Efficiently and Fast Learning a Fine-grained Stochastic Blockmodel from Large Networks

Xuehua Zhao[1,2], Bo Yang[1,2,⋆], and Hechang Chen[1,2]

[1] College of Computer Science and Technology, Jilin University, Changchun, China
[2] Key Laboratory of Symbolic Computation and Knowledge Engineer
Ministry of Education, Jilin University, Changchun, China
ybo@jlu.edu.cn

Abstract. Stochastic blockmodel (SBM) has recently come into the spotlight in the domains of social network analysis and statistical machine learning, as it enables us to decompose and then analyze an exploratory network without knowing any priori information about its intrinsic structure. However, the prohibitive computational cost limits SBM learning algorithm with the capability of model selection to small network with hundreds of nodes. This paper presents a fine-gained SBM and its fast learning algorithm, named FSL, which ingeniously combines the component-wise EM (CEM) algorithm and minimum message length (MML) together to achieve the parallel learning of parameter estimation and model evaluation. The FSL significantly reduces the time complexity of the learning algorithm, and scales to network with thousands of nodes. The experimental results indicate that the FSL can achieve the best tradeoff between effectiveness and efficiency through greatly reducing learning time while preserving competitive learning accuracy. Moreover, it is noteworthy that our proposed method shows its excellent generalization ability through the application of link prediction.

Keywords: Network data mining, Social network analysis, Stochastic blockmodel, Model selection, Link prediction.

1 Introduction

As an important statistical network model, stochastic blockmodel (SBM) [1] enables us to reasonably decompose and then properly analyze an exploratory network with zero prior information about its intrinsic structure. That is, we believe, the most attractive merit of the model. Formally, a standard SBM is defined as a triple $(K, \mathit{\Pi}, \mathit{\Omega})$, where K is the number of blocks, $\mathit{\Pi}$ is an $K \times K$ matrix in which π_{ql} denotes the probability that a link from a node in block q connects to a node in block l, and $\mathit{\Omega}$ is an K-dimension vector in which ω_k denotes the probability that a randomly chosen node falls in block k.

SBM is able to approximate any mixture patterns of assortative and disassortative structures ubiquitously demonstrated by the real-world networks, once

[⋆] Corresponding author.

V.S. Tseng et al. (Eds.): PAKDD 2014, Part I, LNAI 8443, pp. 374–385, 2014.
© Springer International Publishing Switzerland 2014

it is properly parameterized through fitting observed networks. Besides, SBM is used either as a generative model to synthesize the artificial networks containing assortative communities, disassortative multi-partites and arbitrary mixtures of them, or as a prediction model to predict missing and spurious links. Therefore, SBM has attracted much attention of researchers from the domains of statistics and machine learning since it was firstly proposed by Fienberg and Wasserman in 1981[1]. So far, various extensions of SBM have been proposed to address the oriented tasks of network analysis, such as, multiple roles SBM [2], overlapping SBM [3], mixture SBM [4], scale-free SBM [5], hierarchical SBM [6], among others.

Although SBM has demonstrated superiority in structure analysis, the intractable time complexity of learning severely limits the model to those applications just involving very tiny networks. For the current available learning algorithms, given the number of blocks, K, that means we take no account of model selection, the time complexity is at least $O(K^2n^2)$ where n denotes the number of the nodes. Otherwise, it will get much more, say $O(n^5)$. In practice, if one uses conventional computers, given K, the algorithms just efficiently deal with the networks with at most thousands of nodes, otherwise at most hundreds of nodes, far from the scales of most real-world networks we are interested in. For most of real-world networks, usually, we have no priori knowledge about them, in this case, only the SBM learning algorithms with model selection provide us with the real powerful tools since it can automatically determine the "true" number of blocks. However, the prohibitive time complexity hinders us from effectively analyzing large networks with thousands of nodes.

To address this issue, on one hand, we present a fine-gained SBM named FSBM, to capture more details of networks, and on the other hand, propose a corresponding fast learning algorithm with the capability of model selection called FSL. The FSL ingeniously combines the Component-wise EM(CEM) with Minimum Message Length(MML) to achieve simultaneous rather than alternative execution of model selection and parameter estimation, being able to smartly select a "good" model from a huge problem space consisting of entire candidate models with a significantly reduced computational cost. To the best of our knowledge, this is the first effort in literature to propose a parallel learning process of SBM.

The rest of this paper is organized as follows: Section 2 reviews the related works. Section 3 presents the fine-gained SBM and its learning method. Section 4 validates proposed models and algorithms as well as demonstrates their applications. Finally, Section 5 concludes this work by highlighting our contributions.

2 Related Work

SBM learning has two subtasks: to learn the parameters of model (Θ, Ω) and to determine the number of the blocks (K), corresponding to parameter estimation and model selection, respectively. Model selection aims at selecting a model with a good tradeoff between description precision and model complexity. As we know, the complexity of a model is determined by the number of parameters in the model. In this sense, the model selection of SBM is actually the determination of a reasonable K, the number of blocks, since the quantity of parameters of SBM can be represented as a function of K.

Most existing algorithms adopt EM or variational EM to estimate the parameters of SBM [2–4, 7, 8] in which SBM utilizes a latent variable Z to indicate the group to which a node belongs. The time complexity of such methods is at least $O(n^2K^2)$. Currently, the available model selection methods applied to SBM fall into three categories including cross-validation(CV), bayesian-based criteria and minimum description length(MDL). The CV has been rarely used in the SBM learning algorithm since its extremely high computation cost [4]. According to the adopted approximation techniques, BIC, ICL and variation-based approximate evidence are three main bayesian based criteria. Airoldi et al. [4] used BIC to select optimal multiple role SBM. Daudin et al. proposed the SICL algorithm which adopted the ICL to select model [9]. Hofman et al. proposed the VBMOD algorithm which adopted variation based approximate evidence for model selection [10]. Latouche et al. proposed SILvb algorithm which adopted ILvb, a newly model selection criterion [11], which has the best performance among the current criteria by now. The MDL is the criterion derived from the information theory and formally coincides with the BIC. Very recently, Yang et al. proposed the GSMDL algorithm that adopted the MDL for model selection [6].

For the above criteria except CV, both of them naturally require that corresponding learning algorithms have to adopt a serial learning strategy. That is to say, the learning process will include the following three steps: firstly, to estimate the parameters of each model in model space, then to evaluate the learned model by a predefined certain criterion, finally to select the model with the best evaluation value as the optimal model. Hence, such a serial learning strategy requires estimating and evaluating each model in the model space even if the models in question are “bad”. This leads to a extremely high time complexity since such strategy needs to estimate parameters not only for “good” models but for “bad” models. In general, the time complexity of the learning strategy, such as SICL [9], SILvb [11] and GSMDL [6], require at least $O(n^5)$.

Recently, Hofman and Wiggins proposed a VBMOD algorithm with time complexity $O(n^4)$ by reducing the quantity of parameters from K^2+K to $K+2$. So far, the VBMOD still remains the lowest time complexity among all available SBM learning algorithms with the capability of model selection. However, its low time complexity is obtained at the expense of flexibility. That is, the strategy of parameter reduction will restrict the VBMOD to deal with the networks with community structures.

Figueiredo et al. [12] ever applied the MML to gaussian mixture model, and their experiments showed that the accuracy of the MML outperforms that of the MDL/BIC, LEC or ICL criteria. Although their method was proposed towards to feature vector space and is not suitable for processing networks, the rationale behind it inspires us to propose a new SBM learning algorithm by designing an parallel integration of model selection and parameter estimation.

3 Model and Method

3.1 Fine-gained Stochastic Blockmodel

As the standard SBM only uses the block connection matrix to characterize the homogeneity and heterogeneity of links between nodes, it is difficult for SBM to capture more detailed structural information. To address this issue, we relax the parameter $\mathit{\Pi}$ into the two parameters $\mathit{\Theta}$ and $\mathit{\Delta}$, which respectively denote the connection probability from blocks to nodes and the connection probability from nodes to blocks. And based on the relaxation, we proposed the fine-gained stochastic blockmodel, FSBM.

Let $N = (V, E)$ be a directed and binary network where $V(N)$ denotes the set of nodes and $E(N)$ denotes the set of directed edges. Let $A_{n\times n}$ be the adjacency matrix of N where n denotes the number of nodes and $A_{ij} = 1$ if there exists an edge from the node i to the node j, otherwise $A_{ij} = 0$. In the case of undirected network, supposing there are two directed edges between nodes and $A_{ij}= A_{ji}$.

The FSBM is defined as $X = (K, Z, \mathit{\Omega}, \mathit{\Theta}, \mathit{\Delta})$, where, K is the number of the blocks, Z is one $n\times K$ matrix in which z_{ik} denotes the block k to which the node i belongs, $\mathit{\Omega}$ is one K-dimension vector in which ω_k denotes the probability that a randomly chosen node falls in block k, $\mathit{\Theta}$ is one $K \times n$ matrix in which θ_{kj} denotes the probability that a link from a particular node in block k connects to vertex j, $\mathit{\Delta}$ is one $K \times n$ matrix in which δ_{kj} denotes the probability that a link from node j connects to a particular vertex in block k. If N is undirected network, then $\mathit{\Theta} = \mathit{\Delta}$. The block matrix of the FSBM can be obtained as follows:

$$\mathit{\Pi}_1 = \mathit{\Theta} Z D^{-1}, \qquad \mathit{\Pi}_2 = \mathit{\Delta} Z D^{-1} \tag{1}$$

where $D = diag(n\mathit{\Omega})$. For undirected network, we have $\mathit{\Pi}_1 = \mathit{\Pi}_2 = \mathit{\Pi}$. Therefore, the standard SBM is a specific case of the FSBM.

The log-likelihood of the observed network N can be written as follows:

$$\log P(N|\mathit{\Omega}, \mathit{\Theta}, \mathit{\Delta}) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} (\prod\nolimits_{j=1}^{n} f(\theta_{kj}, A_{ij}) f(\delta_{kj}, A_{ji}))\omega_k \tag{2}$$

where, $f(x, y) = x^y(1 - x)^{(1-y)}$.

The log-likelihood for complete data can be written as follows:

$$\log P(N, Z|\mathit{\Omega}, \mathit{\Theta}, \mathit{\Delta}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} (\sum_{j=1}^{n} \log(f(\theta_{kj}, A_{ij}) f(\delta_{kj}, A_{ji})) + \log \omega_k) \tag{3}$$

3.2 Fast SBM Learning Method

The proposed fast SBM learning method, FSL, ingeniously combines the CEM algorithm [13] with the MML [12] together to achieve the parallel learning of parameter estimation and model selection. The FSL first obtains the estimated parameter value of one block by the CEM algorithm, then evaluates the block in terms of the MML. The bad block evaluated by the MML, that is the existence probability is zero, is directly annihilated and is not estimated in the next iteration. And so on, until the convergence. In the process, the sequential updating approach of the CEM algorithm and the MML directly evaluating one block provide the support of the parallel learning. Finally, parameters estimation and model selection are parallelly implemented in a time convergence process. In contrast to the serial learning algorithm, the FSL directly finds the "good" mode in the model space and effectively reduces the computational cost by preventing the algorithm estimating the parameters of the "bad" model.

Parameter Estimation. Although the CEM algorithm has almost the same time complexity as the EM algorithm, its faster convergence can reduce the time cost, but the major contribution is that the sequential updating approach of the CEM algorithm provides a smart framework for the parallel learning of parameter estimation and model selection. The CEM algorithm considers the decomposition of the parameter vector, and updates only one block at a time, letting the other parameters unchanged. The E-step and M-step are as follows:

E-step: Given the observed network N and h^{t-1} where h and t respectively denote the model parameters $(\mathit{\Omega}, \mathit{\Theta}, \mathit{\Delta})$ and the current iteration step, compute the conditional expectation of complete log-likelihood, i.e. Q function.

$$Q(h, h^{t-1}) = \sum_{i=1}^{n}\sum_{k=1}^{K}\gamma_{ik}\Big(\sum_{j=1}^{n}(\log f(\theta_{kj}, A_{ij}) + \log f(\delta_{kj}, A_{ji})) + \log \omega_k\Big) \tag{4}$$

where $\gamma_{ik} = E[z_{ik}]$ denotes the posteriori probability of block k to which the node i belongs according to the model h^{t-1}

M-step: Update the parameters of the current block according to Eq. 5-7 which are obtained by maximizing Eq. 4:

$$\omega_k{}^{(t)} = \frac{\sum_{i=1}^{n}\gamma_{ik}}{n} \tag{5}$$

$$\theta_{kj}{}^{(t)} = \frac{\sum_{i=1}^{n}A_{ij}\gamma_{ik}}{\sum_{i=1}^{n}\gamma_{ik}} \tag{6}$$

$$\delta_{kj}{}^{(t)} = \frac{\sum_{i=1}^{n}A_{ji}\gamma_{ik}}{\sum_{i=1}^{n}\gamma_{ik}} \tag{7}$$

where $k = (t./K_{max})$ +1, K_{max} is the number of the blocks and ./ denotes modulus operator. In the current step t, only the parameters of the k-th block are updated according to Eq. 5-7, respectively.

Model Selection. The MML is derived from information theory and the rational behind MML is that the shorter code the data has, the better the data generation model is. The particular MML criterion is as follows:

$$\widehat{h} = \arg\min_{h} \ell(-\log p(h) - \log p(N|h) + \frac{1}{2}\log|\mathbf{I}(h)| + \frac{c}{2}(1 + \log\frac{1}{12})) \tag{8}$$

where N denotes the observed network, h denotes the model parameters, c is dimension of h, $\mathbf{I}(h)$ is the information matrix and $|\mathbf{I}(h)|$ denotes its determinant.

Since FSBM contains the latent variable Z, the information matrix of FSBM can not be obtained analytically. We adopt the information matrix of the complete data log-likelihood, $\mathbf{I}_c(\Theta, \Delta)$, and assume the parameters of the blocks as a priori independent and also independent from the parameter ω. For each factor $p(\Theta_k, \Delta_k)$ and $p(\omega_1, ..., \omega_k)$, we adopt the noninformative Jeffrey' priori. Consequently, (8) becomes

$$\widehat{h} = \arg\min_{h} \ell(\frac{c}{2}\sum_{k=1}^{K}\log(\frac{n\omega_k}{12}) + \frac{K}{2}\log\frac{n}{12} + \frac{K(c+1)}{2} - \log p(N|h)) \tag{9}$$

where $\ell(*)$ is the cost function, K is the number of the blocks, c is the number of parameters specifying each block, h denotes the parameters (Ω, Θ, Δ).

When ω_k is zero, (9) will be nonsense, however, from the view of data coding, the parameters of zero-probability block do not contribute to coding-length [12]. Let k_{nz} is the number of non-zero probability blocks, the cost function becomes

$$\ell(h, N) = \frac{c}{2}\sum_{k:\omega_k>0}^{K}\log(\frac{n\omega_k}{12}) + \frac{k_{nz}}{2}\log\frac{n}{12} + \frac{k_{nz}(c+1)}{2} - \log p(N|h) \tag{10}$$

Minimizing the cost function (10) with respect to h constitutes the solutions of parameters estimated. As we can see, to minimize (10) with respect to θ, δ is the same as to minimize the -Q function since the terms except $-\log p(N|h)$ is dropped, and the difference is ω. The ω obtained by differentiating (10) with k_{nz} fixed is given as follows:

$$\widehat{\omega}_k^{(t)} = \frac{\max\left\{0, (\sum_{i=1}^{n}\gamma_{ik}) - \frac{c}{2}\right\}}{\sum_{j=1}^{K}\max\left\{0, (\sum_{i=1}^{n}\gamma_{ij}) - \frac{c}{2}\right\}} \tag{11}$$

where the γ_{ik} are given by the E-step, c is the dimension of parameters. Noting in (2) that any block for which $\widehat{\omega}_k = 0$ does not contribute to the log-likelihood.

We can see that (11) provides us an approach to evaluate the blocks, that is if $\widehat{\omega}_k^{(t)}$ is zero in the current step t, the block k is regarded as a "bad" block and may directly annihilate it. Based on the property of the MML and the sequentially updating of the CEM algorithm can commonly achieve the parallel learning of parameter estimation and model selection.

Algorithm Description and Complexity Analysis. A detailed pseudocode description of the FSL is listed in Table 1. After convergence of the CEM, there is no guarantee that a minimum of $\ell(h, N)$ has been found since the block annihilation in (11) does not consider the additional decrease in $\ell(h, N)$ caused by the decrease in k_{nz}. According to [12], we check if smaller values of $\ell(h, N)$ are achieved by setting to zero blocks that were not annihilated by (11). To this end, we simply annihilate the least probable block and rerun CEM until convergence.

Table 1. The FSL Algorithm

Algorithm 1. FSL

```
**X=FSL**$(N, K_{min}, K_{max})$
**Input:** $N$, $K_{min}$, $K_{max}$
**Output:** $X_{best}$
**Initial:** $\widehat{h}(0) \leftarrow \{\widehat{h}_1, ..., \widehat{h}_{k_{\max}}, \widehat{\omega}_1, ..., \widehat{\omega}_{k_{\max}}\}$ ; $t \leftarrow 0$; $k_{nz} \leftarrow k_{\max}$; $\ell_{\min} \leftarrow +\infty$; $\varepsilon$
$u_k^i \leftarrow p(N^{(i)}|\widehat{h}_k)$,for $k = 1, ..., k_{max}$ and $i = 1, ..., n$
while $K_{nz} \geq K_{\min}$ do
    repeat
    $t \leftarrow t + 1$
    for m=1 to $K_{\max}$ do
        $\gamma_k^i \leftarrow \widehat{\omega}_k u_k^{(i)} (\sum_{j=1}^{k_{\max}} \widehat{a}_j u_j^{(i)})^{-1}$ , for i=1,,n
        $\widehat{\omega}_k \leftarrow \max\{0, (\sum_{i=1}^{n} \gamma_k^{(i)} - \frac{c}{2})\} \times (\sum_{j=1}^{k} \max\{0, (\sum_{i=1}^{n} \gamma_k^( i) - \frac{c}{2})\})^{-1}$
        $\{\widehat{\omega}_1, ..., \widehat{\omega}_m\} \leftarrow \{\widehat{\omega}_1, ..., \widehat{\omega}_m\}(\sum_{m=1}^{K_{\max}} \widehat{\omega}_m)^{-1}$
        if $\widehat{\omega}_k > 0$ then
            $\widehat{h_k} \leftarrow \arg\max_{h_k} \log p(N, \gamma|h)$
            $u_k^i \leftarrow p(N^{(i)}|\widehat{h}_k)$
            else
            $K_{nz} \leftarrow K_{nz} - 1$
        end if
    end for
    $\widehat{h}(t) \leftarrow \{\widehat{\theta}_1, ..., \widehat{\theta}_{K_{\max}}, \widehat{\delta}_1, ..., \widehat{\delta}_{K_{\max}}, \widehat{\omega}_1, ..., \widehat{\omega}_{K_{\max}}\}$
    $\ell[\widehat{h}(t), N] \leftarrow \frac{c}{2} \sum_{k:\widehat{\omega}_k > 0} \log \frac{n\widehat{\omega}_k}{12} + \frac{K_{nz}}{2} \log \frac{n}{2} + \frac{K_{nz}c + K_{nz}}{2} - \sum_{i=1}^{n} \log \sum_{k=1}^{K} \widehat{\omega}_k u_k^{(i)}$
    until $\ell[\widehat{h}(t-1), N] - \ell[\widehat{h}(t), N] < \varepsilon$
    if $\ell[\widehat{h}(t), N] < \ell_{\min}$ then
        $\ell_{\min} \leftarrow \ell[\widehat{h}(t), N]$
        $\widehat{h}_{best} \leftarrow \widehat{h}(t)$
        $Z$=compute($\gamma$ )
        $Xbest$=$(Z, \widehat{h}_{best})$;
    end if
    $k^* \leftarrow \arg\min_k \{\widehat{\omega}_k > 0\}$ ,$\widehat{\omega}_{k^*} \leftarrow 0$ $K_{nz} \leftarrow K_{nz} - 1$
end while
```

For the FSL, calculating the posterior of the latent variable Z and estimating parameters (Θ, Δ, Ω) are time-consuming, which dominate the time complexity of the whole computing process. We can analyse the time complexity by the detailed algorithm steps in Table 1. Given K, the time complexity is $O(In^2K)$, where I is the iterative steps. When K is unknown, in the worst case, the time complexity is $O(In^4)$. Among the available SBM learning algorithms with model selection, the FSL has the same low time complexity as the VBMOD, far less than the other algorithms, such as GSMDL [6] $O(n^5)$, SICL [9] $O(n^5)$, SILvb [11] $O(n^5)$, MBIC[2] $O(n^7)$, Shen [7] $O(n^6)$ and so on. Noting that the FSL

Table 2. Confusion matrices of blocks detected in network with community

	(a) $Q_{true} \backslash Q_{FSL}$							(b) $Q_{true} \backslash Q_{GSMDL}$							(c) $Q_{true} \backslash Q_{VBMOD}$						
	2	3	4	5	6	7	8	2	3	4	5	6	7	8	2	3	4	5	6	7	8
3	0	**100**	0	0	0	0	0	0	**98**	1	1	0	0	0	0	**100**	0	0	0	0	0
4	0	0	**100**	0	0	0	0	0	0	**100**	0	0	0	0	0	0	**100**	0	0	0	0
5	0	0	0	**97**	3	0	0	0	0	0	**100**	0	0	0	0	0	0	**100**	0	0	0
6	0	0	0	1	**94**	5	0	0	0	0	7	**89**	4	0	0	0	0	0	**100**	0	0
7	0	0	0	0	27	**68**	5	11	89	0	30	55	**15**	0	0	0	0	6	82	**12**	0

	(d) $Q_{true} \backslash Q_{SICL}$							(e) $Q_{true} \backslash Q_{SILvb}$						
	2	3	4	5	6	7	8	2	3	4	5	6	7	8
3	0	**100**	0	0	0	0	0	0	**100**	0	0	0	0	0
4	0	0	**100**	0	0	0	0	0	0	**100**	0	0	0	0
5	0	0	0	**100**	0	0	0	0	0	0	**100**	0	0	0
6	0	0	0	23	**77**	0	0	0	0	0	0	**100**	0	0
7	0	5	27	45	23	**0**	0	0	0	0	2	15	**83**	1

can analyse the networks with various structures, such as bipartite, multipartite and mixture structure, but the VBMOD just analysing the networks with the community.

4 Validation

In this section, we validate the proposed algorithm on the synthetic networks and real-world networks, and make comparisons with the other algorithms with model selection, which are GSMDL [6], SICL [9], VBMOD [10] and SILvb [11], respectively. We also validate the generalization ability by the application of link prediction. The experiments are run on the computer with dual-core 2GH CPU and 4GB RAM, and all the programs are implemented by Matlab 2010b.

4.1 Validation on the Synthetic Networks and Real-world Networks

Accuracy Validation. We use the same testing method that mentioned in [11] to validate the FSL. The SBM is used as generation model to produce three types of the synthetic networks, which respectively contain community, bipartite and multipartite structure. Each type of network is divides into 5 groups in which the number of the true blocks Q_{true} is 3, 4, 5, 6 and 7, respectively. And each group contains 100 networks with 50 nodes which are randomly produced.

Table 2-3 respectively show the confusion matrices of the results detected in networks with community and mixture structure. The results listed in Table 2 indicate that the FSL and SILvb are the best algorithm among the five algorithms, especially, when Q_{true} is 7, they can correctly find out 68 and 83 blocks, respectively. The results listed in Table 3 indicate that the VBMOD fails to correctly find any network structure, and the others can effectively find out the number of blocks in networks, especially, when Q_{true} is 7, the SILvb still exhibits

Table 3. Confusion matrices of blocks detected in networks with Multipartite

	(a) $Q_{true}\backslash\ Q_{FSL}$							**(b)** $Q_{true}\backslash\ Q_{GSMDL}$							**(c)** $Q_{true}\backslash\ Q_{VBMOD}$						
	2	3	4	5	6	7	8	2	3	4	5	6	7	8	2	3	4	5	6	7	8
3	0	**100**	0	0	0	0	0	0	**100**	0	0	0	0	0	100	**0**	0	0	0	0	0
4	0	0	**100**	0	0	0	0	0	0	**100**	0	0	0	0	0	100	**0**	1	0	0	0
5	0	0	0	**100**	3	0	0	0	0	0	**100**	0	0	0	0	100	0	**0**	0	0	0
6	0	0	0	0	**96**	4	0	0	0	0	9	**90**	1	0	0	0	100	0	**0**	0	0
7	0	0	0	0	9	**71**	20	0	0	0	19	50	**31**	0	0	0	0	96	4	**0**	0
	(d) $Q_{true}\backslash\ Q_{SICL}$							**(e)** $Q_{true}\backslash\ Q_{SILvb}$													
	2	3	4	5	6	7	8	2	3	4	5	6	7	8							
3	0	**100**	0	0	0	0	0	0	**100**	0	0	0	0	0							
4	0	0	**100**	0	0	0	0	0	0	**100**	0	0	0	0							
5	0	0	0	**100**	0	0	0	0	0	0	**100**	0	0	0							
6	0	0	2	8	**92**	0	0	0	0	0	0	**100**	0	0							
7	1	0	8	36	49	**6**	0	0	0	0	0	21	**79**	1							

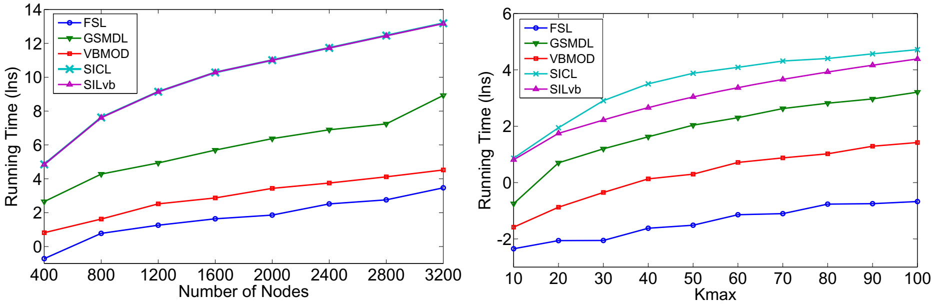
the best performance, the FSL following behind. Finally, we can conclude that the SILvb outperforms other existing learning algorithms in the accuracy test. The FSL is slightly worse than SILvb, but much better than other algorithms. The VBMOD fails to analyse the networks with the other structures except the community.

Time Complexity Validation To validate the time complexity, we use the running time metric to evaluate the time complexity of the algorithms.

First, we test the running time in synthetic networks with different size. The Newman model is used to generate synthetic networks, and the parameters of model are set as follows: K=4, d=16 and z_{out}=2. Let s in turn take value 100, 200, 300, 400, 500, 600, 700 and 800. Finally, we randomly generate eight groups of networks and each group contains 50 networks with the same size. K_{min} and K_{max} are respectively set 1 and 10.

Fig. 1(a) shows the results of the average running time of five algorithms. As we can see, the running time of the FSL is obviously much less than other four algorithms. The VBMOD closely follows behind the FSL, however, the VBMOD achieves its low time complexity at the cost of accuracy since its too few parameters, k+2, enable it to only analyse the networks with the community. Noting that when the number of nodes is 3200, on average the FSL only takes 32s, the VBMOD, MSMDL, SICL and SILvb take 92s, 7513s, 153901s and 153244s, respectively.

Then, we demonstrate how the scale of model space affects computational cost using the football network with 115 nodes [14]. We fix K_{min}=1 and let K_{max} in turn take the value 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100. Fig. 1(b) illustrates the relation of running time and the scale of model space. As we can see, the FSL significantly outperforms the other four algorithms.



(a) (b)

Fig. 1. The comparisons of running time of five algorithms. (a) shows the relation between time cost and network scale, (b) shows the relation between running time and K_{max}.

Table 4. The running time of the five algorithms(s)

Networks	FSL	GSMDL	VBMOD	SICL	SILvb
Karate	0.08	1.22	0.24	2.49	2.04
Dolphins	0.25	6.96	0.80	19.52	16.99
Polbooks	0.61	31.81	3.95	132.73	80.62
Jazz	3.59	605.57	36.77	1426.52	1006.53
Usair	57.83	7797.00	467.06	9429.49	8489.78
Metabolic	412.45	-	1567.97	-	-

4.2 Generalization Ability Validation

We first demonstrate the low computation cost of the FSL in the real-world networks, and use the learned parameters to test its generalization ability according to the performance of link prediction. In the experiment, we selected seven real-world networks, which respectively are Karate network [14], Dolphins network [15], Football network [14], Polbooks network (http://www.orgnet.com), Jazz network [17], Usair network [16] and Metabolic network [17].

Let K_{min}=1 and K_{max} take the number of nodes. The running time of the five algorithms is listed in Table 4. We can see that the running time of the FSL is significantly much lower than other four algorithms, the following is the VBMOD, and the SICL is the worst. Noting that the FSL only consumes 412.453s, but the VBMOD is 1567.97s in Metabolic network with 453 nodes, the running time of other algorithms are far more than that of the FSL, even fails to deal with the networks.

Then we evaluate the generalization ability of the models and algorithms in terms of the performance of link prediction. We also make comparisons with the CN algorithm [18] which is usually used as the baseline algorithm of link prediction. The AUC metric [18] is used to evaluate the performance of the algorithms. The SBM-based approaches to link prediction predict the missing links according to the learned parameters value related to connection probability.

Table 5. AUC value of the six algorithms

Networks	FSL	GSMDL	VBMOD	SICL	SILvb	CN
Karate	**0.946**±.034	0.836±.075	0.687±.072	0.752±.098	0.741±.084	0.680±.065
Dolphins	**0.953**±.015	0.735±.093	0.688±.047	0.697±.063	0.719±.052	0.793±.080
Polbooks	**0.935**±.012	0.883±.036	0.781±.036	0.864±.025	0.863±.023	0.894±.007
Jazz	0.953±.006	0.929±.006	0.770±.013	0.902±.016	0.920±.008	**0.954**±.002
Usair	**0.981**±.003	0.958±.006	0.828±.006	0.948±.007	0.961±.003	0.962±.006
Metabolic	**0.954**±.008	0.868±.013	0.651±.031	0.842±.013	0.861±.015	0.918±.006

We construct the data set by the following way: randomly pick the 10% of edges as test set from the network and the remaining 90% of edges as training set. For each network, we randomly sample 20 times and generate 20 groups of data.

The mean and standard deviation of AUC value of the prediction results are listed in Table 5. As we see, the results indicate that the FSL has the expected best performance among the six algorithms, and the performance of the VBMOD is the worst since the model only uses two parameters to describe the network structure so many information is missing. The main reasons are that the FSBM can capture the more detailed information of network structure than other SBMs, and the FSL can reasonably learn the FSBM.

5 Conclusions

In this paper, we proposed a fine-gained SBM and its fast learning algorithm with the capacity of model selection which adopts the parallel learning strategy to reduce the time complexity. To our best knowledge, this is the first time that the parallel learning strategy is proposed and applied to the SBM learning algorithm. We have validated the proposed learning algorithm on the synthetic networks and real-world networks. The results demonstrate that the proposed algorithm achieves the best tradeoff between effectiveness and efficiency through greatly reducing learning time while preserving competitive learning accuracy. In contrast to existing learning algorithms with model selection just dealing with networks with hundreds of nodes, the proposed algorithm can scale to the networks with thousands of nodes. Moreover, it is noteworthy that our proposed method demonstrates the excellent generalization ability with respect to link prediction.

Acknowledgments. This work was funded by the Program for New Century Excellent Talents in University under Grant NCET-11-0204, and the National Science Foundation of China under Grant Nos. 61133011, 61373053, 61300146, 61170092, and 61202308.

References

1. Holland, P., Laskey, K., Leinhardt, S.: Stochastic blockmodels: First steps. Social Networks 5(2), 109–137 (1983)
2. Airoldi, E., Blei, D., Fienberg, S., Xing, E.: Mixed membership stochastic blockmodels. The Journal of Machine Learning Research 9, 1981–2014 (2008)
3. Latouche, P., Birmel, E., Ambroise, C.: Overlapping stochastic block models with application to the French political blogosphere. The Annals of Applied Statistics 5(1), 309–336 (2011)
4. Newman, M., Leicht, E.: Mixture models and exploratory analysis in networks. Proceedings of the National Academy of Sciences of the United States of America 104(23), 9564–9569 (2007)
5. Karrer, B., Newman, M.: Stochastic blockmodels and community structure in networks. Physical Review E 83(1), 016107 (2011)
6. Yang, B., Liu, J., Liu, D.: Characterizing and Extracting Multiplex Patterns in Complex Networks. IEEE Transactions on Systems Man and Cybernetics, Part B-Cybernetics 42(2), 469–481 (2012)
7. Shen, H., Cheng, X., Guo, J.: Exploring the structural regularities in networks. Physical Review E 84(5), 056111 (2011)
8. Zhu, Y., Liu, D., Chen, G., Jia, H., Yu, H.: Mathematical modeling for active and dynamic diagnosis of crop diseases based on Bayesian networks and incremental learning. Mathematical and Computer Modelling 58(3), 514–523 (2013)
9. Daudin, J., Picard, F., Robin, S.: A mixture model for random graphs. Statistics and Computing 18(2), 173–183 (2008)
10. Hofman, J., Wiggins, C.: Bayesian approach to network modularity. Physical Review Letters 100(25), 258701 (2008)
11. Latouche, P., Birmele, E., Ambroise, C.: Variational Bayesian inference and complexity control for stochastic block models. Statistical Modelling 12(1), 93–115 (2012)
12. Figueiredo, M., Jain, A.: Unsupervised learning of finite mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(3), 381–396 (2002)
13. Celeux, G., Chretien, S., Forbes, F., Mkhadri, A.: A component-wise EM algorithm for mixtures. Journal of Computational and Graphical Statistics 10(4), 697–712 (2001)
14. Girvan, M., Newman, M.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America 99(12), 7821–7826 (2002)
15. Lusseau, D., Schneider, K., Boisseau, O., Haase, P., Slooten, E., Dawson, S.: The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations - Can geographic isolation explain this unique trait? Behavioral Ecology and Sociobiology 54(4), 396–405 (2003)
16. Batageli, V., Mrvar, A.: Pajek datasets, `http://vlado.fmf.uni-lj.si/pub/networks/data/default.htm`
17. Duch, J., Arenas, A.: Community detection in complex networks using extremal optimization. Physical Review E 72(2), 027104 (2005)
18. Lu, L., Zhou, T.: Link prediction in complex networks: A survey. Physica A-Statistical Mechanics and Its Applications 390(6), 1150–1170 (2011)