents random variables, and the existences of edges in the graph represent the relevence of the variables.

In Gaussian graphical model,

$$\begin{aligned} & \forall \text{ random variables } x_j \text{ and } x_k, (1 \leq j, k \leq n) \text{ are conditionally independent} \\ \Leftrightarrow \quad & \operatorname{cov}(x_j, x_k | x_i, i \neq j, k) = 0 \\ \Leftrightarrow \quad & \text{There are no edge connecting nodes } j \text{ and } k. \end{aligned}$$

Let $\Sigma \in \mathcal{S}_{++}^n$ be the covariance matrix, and $\Theta = \Sigma^{-1} \in \mathcal{S}_{++}^n$ be the precision matrix. Then

$$\text{No edge connects nodes } j \text{ and } k \quad \Leftrightarrow \quad \Theta_{jk} = 0.$$

Based on this fact, we can easily describe the GMM with its precision matrix Θ. Besides, we can enforce the sparsity of the model by making as many elements in Θ be zero as possible.

2.2 Graphical Lasso

Given a set of samples drawn from an i.i.d. n-variate Gaussian distribution with mean of zero and covirance matrix $\Sigma \in \mathcal{S}_{++}^n$

$$x_i \sim \mathcal{N}_n(0, \Sigma), i = 1, 2, ..., m.$$

The target of the problem is to estimate the precision matrix $Theta$ with the assumption of its sparsity.

To estimate the precision matrix Θ, many researchers (e.g. [1, 2, 8, 9]) have proposed the log-likelihood problem:

$$\min_{\Theta} -l(S, \Theta) + \|\Theta\|_{1,\Lambda}, \tag{1}$$

where $S = \frac{1}{m}\Sigma_{i=1}^m x_i x_i^T \in \mathcal{S}_+^n$ is the sample covairance matix, Λ is an ℓ_1 regularization parameter matrix with $\Lambda_{ij} > 0$ for all off-diagonal elements and $\Lambda_{ij} \geq 0$ for all diagonal elements, and $l(S, \Theta) = \log\det\Theta - \operatorname{tr}(S\Theta)$ is the log-likelihood funtion. Note that $\Theta \succ 0$ is implicit, for $\log\det\Theta = -\infty$ if $\Theta \not\succ 0$.

Friedman *et al.*[2] proposed the method to solve (1) with block-coordinate descent approach.

3 Problem Formulation

In this section we propose the defination of lifecycle and then the formulation of the estimation problem with lifecycle.

3.1 Lifecycle

Definition 1 (Lifecycle)**.** *In a information network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the lifecycle of a node represents the period of time during which the node is active. For $\forall v \in \mathcal{V}$, $v \in \mathcal{L}_i$ denotes that the lifecycle of node v starts in the ith stage of time. Besides, the life-cycle of every node lasts for L stages.*

The whole observation lasts for T stages. For $\forall v$, if $v \in \mathcal{L}_i$, the activites of node v can only be observed between the $\max\{i, 1\}$th and the $\min\{i + L, T\}$th stage.

Like in those real-world examples, we know that a student can only cooperate with his/her schoolmate while the other one and him are at school at the same time. Furthermore, they are inclined to cooperate with those whose grade is near. We can define such constrain as below.

Definition 2 (Proximity constrain)**.** *Only nodes with start of their lifecycle in stages of time with difference no more than $r > 0$ can have connectivity between each other,* i.e.*,*

$$\Theta_{ij} = 0, \forall v_i \in \mathcal{L}_k, v_j \in \mathcal{L}_l, |k - l| > r. \tag{2}$$

We call such constrain the 'proximity constrain' and r the 'proximity constrain parameter'.

With the two definitions above, we can now formulate the graphical lasso problem with lifecycle.

3.2 Problem Formulation

Consider a set of n nodes with their lifecycle $\ell(v_i), i = 1, 2, ..., n$. Because of the proximity constrain, we now have the estimation problem:

$$\begin{aligned} \min_{\Theta} \quad & -l\left(S, \Theta\right) + \|\Theta\|_{1,\Lambda} \\ \text{s.t.} \quad & \Theta_{ij} = 0, \forall i, j, v_i \in \mathcal{L}_k, v_j \in \mathcal{L}_l, |k - l| > r \end{aligned} \tag{3}$$

where $\mathcal{L}_k, 0 \leq k \leq L$ stands for the set of nodes whose lifecycle start in the kth stage, $\biguplus_{0 \leq k \leq L} \mathcal{L}_k = \{v_1, ..., v_n\}$.

4 Proposed Method

In section, we propose the method based on *pathway graphical lasso* method.

4.1 Pathway Graphical Lasso

Grechkin *et al.* proposed a method for GGMs with additional information, *i.e.* the pathway graphical lasso method. We now briefly illustrate the method.

Define k 'pathways' $P_1, ..., P_k$ are sets of nodes. The method assumes that only nodes in a same pathway are possible to be connnected. In other words, let $\mathcal{F} = \Sigma_{t=1}^{k} \{(v_i, v_j) \,|v_i, v_j \in P_t\}$, and edges out of $\mathcal{F}$ are set to zero. The modified graphical lasso problem

$$\begin{aligned} \min_{\Theta} \quad & -l\left(S, \Theta\right) + \|\Theta\|_{1,\Lambda} \\ \text{s.t.} \quad & \Theta_{ij} = 0, (i, j) \notin \mathcal{F}, \end{aligned} \tag{4}$$

is the target of pathway graphical lasso.

Grechkin *et al.* also employ a version of the block-coordinate descent approach to solve the problem, in which the method updates the parameters corresponding to one pathway with all of the other parameters held fixed. The method is more efficient, and the result of it is more accurate.[3]

4.2 Adaptation

We rewrite the formulation (3) to conform it to the form of pathway graphical lasso. Let

$$P_k = \bigcup_{\substack{|t-k| \leq r \\ 1 \leq t \leq T}} \mathcal{L}_t, \quad 1 \leq k \leq T, \tag{5}$$

and we can regard P_k above as 'pathways'. According to the proximity constrain (2), those nodes who are possible to be connected are put in the same pathway. Now we have $\mathcal{F} = \Sigma_{t=1}^{k} \{(v_i, v_j) \,|\, (v_i, v_j) \in P_t\} = \{(v_i, v_j) \,|v_i \in \mathcal{L}_k, v_j \in \mathcal{L}_l, |k - l| \leq r\}$. With $\mathcal{F}$, problem (3) is equivelant to (4).

Note that pathways obtained by (5) have large scale of overlap with each other; in real-world problems, there can be empty $\mathcal{L}_k$ which will lead to a pathway be the subset of some other pathway. In those cases pathway graphical lasso method can not proceed directly. The situations of overlapping and subsets should be cleared before the employment of the method. Also note that the sample sets we use for an input also have a temporal feature. Samples took in different stages of time match with the connectivities implied by the information of lifecycles.

We present the method in detail in Algorithm 1, in which solving the Problem (4) we use the method of pathway graphical lasso mentioned in section 4.1.

5 Experiments

In this section, we introduce the results of our experiments, from which we can see the better performance of the proposed method.

Algorithm 1 Graphical lasso method with lifecycle

Input: Number of nodes n, sample set X, information of lifecycles $\{\mathcal{L}_i\}_{i=1}^{T}$, and proximity constrain parameter r.
Output: The estimated precision matrix Θ.

```
According to (5), get {P_k}_{k=1}^T using {L_i}_{i=1}^T
pt ← 1
repeat
    if ∃t ≠ pt, P_pt ⊆ P_t then
        Remove P_pt from {P_k}
    else
        pt++
    end if
until pt > card({P_k})
Using {P_k} as constrain to solve the Problem (4). Get the result Θ
return Θ
```

5.1 Data Collection

We conducted experiments on both synthetic and real-world datasets. The synthetic datasets were generated as follows:

Given the number of random variates (or nodes) n, the total number of stages T and the proximity constrain parameter r, we can firstly generate the symmetric positive definite matrix $\Theta^{(0)}$. Then we randomly generate a group of T non-negative numbers $s_1, ..., s_T, \Sigma_{i=1}^{T} = n$, which stands for the number of nodes start their lifecycles in each stage. Modifying the $\Theta^{(0)}$ by setting those elements which are incompatible with the proximity constrain, we now have a precision matrix Θ as ground truth. Generate a group of samples X, in which we combine groups of samples for all T stages to make sure that the samples are also in agreement with lifecycles and the proximity constrain.

We also use one real-world dataset. **DBLP**[1] is a bibliographical database of computer science. Our datasets is a subset of DBLP. We selected the current and former graduate and post-graduate students in Data Mining Research Group, UIUC led by Prof. Jiawei Han with a total number of 50. We drew the lifecycle information from the website of the research group[2], and 508 pulications of the students mentioned above since 2007 from DBLP as samples of observation. In this case, we let $r = 1$.

5.2 Compared Methods

We tested with following methods to validate the effectiveness of our proposal:

1. *The proposed method*, as described in Section 4. Take samples X, lifecycle information of nodes $\mathcal{L}_k, \forall k$ and the proximity parameter r as input, the method returns a precision matrix Θ as result.

2. *Graphical Lasso.* We put the sample X directly into graphical lasso, for it cannot accept any more information. We can have a precision matrix Θ as result.

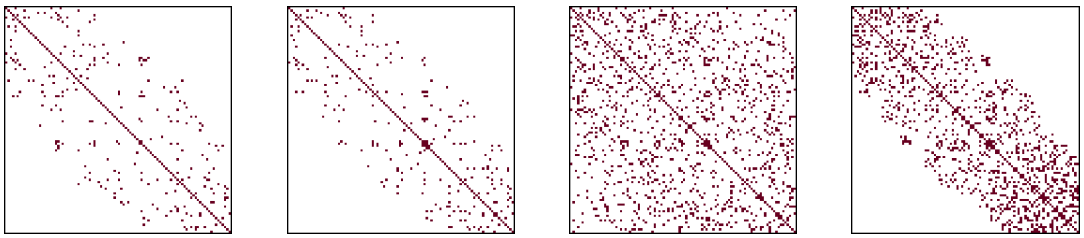
3. *Modified Graphical Lasso.* Given the fact that the graphical lasso cannot take advantage of the additional informaiton, we designed a modification for it. According to pathways obtained by (5), we manually extract samples in each pathway from X and get X_k. By using graphical lasso method on X_k, we get a group of submatrices Θ_k of the precision matrix. Then we combine those submatrices into a precision matrix of all nodes. The modification is specified in Algorithm 2.

[1] http://dblp.uni-trier.de
[2] http://dm1.cs.uiuc.edu/alumni.html

Algorithm 2 Modified graphical lasso

Input: Number of nodes n, sample set X, information of lifecycles $\{\mathcal{L}_i\}_{i=1}^{T}$, and proximity constrain parameter r.
Output: The estimated precision matrix Θ.

for $[i, \mathcal{L}_i]$ in **enumerate**$(\{\mathcal{L}_i\}_{i=1}^{T})$ **do**
  $X^{(i)} \leftarrow X[:][\bigcup_{|k-i|\leq r} \mathcal{L}_k]$ ▷ extract the sample of nodes in $\mathcal{L}_i$
  $\Theta^{(i)} \leftarrow$ **graphicalLasso**$(X^{(i)})$
end for
for every (v_j, v_k) **do** ▷ merge matrices $\Theta^{(i)}$
  if $\Theta_{jk}^{(i)}$ is non-zero, $\forall i, v_j, v_k \in \bigcup_{|l-i|\leq r} \mathcal{L}_l$ **then**
    Θ_{jk} is non-zero
  else
    $\Theta_{jk} \leftarrow 0$
  end if
end for
return Θ



(a) Ground Truth (b) Proposed Method (c) Graphical Lasso (d) Modified GLasso

Figure 2: Learned precision matrices of three methods. Every charts stands for a precision matrix, and red dots in them denotes a non-zero element in the model. There are 100 nodes in it and $T = 20$

5.3 Evaluation Indicators

We intend to illustrate the efficiency and effectiveness of our proposed method. The efficiency can be shown by running time. For effectiveness, we follow [7] and marks in [5] to define an F_1 score:

$$F_1 = \frac{2 \times n_d^2}{n_a n_d + n_g n_d}, \tag{6}$$

where n_d is the number of true edges detected by the algorithm, n_g is the number of edges in the true precision matrix, n_a is the total number of detected edges. The larger the value of F_1, the better the performance. We also use the indicator *Precision* P (not Θ, the precision matrix), defined by equation

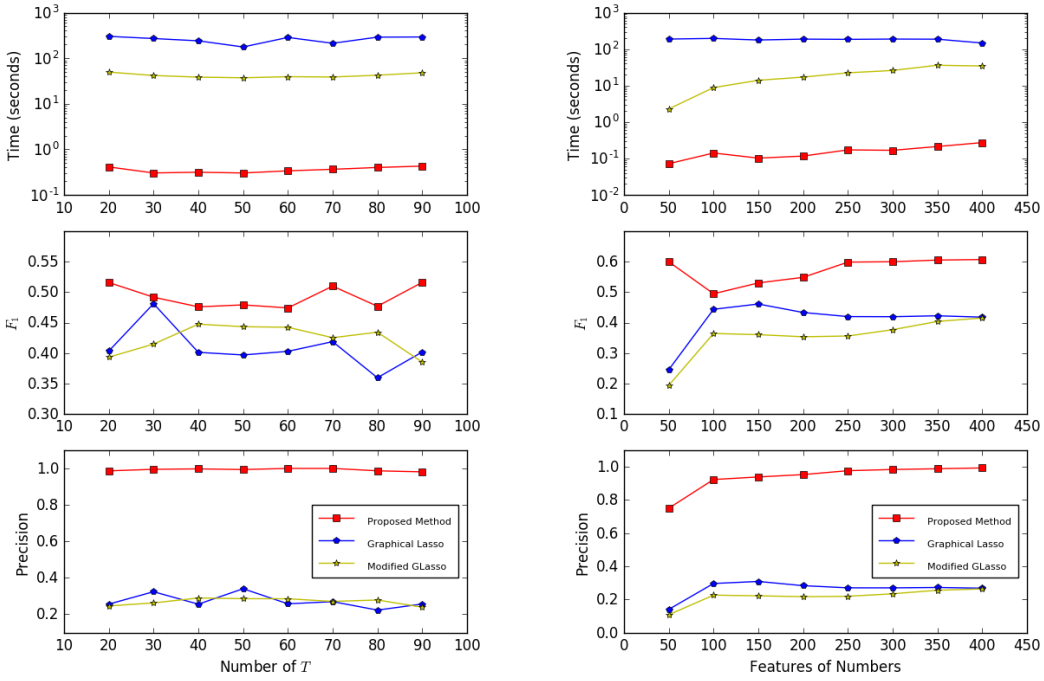
$$P = \frac{n_d}{n_a}, \tag{7}$$

where marks has the same meaning as above. Simply speaking, the precicion is the proportion of the truely detected edges in all edges the method detected.

5.4 Results of Experiments

We first conducted a simple experiments to illustrate the intutive comparison between different methods. As is shown in Figure 2, due to the lack of prior knowledge, graphical lasso returns a noisy result. Though some prior knowledge can be utilized, the modified graphical lasso still gives more non-zero elements. The proposed method of this paper has a precision matrix very close to the ground truth.

We then conducted two groups of experiments. In two groups respectively, we hold other parameters fixed, and change the total number of nodes and stages. We used the three indicators mentioned above



(a) (b)

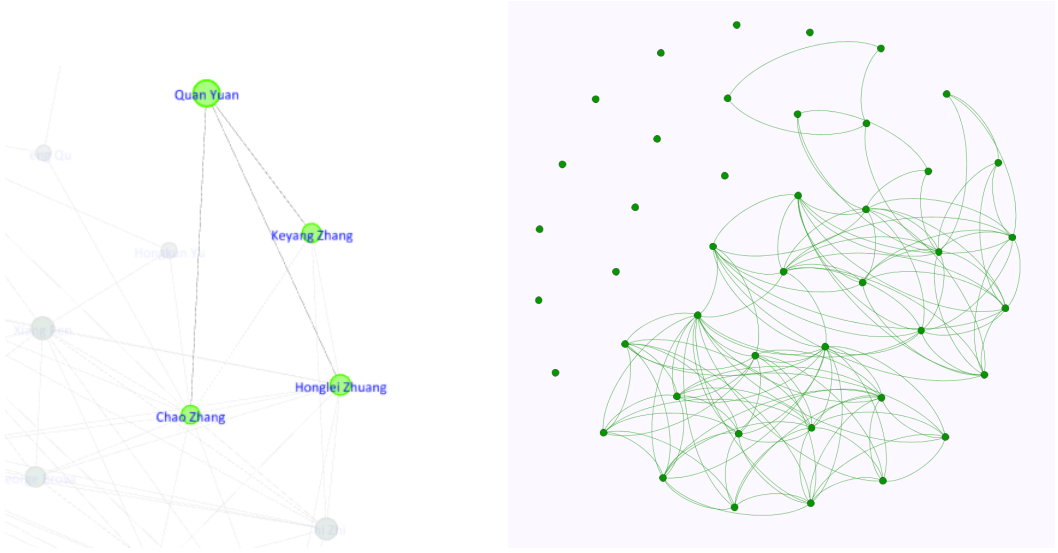
Figure 3: Effectiveness comparison. Charts from top to bottom respectively show running time, F_1 score and P. (a) shows indicators changing with T, the total number of stages. (b) shows indicators changing with the total number of nodes (features). In group (a), $r = 1$. In group (b), $r = 3$.

to evaluate the results, and come up with Figure 3. As is shown in 3, the running times are always smaller, and F_1 score and P are always larger. Besides, as is shown in 3, if the number of nodes held fixed, the number of stages T dosenot significantly effects the accuracy and efficiency.

Experiment conducted on real-world datasets, we used the DBLP subset mentioned above. We take the publication activites of students as samples for input of our proposed method. We get the learned matrix and graph, and then compare the graph with the DBLP database. Results show that the proposed method can give good results. As is shown in Figure 4b, Quan Yuan, a post-doc joined the group in 2015, with many students is in agreement with the proximity constrain; yet the method indicates only 3 of them have a close connection with him, which conforms with the DBLP datasets.

6 Conclusion

In this paper, we first introduced the Gaussian graphical model and the problem of estimation of the sparse graph. Then we introduced the definition of lifecycle and the 'proximity constrain'. With these definitions we proposed the fomulation of the estimation of sparse graph with lifecycle, and proved that such problem can be solved by adapting the pathway graphical lasso method. We run a series of experiments to prove the effectiveness of our proposed method. The results show that our method is more efficient and more accurate.



(a) Subgraph with Quan Yuan (b) the Whole Graph

Figure 4: The Learned Graph of the DBLP Subset. There are 42 nodes and 132 edges in the graph.

7 Prospect

As Section 5.1 of this paper discribed the DBLP, in real-world problems we often encounter with the situation that the lifecycles of nodes vary widely. In this case, it is possible that the students be admitted by the school in any day of a year, while we ignored the date and month, and classify the students by different years; students have different lengths of lifecycles, while we only take the proximity of the start stage of lifecycle as the criteria. The approach is reasonalbe in this case, yet other datasets (genome, proteome, etc.) may not use the same approach. We want to find a approach to normalize the lifecycles of different nodes, to 'unify' the lengths of lifecycles. Thus, we can have a universal method to solve all the sparse graph estimation problem with lifecycle.

Heterogeneous Information Network, or HIN has become a popular field of studying recently[6, 10, 11]. It is natural that in a network, there are different kinds of edges. Take DBLP as an example, relationships between scholars can be schoolmates, teacher and student, colleagues, etc. A information network with different kinds of edges is a HIN. Sparse graph estimation with lifecycle can be adapted to HIN. We hope that the result of this paper can be used to the researches on HIN in the future.

References

[1] Banerjee, Onureena, L. E. Ghaoui, and A. D'Aspremont. "Model Selection Through Sparse Maximum Likelihood Estimation." *Journal of Machine Learning Research* 9.3(2007):485-516.

[2] Friedman, J, T. Hastie, and R. Tibshirani. "Sparse inverse covariance estimation with the graphical lasso." *Biostatistics* 9.3(2008):432.

[3] Grechkin, Maxim, et al. "Pathway Graphical Lasso." *Twenty-ninth Aaai Conference on Artificial Intelligence* Proc Conf AAAI Artif Intell, 2015:2617.

[4] Whittaker, Joe. *Graphical Models in Applied Multivariate Statistics.* Wiley, 2009.

[5] Yang, Sen, et al. "Fused Multiple Graphical Lasso." *Siam Journal on Optimization* 25.2(2013).

[6] Kong, Xiangnan, et al. "Meta path-based collective classification in heterogeneous information networks." *ACM International Conference on Information and Knowledge Management* ACM, 2012:1567-1571.

[7] Yang, Yiming. "An Evaluation of Statistical Approaches to Text Categorization." *Information Retrieval Journal* 1.1(1999):69-90.

[8] Li, Lu, and K. C. Toh. "An inexact interior point method for l_1-regularized sparse covariance selection." *Mathematical Programming Computation* 2.3(2010):291-315.

[9] R. Mazumder and T. Hastie. "Exact covariance thresholding into connected components for large-scale graphical lasso." *JMLR* 13(1):781–794, 2012.

[10] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. "PathSim: Meta path-based top-k similarity search in heterogeneous information networks." *PVLDB* 4(11):992–1003, 2011.

[11] Zhang, Yao, et al. "Meta-Path Graphical Lasso for Learning Heterogeneous Connectivities." *SIAM International Conference on Data Mining* (2017)