12-1-2013

Model Selection for Stochastic Block Models

Xiaoran Yan

Xiaoran Yan

Candidate

Computer Science

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Thomas Hayes , Chairperson

Cristopher Moore

Cosma Shalizi

Jared Saia

Terran Lane

Model Selection for Stochastic Block Models

by

Xiaoran Yan

B.S., Zhejiang University, China, 2007

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Computer Science

The University of New Mexico

Albuquerque, New Mexico

December, 2013

Dedication

To my parents, Hongxia and Jingtian, for their support and encouragement all the way across the Pacific.

"Everything should be kept as simple as possible, but no simpler" – Albert Einstein

Acknowledgments

I would like to thank my advisor, Dr. Cristopher Moore, for his excellent advisement and support throughout the years. Thanks also to all the other members of my diverse committee, including Dr. Cosma Shalizi for his statistical guidance, Dr. Thomas Hayes and Dr. Jared Saia for their theoretical discussions and Dr. Terran Lane for his machine learning expertise. I have several other people I would like to thank, as well[1].

This work was funded by the McDonnell Foundation, and by AFOSR and DARPA under grant FA9550-12-1-0432.

[1]Especially my colleague Yaojia Zhu.

Model Selection for Stochastic Block Models

by

Xiaoran Yan

B.S., Zhejiang University, China, 2007

Ph.D., Computer Science, University of New Mexico, 2013

Abstract

As a flexible representation for complex systems, networks (graphs) model entities and their interactions as nodes and edges. In many real-world networks, nodes divide naturally into *functional communities*, where nodes in the same group connect to the rest of the network in similar ways. Discovering such communities is an important part of modeling networks, as community structure offers clues to the processes which generated the graph. The *stochastic block model* is a popular network model based on community structures. It splits nodes into blocks, within which all nodes are stochastically equivalent in terms of how they connect to the rest of the network. As a generative model, it has a well-defined likelihood function with consistent parameter estimates. It is also highly flexible, capable of modeling a wide variety of community structures, including degree specific and overlapping communities.

Performance of different block models vary under different scenarios. Picking the right model is crucial for successful network modeling. A good model choice should balance the trade-off between complexity and fit. The task of model selection is

to automatically choose such a model given the data and the inference task. As a problem of wide interest, numerous statistical model selection techniques have been developed for classic independent data. Unfortunately, it has been a common mistake to use these techniques in block models without rigorous examinations of their derivations, ignoring the fact that some of the fundamental assumptions has been violated by moving into the domain of relational data sets such as networks.

In this dissertation, I thoroughly exam the literature of statistical model selection techniques, including both Frequentist and Bayesian approaches. My goal is to develop principled statistical model selection criteria for block models by adapting classic methods for network data. I do this by running bootstrapping simulations with an efficient algorithm, and correcting classic model selection theories for block models based on the simulation data. The new model selection methods are verified by both synthetic and real world data sets.

Contents

List of Figures

List of Tables

Glossary

$G(V, E)$	The graph G with the node set V and the edge set E
A_{uv}	The entry of the adjacency matrix specifying interactions between node u and v
SBM	Stochastic block model
$DC - SBM$	Degree-corrected block model
M_i	A candidate model
Π_i	The parameter set of M_i
Model selection	Automatically choose a model given the data and the inferences task
Order selection	The model selection problem of choosing the number of blocks
AIC	Akaike information criterion
BIC	Bayesian information criterion
LRT	Likelihood ratio test
MDL	Minimum description length

Chapter 1

Introduction

As a powerful representation for many complex systems, networks model entities and their interactions as nodes and edges. Food webs for example, have species as nodes, which are connected by edges representing predator-prey relationships. Another example would be computers and their network connections. With modern technology, an unprecedented amount of such relational data is available today, revolutionizing the way we study these complex systems. A new daunting challenge is how to extract useful information on this scale.

Different aspects of real world networks has been proposed and investigated over the years, like connectivity, degree distribution and so on [61, 33]. These measures are helping us to better understand the data on a seemingly intrackable scale. Among them, community detection has attracted much attention. It follows the canonical reductionist approach, dividing nodes into a hierarchy of categories, characterized by different patterns of connections in between. In online social networks, blogs tend to link to other blogs with similar political views [1]. In vertebrate food webs, predators tend to eat prey whose mass is smaller, but not too much smaller, than their own [25]. Networks of word adjacencies are correlated with those words' parts of speech [64].

In the Internet, different types of service providers form different kinds of links based on their capacities and business relationships [6, 29]—and so on.

Understanding these structures is crucial in deciphering these relational datasets. There has been a great deal of work on efficient algorithms for community detection in networks (see [27, 70] for reviews). However, most of this work defines a "community" as a group of nodes with high density of connections within the group and a low density of connections to the rest of the network. While this type of *assortative* community structure is generally the case in social networks, we are interested in a more general definition of *functional* community—a group of nodes that connect to the rest of the network in similar ways. A set of similar predators form a functional group in a food web, not because they eat each other, but because they feed on similar prey. In English, nouns often follow adjectives, but seldom follow other nouns. Even some social networks have disassortative structure, where nodes are more likely to be connected if they have different types. For example, some human societies are divided into moieties, and only allow marriages between different moieties [45].

The stochastic block model (SBM) provides a simple yet powerful solution [44, 85]. The basic SBM splits nodes into blocks, within which all nodes are stochastically equivalent in terms of how they connect to the rest of the network [86]. As a generative model, it has a well-defined likelihood function with consistent parameter estimates. It is also highly flexible, capable of modeling a wide variety of community structures with an arbitrary mixture of assortative and disassortative structure. It can also readily be extended to many other more elaborate probabilistic models—for instance, those where nodes belong to a mixture of classes [4], a hierarchy of classes and subclasses [21], or degree-corrected block models such as those in [48, 57, 65], which treat the nodes' degrees as parameters rather than data to be predicted. As a result, block models has been widely adopted to model networks in various disciplines (e.g. [10, 83, 37, 41, 7, 43, 74]).

The problem of *model selection* is to automatically choose a model given the data for a specific inferences tasks, among all models being considered [20]. With a good model that fits the task and data, learning of the parameters will be fast and inference will be accurate and noise tolerant. On the other hand, if you try to fit the data to a bad model, much effort will be wasted. For example, performance of different block models vary under different scenarios. A degree-corrected block model would be a good choice if members of the same communities have a wide degree distribution [48]. Even with the same block models, choosing different number of blocks (*order selection*) still leads to very different result [68].

The goal of model selection is to balance the trade-off between the model complexity and its fit to the data. Complex models with more parameters have a natural advantage at fitting data. Simpler models have lower variability, as result are less sensitive to noise in the data. A good model choice should hit the sweet spot, avoiding both over-fitting and under-fitting. In other words, we should only include additional parameters when they really matter. Excessive complexity not only increases the cost of the model, but also hurts the generalization performance [20].

This interest of finding the best model goes beyond statistics to science in general. Throughout history, it implicitly guided the development of many elegant models from observed data. The famous "Occam's razor" states "entities must not be multiplied beyond necessity". Albert Einstein put it as, "Everything should be kept as simple as possible, but no simpler". This "principle of parsimony" is at the heart of model selection.

One meta-framework of model selection that works for any classification model is generalization performance test. It achieves the balance by holding out a part of the data for generalization tests [14]. With network data, however, a single giant instance is usually all we have. An alternatives is to break the available data into sub-sets for multiple samples, which is tricky for graphs where strong correlations are

so prevalent [80]. Nonetheless, these meta-frameworks do provide general baselines for the purpose of comparison.

For models with proper likelihood functions like the block models, model selection can be approached using Frequentist or Bayesian statistical tools. In the classic Frequentist likelihood ratio tests (LRTs) [20, 77], we cast the model selection problem as a hypothesis testing between nested models. We reject the null model in favor of a more elaborate alternative when the likelihood ratio exceeds some threshold. This threshold, in turn, is determined by our desired error rate, and by the distribution of likelihood ratio under the null model. The famous *Akaike information criterion* (AIC) has its root in such Frequentist tests [20, 18].

Under the Bayesian framework, model selection is cast as an optimization problem. While simply maximizing the likelihood term leads to over-fitting, Bayesian approaches integrating over all parameters ensure a complexity-fit trade-off [42, 35, 19, 9, 56]. Based on it, the other popular *Bayesian information criterion* (BIC) are derived under various assumptions and approximations [20, 18, 35, 14]. They take the simple form of a penalized likelihood function just as the AIC. These information criteria offer efficient off-the-shelf methods for model selection on independent data, and are quite effective if used properly.

Unfortunately, it has been a common mistake to apply these information criteria without rigorous examinations of the underlying assumptions [71, 8, 81]. This is especially dangerous in the case of block models, as some of the fundamental assumptions have been violated. Little work has been done to lay down the theoretic foundation of model selection for the SBM and its various extensions, as [13] did for Markov chains. As a result, some employ these information criteria directly without knowing the consequences [7, 39], while others remain skeptical and use them only when no alternative is available [4, 3]. In this dissertation, I will investigate this issue and derive model selection algorithms for block models on a sound statistical

foundation.

1.1 Overview of the Chapters

In this dissertation, I will focus on the model selection problem for choosing between SBM and its degree-corrected cousin, as well as the important order selection problem of the number of blocks given the data and inference task.

In the second chapter, I will first introduce the idea of bias and variance trade-off in model selection problems. Then I will define related model selection information criteria that has been developed for independent data, as well as various block models. With the assumptions of information criteria explicitly listed, and the block models defined, I will investigate the mathematical compatibility between them in later chapters.

In the third chapter I shall introduce scalable inference algorithms for the models we consider, which enabled efficient and accurate experiments in the following chapters. This also include supervised learning for recommendation systems and an active learning framework for block models which can also be used to provide intuitions in our model selection study.

The fourth chapter shall focus on adapting classic Frequentist likelihood ratio tests into the realm of relational network data. I will investigate the model selection problem between pairs of nested models separately, laying down the statistical foundation for these basic situations with the full controllability of margins of error and confidence intervals. Investigations here shall lead to a corrected AIC for sparse networks.

The fifth chapter approaches the whole model selection problem in a single unified Bayesian framework, and establish its equivalence (under certain conditions) to

the minimum description length (MDL) principle [35]. As I will show, while the equivalence appears to be a mere mathematical coincidence, it has a much deeper and more intuitive connection in terms of information coding theories. By going through the Bayesian derivations, I shall propose the correct formulation of BIC for block models.

Chapter 2

Background and Related Work

In this chapter, I shall first introduce the idea of bias and variance trade-off in the setting of model selection. Based on this idea, I will define popular information criteria that was developed for independent data. By going through their derivations, it dose not only remind us of their underlying assumptions, but more importantly, pave the way for the more fundamental statistical analysis I shall employ in the later chapters. I shall then define various block models we will encounter in this dissertation. Notice the violation of many assumptions underlying traditional statistical tools. Finally, the chapter will be completed with a survey of existing model selection techniques for networks in the previous literature.

2.1 Bias and variance trade-off in model selection

For statistical models in general, errors in generalization tests can be decomposed into two main subcomponents: error due to "bias" and error due to "variance". The error due to bias is taken as the difference between the average prediction of

a model and the ground truth we are trying to predict. The error due to variance is taken as the variability of predictions from the same model around this average value [26]. We can visualize the interplay of bias and variance using the following bulls-eye diagram 2.1.

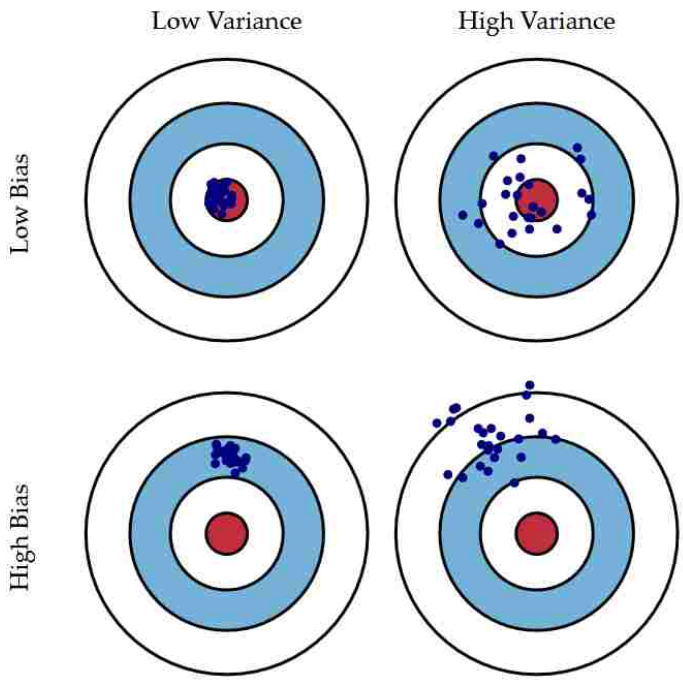


Figure 2.1: Graphical illustration of bias and variance
Imagine that the center of the target is the true model with perfect prediction. As we move away from the bulls-eye, the model gets worse and worse. Here we have built four candidate models, and fitted them to multiple training data sets with variability, forming distributions on the target. Each distribution corresponds to a model with different bias and variance, with individual hit represents a fitted instances for a specific training data set. Image courtesy of Fortmann–Roe, from the web essay [26].

Out of the four candidate models illustrated in Figure 2.1, the model with both low bias and low variance is the obvious choice in an ideal setting. In reality, however, we have no prior knowledge of the true model (the bulls-eye), nor can we ever hope to recover it exactly with finite and noisy data. With the best of both worlds out of

the question, we instead aim for a good model that hit the sweet spot in the middle, minimizing the the total error from both sources. The balance between bias and variance is thus fundamental in general statistical studies.

In the framework of model selection, this trade-off takes the form of balancing model complexity against its fit to the data [20]. In most practical situations, simpler models are more like the bottom left target in Figure 2.1. They have fewer parameters to estimate, leading to lower variance in the distribution of fitted instances. Since the model is most likely wrong, all fitted instances will have a systematic bias, leading to what is called under-fitting in machine learning. On the other hand, more complex models are represented by the top right target. With more parameters and thus a bigger hypothesis space, they usually have a much smaller modeling bias once fitted. The cost, however, is the larger variance and the risk of over-fitting to the noise in the training data.

From this perspective, model selection has become a optimization problem, with a target function composed of competing terms representing fit and complexity respectively. In the following section, I will introduce a class of popular model selection methods called information criteria, which are explicitly designed to follow this intuitive formulation.

2.2 Information criteria and their assumptions

Here we briefly derive two of the most popular information criteria (please refer to [20, 18] for more details). By explicitly listing their assumptions, we hope to identify problems that are preventing them from being directly applied on networks. They also serves as inspiration for two of the chapters later in this dissertation. There are many other information criteria in the literature, but most are derived from them with additional assumptions [20].

Information criteria in general have a rather simple and intuitive formulation:

$$XIC = -P(Y|M_i, \hat{\Pi}) + C(|\Pi|, n) \tag{2.1}$$

where Π is the parameter set and n is the number of independent data samples.

In this general formulation, the first term is the (negative) maximum likelihood of a model, with the second term measuring its model complexity. The complexity term usually is a function of the number of free parameter $|\Pi|$ and the sample size n. By minimizing this formula with explicit fit and complexity terms, the trade-off between bias and variance is achieved.

2.2.1 Akaike's information criterion

Originally named "an information criterion" (AIC) by Hirotsugu Akaike [5], AIC is the first general information criterion for model selection. Versatile and simple to use, it remains one of the most popular strategies until this day.

Founded in information theory, it is effectively a relative measure of the information lost when a given model is used to describe reality. If different model functions are denoted as $M_1, ..., M_m$, with the observed data $Y = \{y_1, ..., y_n\}$, this information lost can be calculated as the Kullback-Leibler divergence between a candidate M_i and the true model M^*,

$$\mathbb{D}_{KL}(P(M^*)||P(M_i)) = \int P(y|M^*) \ln \frac{P(y|M^*)}{P(y|M_i)} dy$$

$$= \int P(y|M^*) \ln P(y|M^*) dy - \int P(y|M^*) \ln P(y|M_i) dy \tag{2.2}$$

$$\approx \sum_j P(y_j|M^*) \ln P(y_j|M^*) - \sum_j P(y_j|M^*) \ln P(y_j|M_i) \tag{2.3}$$

Notice Kullback-Leibler divergence and its empirical approximation is based on the assumption:

Assumption A.1:	**The data generated by $P(y\|M^*)$ are independent, and by the law of large numbers, the empirical distribution formed by the observed data Y become a close estimate of the true distribution, as the number of samples $n \to \infty$.**

When comparing different models, the first term of (2.2) which is the entropy of the true model stays constant. The second term, a variable where the model specification of M_i dependent upon the observed data Y, is the relative measure we are actually interest in. If all models considered are parametric, they can be specified using $\hat{\Pi}$, the maximum likelihood estimator (MLE) of parameters given the data Y, and the expected value of the second term is:

$$E_{\hat{\Pi}}\left[\int P(y|M^*)\ln P(y|M_i)dy\right] = \int_{\hat{\Pi}}\left[\int P(y|M^*)\ln P(y|M_i,\hat{\Pi})dy\right]P(\hat{\Pi}|Y)d\hat{\Pi} \tag{2.4}$$

This expected relative KL divergence is essentially what AIC measures before any approximation.

If the true model M^* is indeed contained in the parametric class of M_i, we can minimize (2.2) or (2.4) to 0 with Π^*, called the least false parameter values. The MLE $\hat{\Pi}$ tends to Π^* in the limit of large sample, even if the true model M^* is outside M_i. If we expand the inner expectation around Π^*,

$$\begin{aligned}
&\int P(y|M^*)\ln P(y|M_i,\hat{\Pi})dy = E_{P(y|M^*)}\left[\ln P(y|M_i,\hat{\Pi})\right] \\
\approx &E_{P(y|M^*)}\left[\ln P(y|M_i,\Pi^*)\right] + [\hat{\Pi}-\Pi^*]^T E_{P(y|M^*)}\left[\left.\frac{\partial \ln P(y|M_i,\Pi)}{\partial \Pi}\right|_{\Pi^*}\right] \\
&+\frac{1}{2}[\hat{\Pi}-\Pi^*]^T E_{P(y|M^*)}\left[\left.\frac{\partial^2 \ln P(y|M_i,\Pi)}{\partial \Pi^2}\right|_{\Pi^*}\right][\hat{\Pi}-\Pi^*]
\end{aligned}$$

Now we take the outer expectation with respect to $\hat{\Pi}$,

$$\begin{aligned}
&E_{\hat{\Pi}}\left[E_{P(y|M^*)}\left[\ln P(y|M_i,\hat{\Pi})\right]\right] \\
\approx &E_{P(y|M^*)}\left[\ln P(y|M_i,\Pi^*)\right] - \frac{1}{2}E_{\hat{\Pi}}\left[[\hat{\Pi}-\Pi^*]^T E_{P(y|M^*)}[I(y|\Pi^*)][\hat{\Pi}-\Pi^*]\right] \quad (2.5)
\end{aligned}$$

where we denote the second derivative matrix evaluated at Π^* as $-I(y|\Pi^*)$. The first order term vanishes because Π^* is the minimizer, therefore $E_{P(y|M^*)}[\mu(y|\Pi^*)] = 0$, where we denote the fisrt derivative evaluated at Π^* as $\mu(y|\Pi^*)$.

(2.5) becomes:

$$\begin{aligned}
&E_{P(y|M^*)}\left[\ln P(y|M_i,\Pi^*)\right] - \frac{1}{2}E_{\hat{\Pi}}\left[\mathrm{tr}\left[J(y|\Pi^*)[\hat{\Pi}-\Pi^*][\hat{\Pi}-\Pi^*]^T\right]\right] \\
\approx &E_{P(y|M^*)}\left[\ln P(y|M_i,\Pi^*)\right] - \frac{1}{2}\,\mathrm{tr}\left[J(y|\Pi^*)E_{\hat{\Pi}}\left[[\hat{\Pi}-\Pi^*][\hat{\Pi}-\Pi^*]^T\right]\right] \\
\approx &E_{P(y|M^*)}\left[\ln P(y|M_i,\Pi^*)\right] - \frac{1}{2}\,\mathrm{tr}\left[J(y|\Pi^*)\Sigma\right]
\end{aligned}$$

If we do another Taylor expansion of $P(y|M_i,\Pi^*)$ around $\hat{\Pi}$,

$$\begin{aligned}
&E_{P(y|M^*)}\left[\ln P(y|M_i,\Pi^*)\right] - \frac{1}{2}\,\mathrm{tr}\left[J(y|\Pi^*)\Sigma\right] \\
\approx &E_{P(y|M^*)}\left[\ln P(y|M_i,\hat{\Pi}) + [\Pi^*-\hat{\Pi}]^T\frac{\partial \ln P(y|M_i,\hat{\Pi})}{\partial \Pi}\right. \\
&\left.+\frac{1}{2}[\Pi^*-\hat{\Pi}]^T\frac{\partial^2 \ln P(y|M_i,\hat{\Pi})}{\partial \Pi^2}[\Pi^*-\hat{\Pi}]\right] - \frac{1}{2}\,\mathrm{tr}\left[J(y|\Pi^*)\Sigma\right] \\
\approx &E_{P(y|M^*)}\left[\ln P(y|M_i,\hat{\Pi})\right] - \frac{1}{2}\,\mathrm{tr}\left[J(y|\hat{\Pi})\Sigma\right] - \frac{1}{2}\,\mathrm{tr}\left[J(y|\Pi^*)\Sigma\right]
\end{aligned}$$

Because $\hat{\Pi} \to \Pi^*$ in the limit of large sample size, putting everything together, we have:

$$E_{\hat{\Pi}}\left[\int P(y|M^*)\ln P(y|M_i,\hat{\Pi})dy\right] \approx \int P(y|M^*)\ln P(y|M_i,\hat{\Pi})dy - \mathrm{tr}\left[J(y|\Pi^*)\Sigma\right] \quad (2.6)$$

The inner integral in (2.6) is taken with respect to the true data generating process. In practice, we approximate it with the empirical distribution like we did in (2.3). Using $1/n$ for $P(y_j|M^*)$ and $\hat{\Pi}$ for Π^*, (2.6) becomes:

$$nE_{\hat{\Pi}}\left[\sum_j P(y_j|M^*)\ln P(y_j|M_i)\right] \approx \sum_j \ln P(y_j|M_i,\hat{\Pi}) - \mathrm{tr}\left[\frac{1}{n}\left[\sum_j I(y_i|\hat{\Pi})\right] n\Sigma\right] \tag{2.7}$$

where we multiplied both sides by n, so that the first term is now the maximum likelihood of the candidate model given data $\ln P(Y|M_i,\hat{\Pi})$. The second term, can be rewritten as

$$\begin{aligned}\mathrm{tr}\left[\frac{1}{n}\left[\sum_j I(y_i|\hat{\Pi})\right] n\Sigma\right] &\approx \mathrm{tr}\left[J(Y|\hat{\Pi})\frac{n}{n}J^{-1}(Y|\hat{\Pi})K(Y|\hat{\Pi})J^{-1}(Y|\hat{\Pi})\right] \\ &\approx \mathrm{tr}\,\mathbb{1} = |\Pi|\end{aligned}$$

where we approximated Σ with $n^{-1}J^{-1}(Y|\hat{\Pi})K(Y|\hat{\Pi})J^{-1}(Y|\hat{\Pi})$. Multiplying both sides of (2.7) by -2, we now have the standard AIC equation:

$$AIC(M_i) = -2\ln P(Y|M_i,\hat{\Pi}) + 2|\Pi| \tag{2.8}$$

In summary, this derivation of (2.8) assumes:

Assumption A.2:	**All models considered are simple parametric models. The parameters are twice differentiable.**

Assumption A.3:	**When $n \to \infty$, the MLE of the parameters $\hat{\Pi}$ tends to Π^*, as $\sqrt{n}(\hat{\Pi} - \Pi^*) \sim N_k(0, n^{-1}J^{-1}(Y|\hat{\Pi})K(Y|\hat{\Pi})J^{-1}(Y|\hat{\Pi}))$.**

2.2.2 Bayesian information criterion

Given the option of selecting a single model from multiple candidates, a "Bayesian" procedure would select the model with the maximum a posteriori (MAP) probability [78]. The posterior probability of a particular model M_i is by Bayes' theorem:

$$P(M_i|Y) = \frac{P(M_i)P(Y|M_i)}{P(Y)}$$

Since the data is constant, if we assume all models have uninformative or uniform priors, the posterior ends up proportional to the marginal likelihood:

$$\begin{aligned} P(M_i|Y) \propto P(Y|M_i) &= \int_{\Pi} P(Y|M_i,\Pi)P(\Pi|M_i)d\Pi \\ &= \int_{\Pi} exp(\frac{n}{n}\ln(P(Y|M_i,\Pi)P(\Pi|M_i)))d\Pi \end{aligned} \tag{2.9}$$

where Π stands for the set of parameters in M_i.

Assumption B.1:	**Prior knowledge is available for observed data (constant) and candidate models (uniform).**

Applying the Laplace approximation on the integral gives,

$$P(Y|M_i) \approx (\frac{2\pi}{n})^{|\Pi|/2} \left| \frac{\partial^2 1/n \ln P(Y|M_i,\Pi)P(\Pi|M_i)}{\partial^2 \Pi} \right|_{\hat{\Pi}}^{-1/2} e^{\ln(P(Y|M_i,\hat{\Pi})P(\hat{\Pi}|M_i))}$$

$$\ln P(Y|M_i) \approx \frac{|\Pi|}{2}(\ln(2\pi) - \ln n) + \ln P(Y|M_i,\hat{\Pi}) + \ln P(\hat{\Pi}|M_i) - \frac{1}{2}H_i(\hat{\Pi})$$

where $H_i(\hat{\Pi})$ denotes the the Hessian matrix evaluated at the MLE $\hat{\Pi}$, and $|\Pi|$ represents the number of parameters in model M_i. If n is the number of data points, we have only two terms scale with it. By ignoring the constant terms, and multiplying both sides by -2, we have the BIC score as:

$$BIC(M_i) = -2\ln P(Y|M_i,\hat{\Pi}) + |\Pi|\ln n \tag{2.10}$$

With the Bayesian approach, BIC introduces additional assumptions on data and model priors. On the other hand, all the parameters in the set Π^* are integrated

out, therefore the assumptions on their convergence to MLEs is no longer needed. Furthermore, the distribution of the data samples are no longer required to be i.i.d., as the joint distribution $P(Y|M_i, \Pi)$ is not factored into a product.

Assumption B.2:	**All models considered are simple parametric models. The parameters are twice differentiable.**

2.3 The stochastic block model and variants

Recall that the SBMs provides a simple yet flexible model for the task of community detection in networks [44, 85]. Here I will first introduce various kinds of block models we will study in this dissertation. On a higher level, if we treat each model as a node, they form a directed network in terms of model elaboration, as shown in figure 2.2.

For all variants of block models we will consider, I represent our network as an undirected graph without self-loops $G = (V, E)$. G has n nodes in the set V, m edges in the set E, and they can be specified by an adjacency matrix A where each entry A_{uv} indicates how the node pair $\{u, v\}$ are connected. I assume that there are k blocks of nodes, so that each node u has a block label $g(u) \in \{1, \ldots, k\}$. Here $n_s = |\{u \in V : g(u) = s\}|$ is the number of nodes in block s, and $m_{st} = |\{u < v \& (u, v) \in E : g(u) = s, g(v) = t\}|$ is the number of edges connecting between block s and block t, or twice that number if $s = t$.

Among all the different block models in the above figure, I will focus on the model selection problems of the SBM, the Poisson block model, the Degree-corrected block model and their order selection problems in this dissertation. The notion "block model" will be used as a general term for all these variants. I will now define these three models in the following subsections.

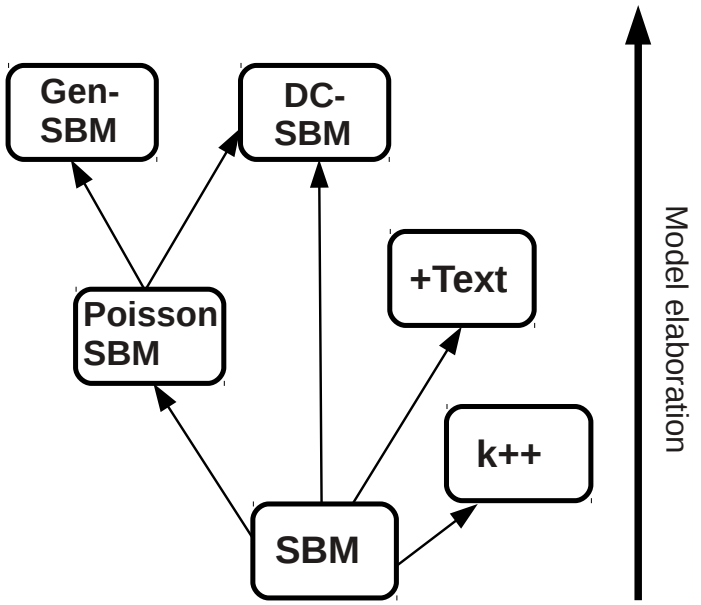


Figure 2.2: The network of complexity hierarchy of variants of block models
The model on the origin of an edge in this network is strictly a special case of the model on the target. They form a partial order with simpler models at the bottom. $k++$ means a SBM with more blocks, which can also be applied to all the other variants. For details of the +Text SBM, please refer to our paper [93]. Gen-SBM is defined in the paper [2].

2.3.1 The stochastic block model

I assume that G is generated by a SBM, or a "vanilla SBM" as I will often call it throughout this dissertation for distinction. For each pair of nodes u, v, there is an edge between u and v with the probability $p_{g(u),g(v)}$ specified by the $k \times k$ block affinity matrix p. Each node label $g(u)$ is first independently generated according to the prior probability $q_{g(u)}$ with $\sum_{s=0}^{k} q_s = 1$. Given a block assignment, i.e., a function $g : V \to \{1, \ldots, k\}$ assigning a label to each node, the probability of

generating a given graph G in this model is

$$\begin{aligned} P(G, g \mid q, p) &= \prod_u q_{g(u)} \left(\prod_{u<v,(u,v)\in E} p_{g(u)g(v)} \right) \left(\prod_{u<v,(u,v)\notin E} (1 - p_{g(u)g(v)}) \right) \\ &= \prod_{s=1}^{k} q_i^{n_s} \prod_{s,t=1}^{k} p_{st}^{m_{st}/2} (1 - p_{st})^{(n_s n_t - m_{st})/2} \\ &= P(V, g \mid q) \times P(E, g \mid p) \,. \end{aligned} \tag{2.11}$$

This likelihood factors into terms for nodes (first order) and edges (second order), conditioned on their parameters q, p respectively.

If we wish to model directed graphs, we can modify this expression by taking away the restriction $s \leq t$.

Take the log of (2.11), we have the log-likelihood

$$\begin{aligned} &\log P(G, g \mid q, p) \\ &= \sum_{s=1}^{k} n_s \log q_s + \frac{1}{2} \sum_{s,t=1}^{k} \left(m_{st} \log p_{st} + (n_s n_t - m_{st}) \log(1 - p_{st}) \right) \,. \end{aligned} \tag{2.12}$$

2.3.2 Poisson block model

As I have just shown, the vanilla SBM is for simple graphs, where each entry A_{uv} of the adjacency matrix is 0 or 1. Following e.g. [48], I propose a multi-graph generalization called the Poisson block model (Poisson-SBM), where the A_{uv} entries are now Poisson-distributed. According to the block assignment g, the model generates the number of edges A_{uv} between each pair of nodes u and v by making an independent Poisson draw. The means of these Poisson draws are specified by the $k \times k$ block affinity matrix ω (replacing), which replaces the p matrix in the vanilla SBM. Given

the block assignment g along with the data G, the likelihood would be

$$\begin{aligned} P(G, g \,|\, \omega, q) &= \prod_u q_{g_u} \prod_{u<v} \frac{\omega_{g_u g_v}^{A_{uv}} \mathrm{e}^{-\omega_{g_u g_v}}}{A_{uv}!} \\ &= \prod_{s=1}^{k} q_s^{n_s} \prod_{s,t=1}^{k} \omega_{st}^{m_{st}/2} \exp(-\frac{1}{2} n_s n_t \omega_{st}) \prod_{u<v} \frac{1}{A_{uv}!} \,. \end{aligned} \tag{2.13}$$

Here the last term is constant in the parameters, and is identically 1 for simple graphs, so we will discard it in what follows. The log-likelihood is then

$$\log P(G, g \,|\, \omega, q) = \sum_{s=1}^{k} n_s \log q_s + \frac{1}{2} \sum_{s,t=1}^{k} (m_{st} \log \omega_{st} - n_s n_t \omega_{st}) \,. \tag{2.14}$$

Compare it with (2.12), if $\omega_{st} = p_{st}$ and both are very close to 0, and thus $m_{st} \ll n_s n_t$,

$$\begin{aligned} \log P(G, g \,|\, q, p) &\approx \sum_{s=1}^{k} n_s \log q_s + \frac{1}{2} \sum_{s,t=1}^{k} (m_{st} \log p_{st} - (n_s n_t - m_{st}) p_{st}) \\ &\approx \log P(G, g \,|\, \omega, q) \,. \end{aligned}$$

In other words, when graph is very sparse, multi-edges are so rare that Poisson-SBM converge to the vanilla (Bernoulli) SBM. In the following chapters, we will often choose one of these two basic models for mathematical convenience in different situations.

2.3.3 Degree-corrected block model

For the above two block models I have introduced, any two nodes in the same block have the same degree distribution. Moreover, their degrees are sums of independent Poisson (Bernoulli) variables, so this distribution is also Poisson. As a consequence, these block models "resist" putting nodes with very different degrees in the same

block. This leads to problems with real networks where the degree distribution is highly skewed.

The degree-corrected block model(DC-SBM) addresses this problem by allowing heterogeneity of degree within blocks. Nodes are assigned to blocks as before, but each node also gets an additional parameter θ_u, which scales the expected number of edges connecting it to other nodes [48],

$$A_{uv}|g \sim \mathrm{Poi}(\theta_u \theta_v \omega_{g_u g_v})\,.$$

The parameter θ_u gives us a mean to explicitly model the expected degree of each node, which for instance, could be a measure of popularity in social networks. Since setting $\theta_u = 1$ for all u recovers the SBM, we say Poisson-SBM is *nested* inside the DC-SBM model, which is strictly more general.

The likelihood stays the same if we increase θ_u by some factor c for all nodes in block r, provided we also decrease ω_{st} for all s by the same factor. Thus identification demands a constraint, and a convenient one forces θ_u to sum to the total number of nodes within each block: $\sum_{u:g_u=s} \theta_u = n_s$. The complete-data likelihood of the DC-SBM model is then

$$\begin{aligned} P(G, g \,|\, \theta, \omega, q) &= \prod_u q_{g_u} \prod_{u<v} \frac{(\theta_u\, \theta_v\, \omega_{g_u g_v})^{A_{uv}}}{A_{uv}!} \exp(-\theta_u\, \theta_v\, \omega_{g_u g_v}) \\ &= \prod_u \theta_u^{d_u} \prod_{s=1}^{k} q_s^{n_s} \prod_{s,t=1}^{k} \omega_{st}^{m_{st}/2} \exp(-\frac{1}{2} n_s n_t \omega_{st}) \prod_{u<v} \frac{1}{A_{uv}!} \qquad (2.15) \\ &= P(\Theta, g \,|\, \theta) \times P(V, g \,|\, q) \times P(E, g \,|\, p)\,, \end{aligned}$$

where n_s and m_{st} are as before, and Θ is the factor containing all the θ parameters. Again ignoring the constant term, the log-likelihood is

$$\log P(G, g \,|\, \theta, \omega, q) = \sum_{s=1}^{k} n_s \log q_s + \sum_u d_u \log \theta_u + \frac{1}{2} \left(\sum_{s,t=1}^{k} m_{st} \log \omega_{st} - n_s n_t \omega_{st} \right). \qquad (2.16)$$

2.3.4 Block models for directed networks

So far we have defined the vanilla SBM, Poisson-SBM and the DC-SBM for undirected graphs. The directed counterparts actually have simpler mathematical forms with fewer constrains, thanks to their asymmetrical nature. Their likelihood functions are:

$$\begin{aligned} P(G, g \,|\, q, p) &= \prod_u q_{g(u)} \left(\prod_{(u,v)\in E} p_{g(u)g(v)} \right) \left(\prod_{(u,v)\notin E} (1 - p_{g(u)g(v)}) \right) \\ &= \prod_{s=1}^{k} q_i^{n_s} \prod_{s,t=1}^{k} p_{st}^{m_{st}} (1 - p_{st})^{n_s n_t - m_{st}} \,, \end{aligned} \tag{2.17}$$

$$\begin{aligned} P(G, g \,|\, \omega, q) &= \prod_u q_{g_u} \prod_{u,v} \frac{\omega_{g_u g_v}^{A_{uv}} \, \mathrm{e}^{-\omega_{g_u g_v}}}{A_{uv}!} \\ &= \prod_{s=1}^{k} q_s^{n_s} \prod_{s,t=1}^{k} \omega_{st}^{m_{st}} \, \exp(-n_s n_t \omega_{st}) \prod_{u<v} \frac{1}{A_{uv}!} \,, \end{aligned} \tag{2.18}$$

$$\begin{aligned} P(G, g \,|\, \theta, \omega, q) &= \prod_u q_{g_u} \prod_{u,v} \frac{(\theta_u \, \theta_v \, \omega_{g_u g_v})^{A_{uv}}}{A_{uv}!} \exp(-\theta_u \, \theta_v \, \omega_{g_u g_v}) \\ &= \prod_u \theta_u^{d_u} \prod_{s=1}^{k} q_s^{n_s} \prod_{s,t=1}^{k} \omega_{st}^{m_{st}} \exp(-n_s n_t \omega_{st}) \prod_{u<v} \frac{1}{A_{uv}!} \,, \end{aligned} \tag{2.19}$$

now with $m_{st} = |\{u < v \& (u,v) \in E : g(u) = s, g(v) = t\}|$ as the number of edges connecting between block s and block t, even if $s = t$.

For mathematical convenience, I shall be using the directed block models for the learning algorithms in the next chapter, although the experiments in later chapters are actually based on the undirected versions.

2.4 Existing related model selection methods

With the block models properly defined, I will do a brief literature review on related model selection methods beside information criteria to motivate my work in later chapters. These include machine learning meta-frameworks of both model driven and data driven varieties, as well as a method based on information coding theories.

Before I question the compatibility of these methods with block models, I want to first point out the biggest common pitfall in the network modeling community is to simply ignore the model selection problem all together. Increasingly complex models are built based on ad-hoc heuristics [61, 33]. The design choice in these models are often made out of personal preferences. Without proper statistical foundations, they are getting more and more susceptible to over-fitting as the model complexity keeps to grow.

A better alternative is to built/choose models based on domain knowledge. This is a popular approach for building network models in matured fields like biology and sociology, where plenty of domain knowledge is available from the previous literature [39, 87, 88]. These prior knowledge can help to reduce the potential space of candidate models, and confirm key hypothesis and observations. However, models based on such prior knowledge are at the same time constrained by it, incapable of exploring new discoveries from the data. Not to mention domain expertise are usually hard to come by.

2.4.1 Non-parametric models and generalization tests

Non-parametric models employ domain knowledge in a more model driven fashion. They can grow indefinitely in complexity, and the scaling is automatic based on the input data. Nonetheless, these models are severely restricted by the model family

they are defined in. For example the Chinese restaurant process is based on the conditional distribution of a Dirichlet mixture [69]. The same can be said about its numerous variants including the Indian buffet process [34]. To make the matter worse, the complexity scaling rules are usually designed for mathematical convenience rather than empirical evidence, making them not that practical for real data.

A more data driven framework of model selection is generalization performance test such as cross-validation. It achieves the balance between bias and variance by holding out a part of the data for generalization tests. In model selection problems, beside the usual training set and testing set, a separate validation set is used for tuning hyper parameters [14]. Many network problems like the community detection, however, has no ground truth to refer to. A counterpart for such clustering models is stability testing, which focus on a model's robustness to random noise to avoid over-fitting [84].

These methods work for any classification model provided that there is enough data sample. Unfortunately, we do not have this luxury for most networks, where a single giant instance is all that is available. An alternatives is to break the available data into sub-sets for multiple samples, which is tricky even in time-series data [72, 17], let alone for graphs where strong correlations is so prevalent [80]. Even if we manage to divide the data, repeated runs over the data pieces could lead to performance slowdowns. Nonetheless, these meta-frameworks do provide an general base line for the purpose of comparison.

2.4.2 The minimum description length principle

By compressing data with different codes, information scientists have long been working with the trade-off between complexity and fit. Because data compression is formally equivalent to a form of probabilistic prediction, searching for the model with

best predictive performance is essentially finding a coding scheme that lead to the minimum description length (MDL) [35, 74].

Under the MDL principle, the trade-off takes the from of balancing between the description length of the coding scheme and that of the message body given the code. When applied for block models, the description length of the coding scheme corresponds to the complexity of the SBM, which grows with the number of blocks as well as additional parameters such like the θ parameters in DC-SBM. The message body corresponds to the description length of the graph G given the SBM. It becomes shorter as the model gets more complex, since the data can be better fitted [75, 67, 66].

The minimum description length principle has a connection with Bayesian model selection in general [35]. This connection, while appear to be purely mathematical, actually has much deeper roots in carefully designed Bayesian codes. I will show how this can be done on block models in Chapter 5.

Chapter 3

Scalable Learning Algorithms

Before the statistical inquiry, I shall first explain the algorithms for learning the block models which make the experiments in later chapters possible. Beside traditional Monte Carlo sampling methods, I developed message passing algorithms for efficient estimation of the partition function. They fall into a more general framework of variational inference*, which provides a range of approximations with desirable balance between accuracy and scalability. Supervised learning from labeled data is another important task. Active learning goes one step further, querying pro-actively for the most informative label. I build an active learner for node community labels which uses information-theoretic measures gathered during the conditional Monte Carlo sampling algorithm at no additional cost.*

3.1 The partition function

In the previous chapter, I have defined likelihood functions for the various block models given a specific block assignment 2.3. However, the block assignment g is

usually not observed, but rather are what we most want to infer. We could try to infer g by maximizing the likelihood over the parameter set Π; in terms borrowed from statistical physics, this amounts to finding the *ground state* $\hat{g}$ that minimizes the *energy* (a connection I will explain later). When this $\hat{g}$ can be found, it recovers the correct g *exactly* if the graph is dense enough [10].

This approach, however, violates the assumptions of AIC and BIC we listed in the previous chapter 2.2.1 (assumptions A.2 and B.2). Thanks to the discrete nature of the latent states g, Taylor expansions on them are not possible, which is fundamental for deriving not only information criteria, but also the classic χ^2 likelihood ratio test we will see later. For latent state models in general, this maximum likelihood approach also runs a greater risk of over-fitting. Take an Erdős–Rényi random graph for instance, one can easily fit a specific block assignment g on top of it and get a much higher likelihood. These illusory block structures are actually capturing the random noise in this model, and therefore indications of over-fitting.

This is a common issue for any model with discrete latent variables. They cannot be simply ignored, since they are not fixed constants. A principled solution is to marginalize over them [28, 52, 9]. This is also a natural requirement for the Bayesian model selection in later chapters [20, 18, 14]. For block models, it means summing over all the possible group assignments. As an example, I will use the vanilla SBM for its simplicity, but the following result can be applied to any block model introduced in the previous chapter (see Section 2.3) .

$$P(G \,|\, q, p) = \sum_{g} P(G, g \,|\, q, p)\,. \tag{3.1}$$

In statistical physics terms the sum in (3.1) corresponds to the partition function $Z(\beta)$ at $\beta = 1$ of the Boltzmann distribution from which we sample the discrete group assignment variables. The probability density of $P(g \,|\, G, q, p)$ under the Boltzmann

distribution is given by:

$$P(g \mid G, q, p) = \frac{e^{\beta \ln P(G,g \mid q,p)}}{\sum_g e^{\beta \ln P(G,g \mid q,p)}}\,, \tag{3.2}$$

where the denominator is the partition function

$$Z(\beta) = \sum_g P(G, g \mid q, p)^\beta\,.$$

In statistical physics, the minus logarithm of a probability is often viewed as an energy function, as the exponents in equation (3.2). The minus logarithm of the maximum likelihood $P(G \mid \hat{g}, q, p)$ is called the ground state energy. The partition function $P(G \mid q, p)$, being a marginal probability itself, is connected with free energies. In fact, the vanilla SBM has a direct counterpart in the Ising model in statistical mechanics. These connections between probability theory and statistical physics will prove to be instrumental for designing and understanding the algorithms later in this chapter.

3.2 Calorimetry approximations for the partition function

With the partition function $P(G \mid q, p)$, the discrete states are summed over, we are now back in the continuous regime, problem solved. The sum, however, quickly becomes intractable since the state space of g explodes exponentially with the number of nodes in the graph. This is where the calorimetry trick from statistical physics comes to the rescue, providing us means to estimate the log partition function $\ln Z$ based on the free energy at various temperatures. There are several annealing techniques available for efficient sampling on these models.

3.2.1 Simulated annealing

Simulated annealing is inspired by the annealing technique in metallurgy, where controlled cooling in temperature leads to a decrease in the thermodynamic free energy.

$$\begin{aligned}\frac{\partial}{\partial\beta}\ln Z(\beta) &= \frac{\frac{\partial}{\partial\beta}Z(\beta)}{Z(\beta)} \\ &= \frac{\sum_g \ln P(G,g\,|\,q,p)P(G,g\,|\,q,p)^\beta}{Z(\beta)} \\ &= \sum_g \ln P(G,g\,|\,q,p)P^\beta(g\,|\,G,q,p) \\ &= \langle \ln P(G,g\,|\,q,p)\rangle_\beta \\ &= E_{q,p,\beta}[\ln P(G,g\,|\,q,p)]\,,\end{aligned}$$

where the last line simply indicates the expectation of $\ln P(G,g\,|\,q,p)$ over this Boltzmann distribution with inverse temperature β and parameters q,p. This value can be approximated using the average $\ln P(G,g\,|\,q,p)$ in a Monte Carlo sampling process.

We go through the derivative above so that we can estimate

$$\ln Z(1) = \ln Z(0) + \int_0^1 \frac{\partial}{\partial\beta}\ln Z(\beta)\partial\beta\,,$$

where $\ln Z(1)$ is the log partition function at temperature 1 of the distribution of discrete variables with fixed continuous variables β. It is also equal to the log marginal probability $\ln P(G\,|\,q,p)$. We approximate this integral numerically at q temperature points evenly distributed between $(0,1]$.

$$\int_0^1 \frac{\partial}{\partial\beta}\ln Z(\beta)\partial\beta = \sum_{\beta=t_1}^{t_q=1} \frac{\partial}{\partial\beta}\ln Z(\beta)\times\frac{1}{q}\,.$$

We employ Simulated Annealing to speed up the convergence of the sampler. It starts the system at the highest temperature with $\beta = 0$, and gradually cools down

until we have $\beta = 1$, allowing data collection at each temperature point. To avoid traps of local minima, we employ multiple replicas of MC chain at each temperature, and each is inherited into lower temperatures independently.

3.2.2 Population annealing

Population Annealing differs from Simulated Annealing in that new replicas in lower temperatures β' are not inherited independently, but instead with probability proportional to its $P(G, g \,|\, q, p)^{(\beta'-\beta)}$ [54]. Assuming the population of replicas is R, the expected number of copies of replica r that appear in the re-sampled population at β' is

$$\rho_r(\beta') = \frac{R \times E^r_{q,p,\beta}[P(G, g \,|\, q, p)^{(\beta'-\beta)}]}{\sum_{all\ r} E^r_{q,p,\beta}[P(G, g \,|\, q, p)^{(\beta'-\beta)}]}$$

The normalizing term above is actually the ratio of the partition function at neighboring temperatures:

$$\begin{aligned}
\frac{Z(\beta')}{Z(\beta)} &= \frac{\sum_g P(G, g \,|\, q, p)^{\beta'}}{Z(\beta)} \\
&= \sum_g P(G, g \,|\, q, p)^{(\beta'-\beta)} \frac{P(G, g \,|\, q, p)^{\beta}}{Z(\beta)} \\
&= E_{q,p,\beta}[P(G, g \,|\, q, p)^{(\beta'-\beta)}] \\
&= \frac{1}{R} \sum_{all\ r} E^r_{q,p,\beta}[P(G, g \,|\, q, p)^{(\beta'-\beta)}] \,.
\end{aligned}$$

This enable us to estimate $\ln Z(1)$ as

$$\ln Z(1) = \ln Z(0) + \sum_{\beta=t_0=0}^{t_q=1} \ln \left(\frac{Z(\beta')}{Z(\beta)} \right) . \tag{3.3}$$

Besides the above two calorimetry methods, there are even more sophisticated algorithms like parallel tempering. Unfortunately, all of them suffer from poor scalability. As the network grows, the convergence of the Monte Carlo sampling algorithm

becomes increasingly slow, and the number of samples required also skyrockets for accurate estimation. Since these algorithms are slow but exact, at least if the sampling algorithm converges and the spacing of temperatures is small enough, we use them as a correctness test for the more efficient algorithms we discuss next.

3.3 Variational approximations for the partition function

Variational methods are a family of techniques for approximating intractable sums just like the partition function. An alternative to Monte Carlo sampling methods mentioned previously, variational methods use a simpler *variational distribution* $Q(g)$ to approximate the intractable Boltzmann distribution.

$$\begin{aligned}\log P(G \mid q,p) =& \log \sum_g Q(g) \frac{P(G,g \mid q,p)}{Q(g)} \\ =& \sum_g \log \left[Q(g) \frac{P(G,g \mid q,p)}{Q(g)} \right] + \mathbb{D}_{KL}(Q \parallel P^*) \\ =& \mathbb{E}_{Q(g)} \left[\log \frac{P(G,g \mid q,p)}{Q(g)} \right] + \mathbb{D}_{KL}(Q \parallel P^*) \,. \end{aligned} \tag{3.4}$$

where the Kullback-Leibler divergence would be zero if the "variational distribution" $Q(g)$ was exactly the same as the true Boltzmann distribution P^*. Since $\mathbb{D}_{KL}(Q \parallel P^*)$ is always positive, we can rewrite the above equality in the form of Jensen's inequality,

$$\begin{aligned}\log P(G \mid q,p) =& \log \sum_g Q(g) \frac{P(G,g \mid q,p)}{Q(g)} \\ \geq& \sum_g \log \left[Q(g) \frac{P(G,g \mid q,p)}{Q(g)} \right] \\ =& \mathbb{E}_{Q(g)} \left[\log P(G,g \mid q,p) \right] + \mathbb{S}[Q(g)] \\ =& - \langle \mathbb{E} \rangle + \mathbb{S} \,. \end{aligned} \tag{3.5}$$

Now the approximation for the partition function has become an optimization problem:

$$\log P(G \mid q, p) \approx \sup_{Q} \left[\mathbb{E}_{Q(g)}[\log P(G, g \mid q, p)] + \mathbb{S}[Q(g)] \right] . \tag{3.6}$$

The key design problem of a variational approximation is the choice of the variational distribution $Q(g)$. We want $Q(g)$ to be simple in form and thus easy to infer, yet we need it to be complex enough to make an accurate approximation. Given the form of the likelihood function $\log P(G, g \mid q, p)$, a very desirable property of $Q(g)$ is to be able to factor it into terms which can be locally optimized. One such family of distributions are the "cluster variational distributions" from the field of statistical physics.

3.3.1 Kikuchi approximations and local free energies

In statistical physics, the formula $\langle \mathbb{E} \rangle - \mathbb{S}$, which is exactly the minus of what we are trying to maximize, is called the Gibbs free energy. The Kikuchi method, otherwise known as the "cluster variational method", approximates the Gibbs free energy as a sum of local free energies [89]. The first order Kikuchi approximation, also known as the mean-field approximation, assumes that $Q(g)$ is a product distribution, with single-site marginals at each node.

For each node u and type s, define b_u^s as the marginal belief that node u is of type s. They should obey the normalization conditions $\sum_s b_s^u = 1$. It follows that the two-node beliefs are simply $b_{st}^{uv} = b_s^u \times b_t^v$. Now we can define the local free energy involving a single node u as:

$$\mathbb{F}_{Kikuchi}^{u} = \sum_{t} b_t^u (\ln b_t^u - \ln q_t) ,$$

and the local free energy involving an edge/non-edge correlation (u, v) (excluding

node u and v) as:

$$\mathbb{F}^{uv}_{Kikuchi} = \sum_{st} b^{uv}_{st}(\ln b^{uv}_{st} - \ln f(s,t)_{uv}) \,,$$

where the function of $f(s,t)_{u,v}$ depends on the model. For mathematical convenience in the following subsection, I shall first use a directed vanilla SBM:

$$f(s,t)_{u,v} = \begin{cases} p(st)p(ts) & (u,v) \in E, (v,u) \in E \\ p(st)(1-p(ts)) & (u,v) \in E, (v,u) \notin E \\ (1-p(st))p(st) & (u,v) \notin E, (v,u) \in E \\ (1-p(st))(1-p(ts)) & (u,v) \notin E, (v,u) \notin E \end{cases}$$

With a particularly simple form of the joint belief:

$$B_{MF}(g \,|\, G, q, p) = \prod_{u} (b^{u}_{g(u)}) \,,$$

we have the mean-field free energy:

$$\begin{aligned} \mathbb{F}_{MF} &= \sum_{u} \mathbb{F}^{u}_{MF} + \sum_{u \neq v} \mathbb{F}^{uv}_{MF} \\ &= \sum_{u} \sum_{s} \left[b^{u}_{s}(\ln b^{u}_{s} - \ln q_s) + \sum_{v<u} \sum_{t} b^{uv}_{st}(\ln b^{uv}_{st} - \ln f(s,t)) \right] \\ &= \sum_{u} \sum_{s} \left[b^{u}_{s} \ln b^{u}_{s} + b^{u}_{s} \sum_{v<u} \sum_{t} b^{v}_{t} \ln(b^{u}_{s} b^{v}_{t}) - b^{u}_{s}(\ln q_s + \sum_{v<u} \sum_{t} b^{v}_{t} \ln f(s,t)) \right] \\ &= -\mathbb{S}_{MF} - \sum_{g} B_{MF}(g \,|\, G, q, p) \ln \left[q_s \sum_{v<u} \sum_{t} b^{v}_{t} f(s,t) \right] . \end{aligned} \tag{3.7}$$

While the first order mean-field approximation is simple and fast [9], it does not work well for block models where the correlations (edges) are important.

The second order Kikuchi approximation, which expands the range of local belief to pairs of nodes, is the Bethe free energy.

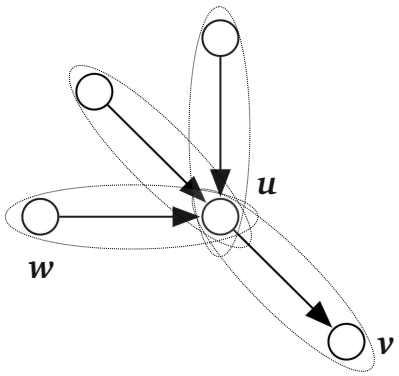


Figure 3.1: Regions of local energies in Bethe approximation
Notice that each node is overlapped d times, where d is its degree.

The Bethe free energy approximates the Gibbs free energy using one-node beliefs b_s^u, as well as two-node beliefs b_{st}^{uv} [90]. For each pair of vertices u, v and pair of types s, t, define b_{st}^{uv} as the pairwise marginal belief that vertices u and v are of type s and t respectively. They should obey the marginalization conditions $\sum_t b_{st}^{uv} = b_s^u$. The Bethe estimate of the joint belief is

$$B_{Bethe}(g \,|\, G, q, p) = \frac{\prod_{(u,v)\in E} b_{st}^{uv}}{\prod_u (b_s^u)^{d_u - 1}} \,. \tag{3.8}$$

For block models on simple graphs, the Bethe estimate of average energy is exact:

$$\begin{aligned} \langle \mathbb{E}_{Bethe} \rangle &= -\sum_g B_{Bethe}(g \,|\, G, q, p) \ln P(G, g \,|\, q, p) \\ &= -E_{B_{Bethe}(g \,|\, G, q, p)} \left[\sum_{u=1}^{n} \ln q_{g(u)} + \sum_{u \neq v} \ln f(s, t) \right] \\ &= -\sum_{u=1}^{n} \sum_s b_s^u \ln q_s - \sum_{u \neq v} \sum_{st} b_{st}^{uv} \ln f(s, t) \,. \end{aligned} \tag{3.9}$$

The Bethe estimate of the entropy, on the other hand, is only exact when the

graph is a tree.

$$\begin{aligned}
-\mathbb{S}_{Bethe} &= \sum_{g} B_{Bethe}(g \,|\, G, q, p) \ln B_{Bethe}(g \,|\, G, q, p) \\
&= E_{B_{Bethe}(g \,|\, G,q,p)} \left[\sum_{u \neq v} \ln b_{st}^{uv} - \sum_{u} \ln (b_s^u)^{d_u - 1} \right] \\
&= \sum_{u \neq v} \sum_{st} b_{st}^{uv} \ln b_{st}^{uv} - \sum_{u=1}^{n} (d_u - 1) \sum_{s} b_s^u \ln (b_s^u) \,. \qquad (3.10)
\end{aligned}$$

Putting both together, with some rearrangement,

$$\begin{aligned}
\mathbb{F}_{Bethe} =& \langle \mathbb{E}_{Bethe} \rangle - \mathbb{S}_{Bethe} \\
=& \sum_{u \neq v} \sum_{st} b_{st}^{uv} (\ln b_{st}^{uv} - \ln f(s,t)) - \sum_{u} \sum_{s} b_s^u (\ln (b_s^u)^{d_u - 1} + \ln q_s) \\
=& \sum_{u \neq v} \sum_{st} b_{st}^{uv} (\ln b_{st}^{uv} - \ln f(s,t) - \ln q_s - \ln q_t) \\
& - \sum_{u} (d_u - 1) \sum_{s} b_s^u (\ln b_s^u - \ln q_s) \\
=& \sum_{u \neq v} \mathbb{F}_{Bethe}^{uv+u+v} - \sum_{u} (d_u - 1) \mathbb{F}_{Bethe}^{u} \,. \qquad (3.11)
\end{aligned}$$

As we can see, higher order Kikuchi methods follows the inclusion–exclusion principle when summing up the local free energy components (see Figure 3.1 for the second order case). In general, the accuracy of Kikuchi methods improves as the order increases, and it is exact when the largest local component becomes the largest clique in the graph. When going for even higher order, the inclusion–exclusion equation might look very complicated, but everything boils down to including each local components exactly once, like (3.7) and (3.11) for block models.

Through the connection between probability and energy, these Kikuchi free energy formulations provide a range of candidates for the variational distribution $Q(g)$, forming a hierarchy of approximations with various scalability and accuracy. In particular, the second order Bethe approximation, with its exact average energy estima-

tion, turns out to hit the sweet point of balance for learning block models [23, 58]. As a result, we shall have the following variational distribution for the rest of the dissertation:

$$Q(g) = B_{Bethe}(g \,|\, G, q, p) = \frac{\prod_{u \neq v} b_{st}^{uv}}{\prod_u (b_s^u)^{n-2}}\,, \tag{3.12}$$

where we have plugged in $n-1$ as the degree for every node because all pairs of nodes in the block models are explicitly modeled, no matter if there is an edge or not. While this complete interaction graph might looks very far away from the tree condition for the Bethe approximation to be exact, empirical results will nevertheless validate its accuracy.

Now the approximation of the log partition function $\ln Z$ becomes an optimization problem of the Bethe free energy instead. In the paper [90], the authors proved that a message passing algorithm called Belief Propagation converges to the same fixed points as the Bethe free energy minimization process. It makes a scalable implementation with great parallelism potential possible, when analytical minimization is difficult. In statistical physics, it is called the cavity method and has already been applied to block models [24, 23].

3.3.2 The Belief Propagation algorithm

In this subsection I will describe the Belief Propagation (BP) algorithm for minimizing the Bethe free energy. To keep the mathematical notations simple, I will be building an algorithm for directed vanilla SBMs. Another version for undirected DC-SBMs shall be introduced in the next section. With these examples, readers should be able to generalize it to the other cases.

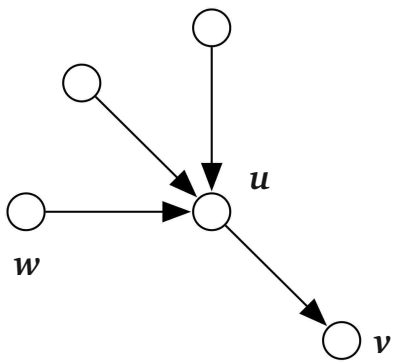


Figure 3.2: Belief Propagation on Figure 3.1
The message from u to v is based on the messages that u receives from its other neighbors like w.

The idea of belief propagation is each vertex u sends a "message" to each of its neighbors v, consisting of the marginal distribution that u would have if v were not in the network. We denote this $\mu_t^{u\to v}$, the probability that u would be of type t if v were absent. We update $\mu^{u\to v}$ according to the messages that u receives from its *other* neighbors w.

Finally, we assume that these neighbors are independent when conditioned on the label of u. In other words, we ignore the effect of paths that don't go through u. This assumption holds, just like the exact condition for the Bethe approximation, when the graph is a tree. If the graph is locally treelike and correlations decay, then it will hold approximately. In the rare event of double edges occurring in both directions for a pair of vertices, I assume that they are independent events.

In the vanilla SBM, where an edge from type s to type t exists with probability p_{st}, we have the following update rule:

$$\mu_t^{u\to v} = \frac{\xi_t^{u\to v}}{\sum_{t'=1}^{k} \xi_{t'}^{u\to v}},$$

where the numerator $\xi_t^{u\to v}$ is just the un-normalized versions of the $\mu_t^{u\to v}$:

$$\xi_t^{u\to v} = q_t \left(\prod_{\substack{w:(w,u)\in E \\ w:(u,w)\in E \\ w\neq u,v}} \sum_{s=1}^{k} \mu_s^{w\to u} p_{st} p_{ts} \right) \left(\prod_{\substack{w:(w,u)\notin E \\ w:(u,w)\in E \\ w\neq u,v}} \sum_{s=1}^{k} \mu_s^{w\to u} (1-p_{st}) p_{ts} \right) \left(\prod_{\substack{w:(w,u)\in E \\ w:(u,w)\notin E \\ w\neq u,v}} \sum_{s=1}^{k} \mu_s^{w\to u} p_{st} (1-p_{ts}) \right) \left(\prod_{\substack{w:(w,u)\notin E \\ w:(u,w)\notin E \\ w\neq u,v}} \sum_{s=1}^{k} \mu_s^{w\to u} (1-p_{st})(1-p_{ts}) \right) \tag{3.13}$$

In most contexts, we wouldn't have the product over the non-edges. But if the non-edges matter to us, we have to take these into account. In block models, this turns the network into a complete graph. But then every vertex sends messages to every other vertex, giving us n^2 different messages we have to keep track of.

We can simplify things by assuming that each vertex sends the same messages to all its non-neighbors. In other words, $\mu^{u\to v}$ is the same for all v such that neither (u, v) nor (v, u) is $\in E$. Denote this μ^u. Then μ^u is the marginal of vertex u taking

the messages from all the other vertices into account. This gives the update rules

$$\xi_t^{u\to v} = q_t \left(\prod_{\substack{w:(w,u)\in E \\ w:(u,w)\in E \\ w\neq u,v}} \sum_{s=1}^{k} \mu_s^{w\to u} p_{st} p_{ts} \right) \left(\prod_{\substack{w:(w,u)\notin E \\ w:(u,w)\in E \\ w\neq u,v}} \sum_{s=1}^{k} \mu_s^{w\to u} (1-p_{st}) p_{ts} \right)$$
$$\left(\prod_{\substack{w:(w,u)\in E \\ w:(u,w)\notin E \\ w\neq u,v}} \sum_{s=1}^{k} \mu_s^{w\to u} p_{st} (1-p_{ts}) \right) \left(\prod_{\substack{w:(w,u)\notin E \\ w:(u,w)\notin E \\ w\neq u,v}} \sum_{s=1}^{k} \mu_s^{w} (1-p_{st})(1-p_{ts}) \right)$$
$$for\ (u,v)\in E\ or\ (v,u)\in E$$
$$\xi_t^{u} = q_t \left(\prod_{\substack{w:(w,u)\in E \\ w:(u,w)\in E \\ w\neq u}} \sum_{s=1}^{k} \mu_s^{w\to u} p_{st} p_{ts} \right) \left(\prod_{\substack{w:(w,u)\notin E \\ w:(u,w)\in E \\ w\neq u}} \sum_{s=1}^{k} \mu_s^{w\to u} (1-p_{st}) p_{ts} \right)$$
$$\left(\prod_{\substack{w:(w,u)\in E \\ w:(u,w)\notin E \\ w\neq u}} \sum_{s=1}^{k} \mu_s^{w\to u} p_{st} (1-p_{ts}) \right) \left(\prod_{\substack{w:(w,u)\notin E \\ w:(u,w)\notin E \\ w\neq u}} \sum_{s=1}^{k} \mu_s^{w} (1-p_{st})(1-p_{ts}) \right)$$

If the network has n vertices, m edges, and k is constant, then the total number of variables we need to keep track of is $O(n+m)$. Moreover, we can update them in $O(n+m)$ time by first computing the product

$$\prod_{all\ w} \sum_{s=1}^{k} \mu_s^{w} (1-p_{st})(1-p_{ts})$$

for each type t, which gives the overall effect of all the vertices on all the others ones assuming that there are no edges. We can then obtain ξ_t^u and $\xi_t^{u\to v}$ for each u by dividing and multiplying by a finite number of terms.

Once we reach a fixed point in the messages, they can be used in place of marginal

beliefs in the Bethe formula (3.9) and (3.10),

$$b_s^u = \mu_s^u \tag{3.14}$$

$$b_{st}^{uv} \propto \begin{cases} \mu_s^{u\to v}\mu_t^{v\to u}p_{st}p_{ts} & (u,v)\in E, (v,u)\in E \\ \mu_s^{u\to v}\mu_t^{v\to u}p_{st}(1-p_{ts}) & (u,v)\in E, (v,u)\notin E \\ \mu_s^{u\to v}\mu_t^{v\to u}(1-p_{st})p_{ts} & (u,v)\notin E, (v,u)\in E \\ \mu_s^{u}\mu_t^{v}(1-p_{st})(1-p_{ts}) & (u,v)\notin E, (v,u)\notin E \end{cases}, \tag{3.15}$$

where we normalize each of these by summing over all s or all s,t. By the results of the paper [90], these beliefs should be the same as the minimizers of the Bethe free energy (3.11), which is in turn the approximation we seek for the partition function (3.1).

3.3.3 Preliminary results

Here we compare the partition function estimation from the methods detailed in section 3.2.2 and section 3.3.1 (for population annealing, here I take 100 temperature points with a population size of 10). As we will see, BP is much faster than calorimetry, and is a good approximation in all cases. Furthermore, among different BP constructions (please refer to the Appendix A for details), the variational Bethe estimate (3.11) usually provides a slightly better approximation. Constrained by the speed of calorimetry methods, we cannot scale to much bigger networks here. The runtime is averaged 10 independent runs.

Toy graphs with 6 nodes

These are synthetic graphs hand written for testing purpose. They both have only 6 nodes, with assortative and disassortative block structure respectively. These toy

graphs are so small that we can compute the partition function explicitly by summing over all the possible 2^6 block assignments:

$$Z(1) = \sum_{g} P(G, g|q, p)$$

Table 3.1: Specifications and learning results of toy SBM #1

#nodes	6	
True q	0.5	0.5
True P (k=2)	0.667 0.112	0.112 0.667
Exact $\ln Z$	-18.053	

Method	Bethe (3.11)	BP (A.4)	Calorimetry (3.3)
Estimated $\ln Z$	-18.604	-18.189	-17.997
Runtime(ms)	1537	1537	155937

Table 3.2: Specifications and learning results of toy SBM #2

#nodes	6	
True q	0.5	0.5
True P (k=2)	0.167 0.778	0.778 0.167
Exact $\ln Z$	-19.097	

Method	Bethe (3.11)	BP (A.4)	Calorimetry (3.3)
Estimated $\ln Z$	-19.098	-19.181	-19.033
Runtime(ms)	1594	1594	161023

Synthetic graph with 200 nodes

This graph is generated using the block model, with a much bigger $n = 200$. Since the exact partition function is intractable here, we validate the BP algorithms by

the result from the calorimetry method with extended running time. The same goes for the next word adjacency network.

Table 3.3: Specifications and learning results of a 200 nodes SBM

#nodes	200	
True q	0.5	0.5
True P	0.002	0.1
(k=2)	0.1	0.002
Exact $\ln Z$	N/A	

Method	Bethe (3.11)	BP (A.4)	Calorimetry (3.3)
Estimated $\ln Z$	-6995.47	-6818.25	-6683.09
Runtime(ms)	515587	515587	32766328

Word adjacency network of David Copperfield

This is a real world network made from the 60 most commonly occurring nouns and the 60 most commonly occurring adjectives in the novel *David Copperfield* by Charles Dickens. They are represented by vertices and a directed edge connects any pair that appear adjacent in the corpus, pointing from the preceding word to the following one. Eight of the words never appear adjacent to any of the others and are excluded from the network, leaving a total of 112 vertices [63]. It is a highly disassortative network, meaning most of the edges are between nouns and adjectives.

Table 3.4: Specifications and learning results of the Word adjacency network

#nodes	112	
True q	0.509	0.491
True P	0.050	0.129
(k=2)	0.010	0.012
Exact $\ln Z$	N/A	

Method	Bethe (3.11)	BP (A.4)	Calorimetry (3.3)
Estimated $\ln Z$	-2084.49	-2102.74	-2061.18
Runtime(ms)	240849	240849	57096614

3.4 The variational EM framework

In the previous sections, I explained how to estimate the partition function $P(G \,|\, p, q)$ given the parameter set Π, circumventing the latent state g and its discreteness altogether. As is usual with many generative models, the parameters in Π also need to be inferred from the data. In this section, I will put both inference tasks under a variational expectation maximization (EM) framework [59], and to showcase the flexibility of this framework, I will use the most general DC-SBM as an example.

The partition function for the DC-SBM is:

$$P(G \,|\, \theta, \omega, q) = \sum_{g} P(G, g \,|\, \theta, \omega, q)\,,$$

where the sum is over all k^n possible block assignments.

Under the variational EM framework, the E-step approximates the average over g with respect to the Boltzmann distribution, and the M step estimates θ, q and ω in order to maximize that average. Assuming that we know the marginal distributions b_s^u of each u, and joint marginal distributions of b_{st}^{uv} for each pair of nodes $\{u, v\}$, the

M-step sets θ, q and ω to their most likely estimates (MLEs),

$$\widehat{q}_s = \frac{\bar{n}_s}{n} = \frac{\sum_u b_s^u}{n}\,, \quad \widehat{\omega}_{st} = \frac{\overline{m}_{st}}{n_s n_t} = \frac{\sum_{u\neq v: A_{uv}\neq 0} A_{uv} b_{st}^{uv}}{(\sum_u b_s^u)(\sum_u b_t^u)}\,, \tag{3.16}$$

with $\widehat{\theta}_u = \frac{d_u}{\overline{d}_{g_u}}$ (where $\overline{d}_{g_u} = \frac{\sum_u b_s^u d_u}{\sum_u b_s^u}$ is the weighted average of vertex degrees of block s) being fixed given the graph and independent of the other parameters.

One approach to the E-step would use a Monte Carlo sampling algorithm to sample g from the Boltzmann distribution. However, as I have just shown in 3.16, in order to determine θ, q and ω it suffices to estimate the marginal distributions of b_s^u for each u, and joint marginal distributions of b_{st}^{uv} for each pair of nodes $\{u, v\}$ [9]. This leads to the Bethe variational approximation I have shown in the previous section (thus the abuse of belief notations b). Here I will rewrite it for the undirected DC-SBM model.

3.4.1 The variational E-step for DC-SBM

I have already shown that belief propagation is an efficient and accurate algorithm for approximating both the free energy and the marginals. Here I describe how belief propagation works for the undirected DC-SBM model, extending the treatment of the directed vanilla SBM in the previous section 3.3.2.

Recall that $\mu_s^{u\to v}$ is the probability that u would be of type s in the absence of v. Then $\mu^{u\to v}$ gets updated in light of the messages u gets from the *other* nodes as follows. Let

$$f(\theta_u, \theta_v, \omega_{st}, A_{uv}) = \frac{(\theta_u\theta_v\omega_{st})^{A_{uv}}}{A_{uv}!}\exp(-\theta_u\theta_v\omega_{st}) \tag{3.17}$$

denote the probability that A_{uv} takes its observed value assuming that $g_u = s$ and $g_v = t$. Then

$$\mu_s^{u\to v} = \frac{1}{Z^{u\to v}}\, q_s \prod_{w\neq u,v}\sum_{t=1}^{k} \mu_t^{w\to u} f(\theta_w, \theta_u, \omega_{st}, A_{wu})\,, \tag{3.18}$$

where $Z^{u\to v}$ is a normalization factor set so that $\sum_s^k \mu_s^{u\to v} = 1$. As usual in belief propagation, I assume here that the block assignment g_w of the other nodes are independent conditioned on g_u.

Note that each node sends messages to every other node, not just to its neighbors, since non-edges are also informative about g_u and g_v. Thus we have a Markov random field on a weighted complete graph, as opposed to just on the network itself. However, keeping track of n^2 messages is cumbersome. For sparse networks, we can restore scalability by noticing that, up to $O(1/n)$ terms, each node u sends the same message to all of its non-neighbors. That is, for any v such that $A_{uv} = 0$, we have $\mu_s^{u\to v} = \mu_s^u$ where

$$\mu_s^u = \frac{1}{Z^u}\, q_s \prod_{w\neq u} \sum_{t=1}^{k} \mu_t^{w\to u} f(\theta_w, \theta_u, \omega_{st}, A_{wu})\,. \tag{3.19}$$

This simplification reduces the number of messages to $O(n+m)$. We can then write

$$\begin{aligned}\mu_s^{u\to v} =& \frac{1}{Z^{u\to v}}\, q_s \times \prod_{w} \sum_{t=1}^{k} \mu_t^w f(\theta_w, \theta_u, \omega_{st}, 0) \\ &\times \prod_{w\neq v, A_{uw}\neq 0} \frac{\sum_{t=1}^k \mu_t^{w\to u} f(\theta_w, \theta_u, \omega_{st}, A_{wu})}{\sum_{t=1}^k \mu_t^w f(\theta_w, \theta_u, \omega_{st}, 0)}\,.\end{aligned} \tag{3.20}$$

Since the second product depends only on θ_u, we can compute it once for each degree in the network, and then update the messages for each u in $O(k^2 d_u)$ time. Thus, for fixed k, the total time it takes to update all the messages is $O(m + \ell n)$, where ℓ is the number of distinct degrees. As discussed in [23], for many networks only a constant number of updates are necessary in order to reach a fixed point, making the entire E-step of linear scalability in terms of number of edges. Please refer to [23] for details.

The BP estimate of the marginals are

$$\Pr[g_u = s] = b_s^u \propto \mu_s^u\,, \tag{3.21}$$

$$\Pr[g_u = s, g_v = t] = b_{st}^{uv} \propto f(\theta_u, \theta_v, \omega_{st}, A_{uv}) \mu_s^{u\to v} \mu_t^{v\to u}\,. \tag{3.22}$$

These are normalized so that $\sum_s^k b_s^u = 1$, and $\sum_{s,t=1}^k b_{st}^{uv} = 1$.

As we have discussed in Section 3.3.1, marginals based on convergent BP messages also minimize the Bethe free energy, which is a second order Kikuchi approximation to the log partition function [89]:

$$\log P(G \mid \theta, \omega, q) \approx \sum_u \log Z^u + \frac{1}{2} \sum_{s,t=1}^k \omega_{st} n_s n_t - \sum_{u \neq v, A_{uv} \neq 0} \log \left[\sum_{s,t=1}^k f(\theta_u, \theta_v, \omega_{st}, A_{uv}) \mu_s^{u \to v} \mu_t^{v \to u} \right].$$

I want to emphasize that while I use the linear-time BP algorithm for the experiments in this dissertation, the results on model selection in the following chapters are quite indifferent as to *how* the likelihood or partition function is computed. Under the variational EM framework alone, many other choices of $Q(g)$ is possible, including the whole range of Kikuchi approximations we have seen earlier 3.3.1.

However, the Bethe approximation and the linear-time BP algorithm does achieve the desired balance between scalability and accuracy for the task of learning block models, as validated by its linear scalability and optimal detectability [23, 58]. The variational EM framework with the linear-time BP as the E-step will be the main inference algorithm of choice in the following chapters.

3.5 Supervised learning on block models

So far, we have been learning the block models solely from the data G. This is called unsupervised learning. In many real-world networks, nodes and edges can have class labels, or variables/attributes that can affect the network's topology. If some or all of the labels are given with some probability or confidence level, we would like to take advantage of what is available to help predict the missing labels on other nodes

or edges. On a higher level, we can also cast model selection problems as a targeted learning task based on such knowledge.

This is where supervised learning comes in, making our inference process conditioned on observed ground truth, and complement our model–based approaches with the power of data–driven approaches. In this section, I will show how to adapt block models as well as the variational EM framework for such supervised learning tasks. To highlight the flexibility of this framework without digressing too much into other learning tasks beside model selection, much of the algorithmic details are left in the Appendices (see B.1 and B.2).

3.5.1 Supervised link prediction

An example for such a supervised learning tasks is the recommendation system on e-commerce websites. Recommendation systems make predictions about preferences of items based on preferences or ratings previously expressed by users. If we assume that similar users have similar preferences with similar items, block models become the perfect model for the task.

Following the model in [36], all the users $u \in U$ and items $i \in I$ are nodes, observed ratings $r \in R^O$ can be modeled by weighted edges. The graph $G = (U + I, R^O)$ is then bipartite, and the adjacency matrix will be dominated by missing links/ratings waiting to be predicted.

Now I assume that G is generated by a special SBM. Labels for both user u and item i are first independently generated. For each pair of nodes $\{u \in U, i \in I\}$, the rating from u to i follows a categorical distribution that depends only on these labels. Given a block assignment g, the probability of generating the observed ratings R^O

under this model is

$$P(R^O, g \,|\, p, \gamma, \eta) = \prod_{g(u)\in P_U} \prod_{g(i)\in P_I} \prod_{r=1}^{K} \gamma_{g(u)}^{n_{g(u)}} \eta_{g(i)}^{n_{g(i)}} (p^r_{g(u)g(i)})^{n^r_{g(u)g(i)}}\,, \tag{3.23}$$

where $\gamma_{g(u)}$ is the prior probability that a user belongs to a group $g(u)$, and $n_{g(u)}$ is the number of user actually in the group $g(u)$ under the block assignment g. The same applies to $\eta_{g(i)}$ and $n_{g(i)}$ for the items. p^r_{st} is the categorical probability for rating r and n^r_{st} represents the number of observed ratings r between the user block s and the item block t.

Besides the straightforward change from Bernoulli to categorical distributions, the above model differs from the vanilla SBM in how data is observed. For all block models introduced in 2.3, we treat edges and non-edges between every pair of nodes as observed. Here, however, the model is only based on observed ratings. Non-ratings are completely irrelevant under this model.

Another key difference between the models is the inference task. The ultimate goal here is to predict missing links (edges/ratings), whereas the block models in 2.3 are built for community detection. With this in mind, we can further simplify the model by targeting the following conditional distribution directly:

$$\begin{aligned} P(r_{ui} = r|R^O) =& \frac{1}{Z} \sum_g p^r_{g(u)g(i)} P(R^O, g \,|\, p, \gamma, \eta) \\ =& \frac{1}{Z} \sum_g P(R^O_+, g \,|\, p, \gamma, \eta)\,, \end{aligned} \tag{3.24}$$

where R^O_+ represents the new graph with the r label on the missing link. By marginalizing over all possible latent states g, we end up with the total probability of the missing rating regardless of how the nodes might be assigned to blocks.

The normalization term Z in (3.24) is the partition function of the Boltzman distribution:

$$P(g \,|\, R^O, p, \gamma, \eta) = \frac{e^{\ln P(R^O, g \,|\, p, \gamma, \eta)}}{\sum_g e^{\ln P(R^O, g \,|\, p, \gamma, \eta)}}\,, \tag{3.25}$$

where the denominator is the partition function

$$Z = \sum_g P(R^O, g \,|\, p, \gamma, \eta)\,.$$

The numerator in (3.24) on the other hand corresponds to the partition function Z_+ of the new graph with the r label on the missing link.

$$Z_+ = \sum_g P(R^O_+, g \,|\, p, \gamma, \eta)\,.$$

Just like the partition functions we have seen earlier, we can use the variational EM framework to estimate both Z and Z_+. Because these two only differ on a single rating, the final ratio has a very simple mathematical form. The application of the framework is fairly straightforward. Because I am not focusing on model selection for link prediction in later chapters, please refer to Appendix B.1 for algorithmic details. In fact, this framework is so flexible that the full Bayesian integrated recommendation system can also be solved in a similar fashion (see Appendix B.2).

If we use the block models in Section 2.3 as generative models for the task of link prediction, we can view conditional models like (3.24) as discriminative in nature. Compared with generative models, discriminative models usually enjoy performance advantages in predicting the target variables [30, 14, 47, 50]. However, they require complete knowledge of the label on every single data point to work. In the recommendation system example above, only observed ratings are treated as input data in the model. Another drawback for discriminative models for link prediction is that links are not independent, the conditional distribution (3.24) will be different for different target links, leading to repeated inference for each new target link.

3.5.2 Semi-supervised community detection

Unlike discriminative models, generative models can take advantage of both labeled and unlabeled data. This is especially important in the *semi-supervised* setting,

where we only have partially labeled data [30, 50]. Generative models are based on the joint distribution of all target variables together with observed variables. This also leads to a huge performance advantage on networks, as discriminative approaches are forced to model multiple conditional distributions.

With the generative block models in 2.3, we can do semi-supervised link prediction if we introduce a new edge type to represent "unknown" links, which would change the edge distributions from Bernoulli to categorical as we have seen in the previous subsection. An adapted variational EM algorithm similar to B.1 will provide an efficient solver for the model.

Another learning task is semi-supervised community detection, where we are given the graph and some of the node labels, and the goal is to predict the community membership of the unlabeled nodes. Designed as generative community models, block models in 2.3 can be adapted for semi-supervised node classification by simply fixing the block membership for the labeled nodes.

If the learning algorithm is based on Monte Carlo sampling in the latent block assignment space, supervision can be achieved directly by rejecting moves into the "illegal" label space which contradicts the ground truth. This is the approach I take in the next section. Under the variational EM framework, supervision fixes the block membership for the labeled nodes, leading to a partition function summed over only the unlabeled nodes. Accordingly, the BP in the E-step will have some fixed out-going messages for the labeled nodes. For example, if we know that node u has a true block label of t, then the message it sends to any neighbor v will be fixed as $\mu_t^{u\to v} = 1$, regardless of its in-coming messages during all time steps.

If one prefers the maximum likelihood approach over the partition function, the EM framework can even be changed in to a greedy "maximization-maximization" framework, with a specialized BP for the max-product inference in the E-step (the

standard BP performs a sum-product inference. See Appendix A.2).

3.6 Active Learning on block models

Based on semi-supervised community detection models, we can take one step further and uses information-theoretic techniques to actively choose which labels to explore. Given a SBM, it can help predicting the labels of unexplored nodes after exploring a relatively small fraction of the network, driving the parameters towards correct values even faster.

This so–called *Active Learning*, coupled with a generative model, offers a new approach to analyzing networks where the topology is known, but knowledge of class labels is incomplete and costly to obtain. This could be the case, for instance, if we have a network of blogs and hyperlinks between them (like citations, trackbacks, blogrolls, etc.) and we are trying to classify the blogs according to their political leanings. Another possible application is in online social networks, where friendships are known and we are trying to infer hidden demographic variables. This problem is sometimes referred to as collective classification [79]. However, in that work the focus is on classification of individual nodes. In contrast, our focus is on the discovery of functional communities in the network, and our underlying generative model is designed around the assumption of that these communities exist.

We make no initial assumptions about the structure of the network—for instance, whether its groups are assortative, disassortative, or some mixture of the two. We assume that we can learn the label of any given node, but at a cost, say in terms of work in the field or laboratory. Our goal is to identify a small subset of nodes such that, once we explore them and learn their labels, we can accurately predict the labels of all the others.

We present a general approach to this problem. Our algorithm uses information-theoretic measures to decide which node to explore next—that is, which one will give us the most information about the rest of the network. We start with a probabilistic generative model of the network, called a *SBM* [44, 85], in which groups connect to each other according to a matrix of probabilities. This model allows an arbitrary mixture of assortative and disassortative structure, as well as directed links from one group to another, and has been used to model networks in many fields (e.g. [7, 43, 74]).

We stress, however, that our approach could be applied equally well to many other probabilistic models, such as those where nodes belong to a mixture of classes [4], a hierarchy of classes and subclasses [21], locations in a latent geographical or social space [40], or niches in a food web [87]. It could also be applied to DC-SBMs such as those in [48, 57, 65], which treat the nodes' degrees as parameters rather than data to be predicted.

At each stage of the learning process, some of the nodes' labels are already known and we need to decide which node to explore next. We do this by estimating, for each node, the mutual information between its label and the joint distribution of all the others' labels, conditioned on the labels of the nodes that are known so far. We obtain this estimate by Gibbs sampling, giving each classification of nodes a probability integrated over the parameters of the SBM. We then explore the node for which this mutual information is largest.

A key fact about the mutual information, which we argue is essential to our algorithm's performance, is that it is not just a measure of uncertainty: it is a combination of uncertainty about a node's label and the extent to which it is correlated with the labels of other nodes. Thus the algorithm explores nodes which maximize the expected amount of information it will gain about the entire network. It skips nodes whose labels seem obvious to it, or which are uncertain but have little effect

on other nodes. In an assortative network, for instance, it starts by exploring nodes which are central to their communities, and then explores nodes along the boundaries between them, without being told in advance to pursue this strategy.

We also present an alternate approach which maximizes a quantity we call the *average agreement*. For each node v, this is the average number of nodes at which two independent samples of the Gibbs distribution agree, conditioned on the event that they agree at v. Like mutual information, average agreement is high for nodes that are highly correlated with the rest of the network. A similar idea (but not applied to networks) is present in [76].

We test our algorithm on three real-world networks: the social network of a karate club, a network of common adjacent words in a Charles Dickens novel, and a marine food web of species in the Antarctic. Each of these networks is curated in the sense that we possess the correct node labels, such as the faction of the social network each individual belongs to, the part of speech of each word, or the part of the habitat each species lives in. We judge our algorithm according to how accurately it predicts the labels of the unexplored nodes, as a function of the number of nodes it has explored so far. We also compare our algorithm with several simple heuristics, such as exploring nodes based on their degree or betweenness centrality, and find that it significantly outperforms them.

3.6.1 Related work

The idea of designing experiments by maximizing the mutual information between the variable we learn next and the joint distribution of the other variables, or equivalently the expected amount of information we gain about the joint distribution, has a long history in statistics, artificial intelligence, and machine learning, e.g. Mackay [55] and Guo and Greiner [38]. Indeed, it goes back to the work of Lindley [53] in the

1950s. However, to our knowledge this is the first time it has been coupled with a generative model to discover hidden variables in networks.

In recent work, Zhu, Lafferty, and Ghahramani [92] study active learning of node labels using Gaussian fields and harmonic functions defined using the graph Laplacian. However, this technique only applies to networks where neighboring nodes are likely to be in the same class—that is, networks with assortative community structure. In contrast, our techniques are capable of learning about much more general types of network structure, including disassortative and directed relationships between functional communities.

Another approach to active learning of node labels is found in the work of Bilgic and Getoor [11] and Bilgic, Mihalkova, and Getoor [12], who use collective vector-based classifiers. By properly defining the collective relationships between nodes, both assortative or disassortative communities can be learned in this framework. However, our technique differs from theirs by using mutual information as the active learning criterion, which takes into account not just uncertainty, but correlations as well.

Additional works by Goldberg, Zhu, and Wright [32] and Tong and Jin [82] also perform semi-supervised learning on graphs, and handle the disassortative case. But they work in a setting where they know, for each link, if the ends should have the same or different labels, such as if one writer quotes another with pejorative words. In contrast, we work in a setting where we have no such information: only the topology is available to us, and there are no signs on the edges telling us whether we should propagate similar or dissimilar labels.

3.6.2 The simplified SBM and the Bayesian integration

We represent our network as we did in (2.11), with the additional assumption of q terms being equal

$$P(G \mid g, p) \propto \left(\prod_{(u,v)\in E} p_{g(u),g(v)} \right) \left(\prod_{(u,v)\notin E} (1 - p_{g(u),g(v)}) \right)$$
$$= \prod_{s,t=1}^{k} p_{st}^{e_{st}} (1 - p_{st})^{n_s n_t - e_{st}} . \tag{3.26}$$

This simplified SBM is well-known in the machine learning, statistics, and network communities [10, 83, 37, 41, 43, 74] and has also been used in ecology to identify groups of species in food webs [7]. Unlike e.g. [83, 41, 43], we do not assume that p_{st} takes one value when $i = j$ and a smaller value when $i \neq j$. In other words, we do not assume an assortative community structure, where nodes are more likely to be connected to other nodes of the same class. Nor do we require in general that $p_{st} = p_{ts}$, since the directed nature of the edges may be important—for instance, in a food web or word adjacency network.

If all block assignments g are equally likely *a priori*, then Bayes' rule implies that the Gibbs distribution on the classifications, i.e., the probability of g given G, is proportional to the probability of G given g:

$$P(g \mid G) \propto P(G \mid g) . \tag{3.27}$$

In order to define $P(G \mid g)$, we need to integrate $P(G \mid g, p)$ over some prior probability distribution on p. If we assume that the p_{st} are independent, then this integral factors over the product (3.26). In particular, if each p_{st} follows a beta prior, we have the

Bayesian estimate of edge probabilities

$$\begin{aligned} P(G \,|\, g) &= \iiint_0^1 \mathrm{d}\{p_{st}\}\, P(G \,|\, g, p) \\ &= \prod_{s,t=1}^{k} \int_0^1 \mathrm{d}p_{st}\, \mathrm{Beta}(p_{st} \,|\, \alpha, \beta)\, p_{st}^{e_{st}} (1 - p_{st})^{n_s n_t - e_{st}} \\ &= \prod_{s,t=1}^{k} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \int_0^1 \mathrm{d}p_{st}\, p_{st}^{e_{st}+\alpha-1} (1 - p_{st})^{n_s n_t - e_{st} + \beta - 1} \\ &= \prod_{s,t=1}^{k} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\,\Gamma(\beta)} \frac{\Gamma(e_{st} + \alpha)\,\Gamma(n_s n_t - e_{st} + \beta)}{\Gamma(n_s n_t + \alpha + \beta)} \,. \end{aligned} \tag{3.28}$$

For reasonable choices of the hyper-parameters α and β, the prior dominates only in small data cases, such as very small networks or sparsely populated classes. For such small data cases, the beta prior allows the user to input some domain knowledge about, say, the (dis)assortativity of the target network's community structure. In the limit of large data, the prior will wash out and the data-driven community structure will dominate.

If the user wishes to remain agnostic, however, he or she can specify a uniform prior ($\alpha = \beta = 1$) and allow the learning algorithm to estimate the degree of assortativity, disassortativity, directedness, and so on entirely from the data. We take this approach in this paper, in which case

$$P(G \,|\, g) = \prod_{s,t=1}^{k} \frac{1}{(n_s n_t + 1)\binom{n_s n_t}{e_{st}}} \,. \tag{3.29}$$

An even simpler approach is to assume that the p_{st} take their maximum likelihood values

$$\hat{p}_{st} = \operatorname*{argmax}_p \, P(G \,|\, g, p) = e_{st}/n_s n_t \,, \tag{3.30}$$

and set $P(G \,|\, g) = P(G \,|\, g, \hat{p})$. This approach was used, for instance, for a hierarchical SBM in [21]. When k is fixed and the n_s are large, this will give results similar

to (3.29), since the integral over p is tightly peaked around $\hat{p}$. However, for any particular finite graph it makes more sense, at least to a Bayesian, to integrate over the p_{st}, since they obey a posterior distribution rather than taking a fixed value. Moreover, averaging over the parameters as in (3.29) discourages over-fitting, since the average likelihood goes down when we increase k and hence the volume of the parameter space. This gives us a principled way for order selection, although in this paper we set k by hand.

This Bayesian integration approach is actually a major branch in statistical model selection. I shall dedicate the whole chapter 5 for Bayesian model selection methods in much more details and generality. However, Monte Carlo sampling algorithm provided below does provide one of the main inference frameworks for such Bayesian models. The variational EM framework is a better alternative for full Bayesian approaches where latent states are summed over as well (see Appendix B.2).

3.6.3 Active learning and sampling methods

In the active learning setting, the algorithm can learn the class label of any given node, but at a cost—say, by devoting resources in the laboratory or the field. Since these resources are limited, it has to decide which node to explore. Its goal is to explore a small set of nodes and use their labels to guess the labels of the remaining nodes.

One natural approach is to explore the node v with the largest mutual information (MI) between its label $g(v)$ and the labels $g(G \setminus v)$ of the other nodes according to the Gibbs distribution (3.27). We can write this as the difference between the entropy of $g(G \setminus v)$ and its conditional entropy given $g(v)$,

$$\mathrm{MI}(v) = I(v; G \setminus v) = H(G \setminus v) - H(G \setminus v \,|\, v) \; . \tag{3.31}$$

Here $H(G \setminus v \,|\, v)$ is the entropy, averaged over $g(v)$ according to the marginal of $g(v)$

in the Gibbs distribution, of the joint distribution of $g(G \setminus v)$ conditioned on $g(v)$. In other words, $\mathrm{MI}(v)$ is the expected amount of information we will gain about $g(G \setminus v)$, or equivalently the expected decrease in the entropy, that will result from learning $g(v)$.

Since the mutual information is symmetric, we also have

$$\mathrm{MI}(v) = I(v; G \setminus v) = H(v) - H(v \,|\, G \setminus v) \;, \tag{3.32}$$

where $H(v)$ is the entropy of the marginal distribution of $g(v)$, and $H(v \,|\, G \setminus v)$ is the entropy, on average, of the distribution of $g(v)$ conditioned on the labels of the other nodes. Thus $\mathrm{MI}(v)$ is large if (i) we are uncertain about v, so that $H(v)$ is large, and (ii) v is strongly correlated with the other nodes, so that $H(v \,|\, G \setminus v)$ is small.

We estimate these entropies by sampling from the space of classifications t according to the Gibbs distribution. Specifically, we use a single-site heat-bath Markov chain. At each step, it chooses a node v uniformly from among the unexplored nodes, and chooses its label $g(v)$ according to the conditional distribution proportional to $P(G \,|\, g)$, assuming that the labels of all other nodes stay fixed. In addition to exploring the space, this allows us to collect a sample of the conditional distribution of the chosen node v and its entropy. Since $H(v \,|\, G \setminus v)$ is the average of the conditional entropy, and since $H(v)$ is the entropy of the average conditional distribution, we can write

$$I(v; G \setminus v) = -\sum_{i=1}^{k} \langle P_s \rangle \ln \langle P_s \rangle + \left\langle \sum_{i=1}^{k} P_s \ln P_s \right\rangle \;, \tag{3.33}$$

where P_s is the probability that $g(v) = s$ and $\langle \cdot \rangle$ denotes the average, according to the Gibbs distribution, over the labels of the other nodes.

We offer no theoretical guarantees about the mixing time of this Markov chain, and it is easy to see that there are families of graphs and values of k for which it it

takes exponential time. However, for the real-world networks we have tried so far, it appears to converge to equilibrium in a reasonable amount of time. We test for equilibrium by measuring whether the marginals change noticeably when the number of updates is increased by a factor of 2. We improve our estimates by averaging over many runs, each one starting from an independently random initial state.

We say that the algorithm is in *stage* j if it has already explored j nodes. In that stage, it estimates $\mathrm{MI}(v)$ for each unexplored node v, using the Markov chain to sample from the Gibbs distribution conditioned on the labels of the nodes explored so far. It then explores the node v with the largest MI. We provide it with the correct value of $g(v)$ from the curated network, and it moves on to the next stage.

The mutual information is not the only quantity we might use to identify which node to explore. Another is the *average agreement*, which we define as follows. Given two classifications g_1, g_2, define their *agreement* as the number of nodes on whose labels they agree,

$$|g_1 \cap g_2| = |\{v : g_1(v) = g_2(v)\}| \ . \tag{3.34}$$

Since our goal is to label as many nodes correctly as possible, we wish we could maximize the agreement between an classification g_1, drawn from the Gibbs distribution, and the correct classification g_2. However, the algorithm doesn't know g_2, so it assumes that it is drawn from the Gibbs distribution as well. Exploring v projects onto the part of the joint distribution of (g_1, g_2) where $g_1(v) = g_2(v)$. So, we define $\mathrm{AA}(v)$ as the expected agreement between two classifications g_1, g_2 drawn independently from the Gibbs distribution, conditioned on the event that they agree at v:

$$\mathrm{AA}(v) = \frac{\sum_{g_1,g_2:g_1(v)=g_2(v)} P(g_1)P(g_2)\,|g_1 \cap g_2|}{\sum_{g_1,g_2:g_1(v)=g_2(v)} P(g_1)P(g_2)} \,. \tag{3.35}$$

We estimate the numerator and denominator of $\mathrm{AA}(v)$ using the same heat-bath Gibbs sampler as for $\mathrm{MI}(v)$, except that we sample independent pairs of classifications

(g_1, g_2) by starting the Markov chain at two independently random initial states.

3.6.4 Results and discussion

We tested our algorithms on three different networks from three different fields. The first is Zachary's Karate Club [91]. As shown in Fig. 3.3, this is a social network consisting of 34 members of a karate club, where undirected edges represent friendships. The club split into two factions, indicated by diamonds and circles respectively. One of them centered around the instructor (node 1) and the other around the club president (node 34), each of which formed their own club. Shaded nodes are more peripheral, and have weaker ties to their communities. This network is highly assortative, with a high density of edges within each faction and a low density of edges between them.

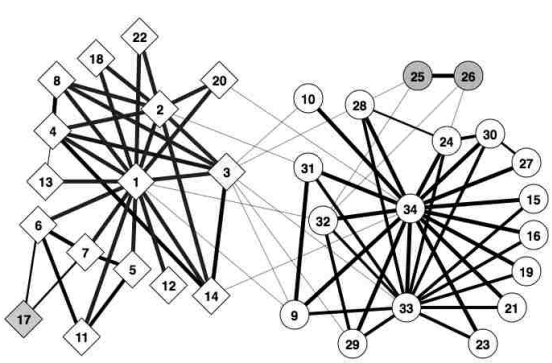


Figure 3.3: Zachary's Karate Club.

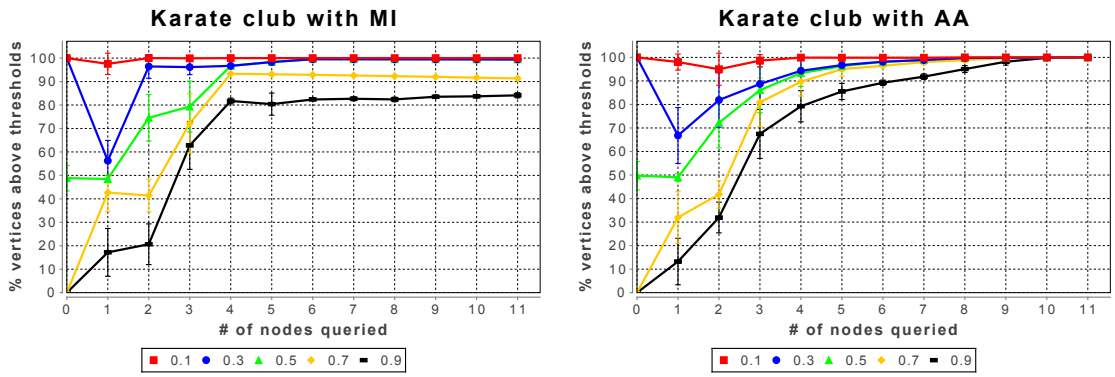


Figure 3.4: Results of the active learning algorithms on Zachary's Karate Club network.

We judge the performance of each algorithm by asking, at each stage and for each node, with what probability the Gibbs distribution assigns it the correct label. In each stage we sampled the Gibbs distribution using 100 independently chosen initial conditions, doing 2×10^4 steps of the heat-bath Markov chain for each one, and computing averages using the last 10^4 steps. Increasing the number of Markov chain steps to 10^5 per stage produced only marginal improvements in performance. Fig. 3.4 shows what fraction of the unexplored nodes are assigned the correct label with probability at least q, for various thresholds $q = 0.1, 0.3, 0.5, 0.7, 0.9$, as a function of the stage j.

After exploring just four or five nodes, both of our algorithms succeed in correctly predicting the labels of most of the remaining nodes—i.e., to which faction they belong—with high accuracy. The AA algorithm performs slightly better than MI, achieving an accuracy close to 100% after exploring nine nodes. Of course, the Karate Club network is quite small, and there are many community-finding algorithms that classify the two factions with perfect or near-perfect accuracy [70, 27].

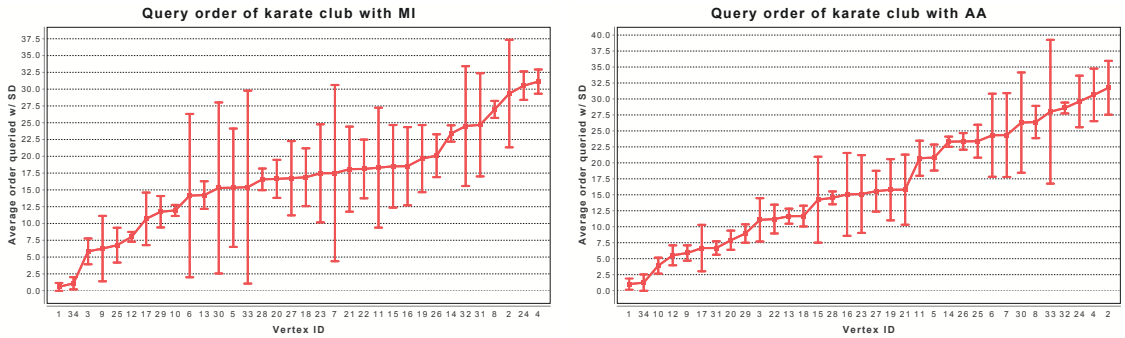


Figure 3.5: The order in which the active learning algorithms explore nodes in Zachary's Karate Club.

Perhaps more interesting is the *order* in which our algorithms choose to explore the nodes. In Fig. 3.5, we sort the nodes in order of the median stage at which they are explored. Error bars show 90% confidence intervals over 100 independent runs of each algorithm. Some nodes show a large variance in the stage in which they are explored, while others are consistently explored at the beginning or end of the process. Both algorithms start by exploring nodes 1 and 34, which are central to their respective communities. Note that these nodes are chosen, as we argued above, not just because their labels are uncertain, but because they are highly correlated with the labels of other nodes.

After learning that nodes 1 and 34 are in class 1 and 2 respectively, the algorithms "know" that the network consists of two assortative communities. They they explore nodes such as 3, 9, and 10 which lie at the boundary between these communities. Once the boundary is clear, they can easily predict the labels of the remaining nodes. The last nodes to be explored are those such as 2, 4, and 24, which lie so deep inside their communities that their labels are not in doubt.

The second network consists of the 60 most commonly occurring nouns and the 60 most commonly occurring adjectives in Charles Dickens' novel *David Copperfield*. A directed edge connects any pair of words that appear adjacently in the text, pointing

from the preceding word to the following one. Excluding eight words which are disconnected from the rest leaves a network with 112 nodes [63]. Unlike Zachary's Karate Club, this network is both directed and highly disassortative. Of the 1494 edges, 1123 of them point from adjectives to nouns. This lets us classify most nodes early on, simply by labeling a node as an adjective or noun if its out-degree or in-degree is large.

figures/queryWords

Figure 3.6: The order in which the active learning algorithm MI explores nodes in word adjacency network from the novel *David Copperfield*.

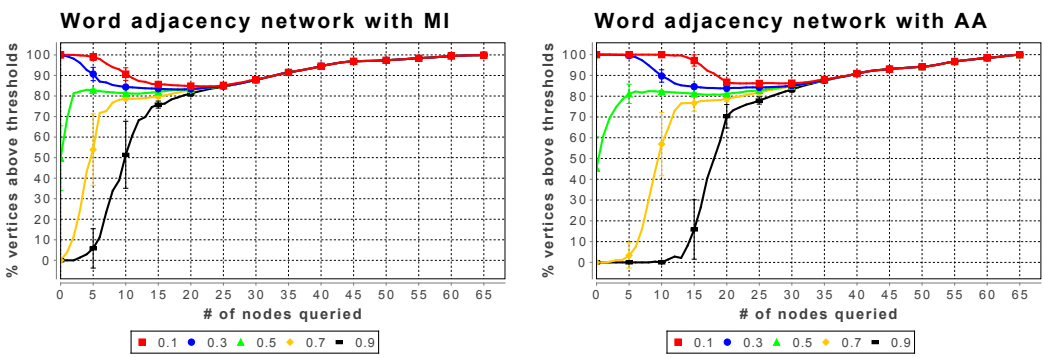


Figure 3.7: Results of the active learning algorithms on word adjacency network in the novel *David Copperfield* by Charles Dickens.

Accordingly, our algorithms focus their attention on words about which they are uncertain, like "early," "low," and "nothing," whose out-degrees and in-degrees in the text are roughly equal, and words like "perfect" that precede words of both classes (see Fig. 3.6, where green and yellow nodes represent nouns and adjectives respectively; rectangular nodes are explored first, and elliptical ones lasg). Once these nodes are resolved, both algorithms achieve high accuracy—80% accuracy after exploring 20 nodes and close to 100% after exploring 65 nodes (see Fig. 3.7).

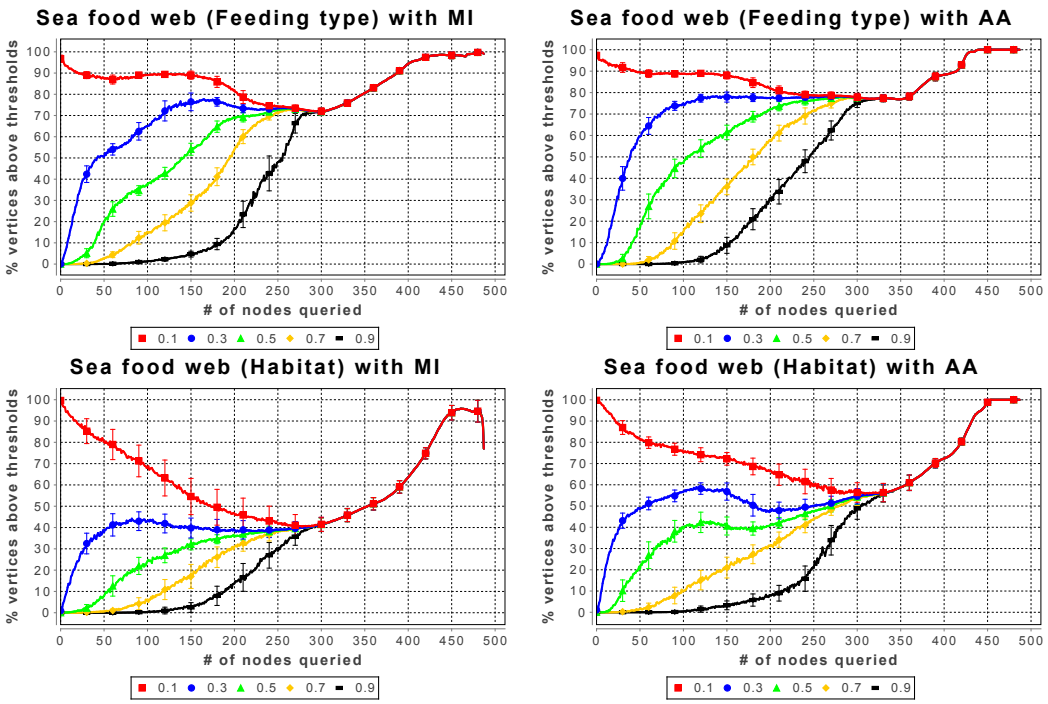


Figure 3.8: Results for the Weddell Sea food web.

In each stage we sampled the Gibbs distribution using 100 independently chosen initial conditions, doing 5×10^4 steps of the heat-bath Markov chain for each one, and computing averages using the last 2.5×10^4 steps. Increasing the number of Markov chain steps to 10^5 per stage produced only marginal improvements in performance. As in Fig. 3.4, the y-axis shows the fraction of unexplored nodes which are labeled correctly by the conditional Gibbs distribution with probability at least q, for $q = 0.1, 0.3, 0.5, 0.7, 0.9$. The performance of the two algorithms is similar in the later stages, but unlike the Karate Club, here MI performs noticeably better than AA in the early stages.

The third network is a food web of 488 species in the Weddell Sea in the Antarctic [25, 16, 46], with edges pointing to each predator from its prey. This data set is very rich, but we focus on two particular variables—the feeding type and the habitat in which the species lives. The feeding type takes $k = 6$ values, namely pri-

mary producer, omnivorous, herbivorous/detrivorous, carnivorous, detrivorous, and carnivorous/necrovorous. The habitat variable takes $k = 5$ values, namely pelagic, benthic, benthopelagic, demersal, and land-based.

We show results of our algorithms for both variables in Fig. 3.8. The results are averaged over 100 runs of each algorithm. In each stage we sampled the Gibbs distribution using 100 independently chosen initial conditions, doing 5×10^4 steps of the heat-bath Markov chain for each one, and computing averages using the last 2.5×10^4 steps. For the feeding type, after exploring half the nodes, both algorithms correctly label about 75% of the remaining nodes. For the habitat variable, both algorithms are less accurate, although AA performs somewhat better than MI. Note that the accuracy only includes the unexplored nodes, not the nodes we have already explored. Thus it can decrease if we explore easily-classified nodes early on, so that hard-to-classify nodes form a larger fraction of the remaining ones.

Fig. 3.8 shows that both algorithms get to a state where they are confident, but wrong, about many of the unexplored nodes. For the feeding type variable, for instance, after the AA algorithm has explored 300 species, it labels 75% of the remaining nodes correctly with probability 90%, but it labels the other 25% correctly with probability less than 10%. In other words, it has a high degree of confidence about all the nodes, but is wrong about many of them. Its accuracy improves as it explores more nodes, but it doesn't achieve high accuracy on all the unexplored nodes until there are only about 60 of them left.

Why is this? We argue that the fault lies, not with our learning algorithms and the order in which they explore the nodes, but with the SBM and its ability to model the data. For example, for the habitat variable, these algorithms perform well on pelagic, demersal, and land-based species. But the benthic habitat, which is the largest and most diverse, includes species with many feeding types and trophic levels.

These additional variables have a large effect on the topology, but they are not taken into account by the SBM. As a result, more than half the benthic species are mislabeled by the SBM in the following sense: even if we condition on the correct habitats of *all* the other species, the species' most likely habitat is pelagic, benthopelagic, demersal, or land-based. Specifically, 219 of the 488 species are mislabeled by the most likely SBM, 94% of them with confidence over 0.9.

Of course, we can also regard our algorithms' mistakes as evidence that these habitat classifications are not cut and dried. Indeed, ecologists recognize that there are "connector species" that connect one habitat to another, and belong to some extent to both.

To test our hypothesis that it is the SBM's inability to model the data that causes some nodes to be misclassified, we artificially modified the data set to make it consistent with the SBM. Starting with the nodes' original class labels, we updated the habitat of each species to its most likely value according to the SBM, given the habitats of all the other species. After iterating this process six times, we reached a fixed point where each species' habitat is consistent with the SBM's predictions. On this synthetic data set both of our learning algorithms perform perfectly, predicting the habitat of every species with close to 100% accuracy after exploring just 18% of them.

More generally, it is important to remember that the topology of the network is only imperfectly correlated with the nodes' types. Zachary [91] relates that one of members of the Karate Club joined the instructor's faction even though the network's topology suggests that he was more strongly connected to the president. The reason is that he was only three weeks away from a test for his black belt when the split occurred. He had already invested four years learning the instructor's style of karate, and if he had joined the president's club he would have had to start over with a white belt. In any real-world network, there is information of this kind that is not

reflected in the topology and which is hidden from our algorithm. If a node is of a given class for idiosyncratic reasons like these, we cannot expect any algorithm based solely on topology and the other nodes' class labels—no matter how sophisticated a probabilistic model we use—to correctly classify it.

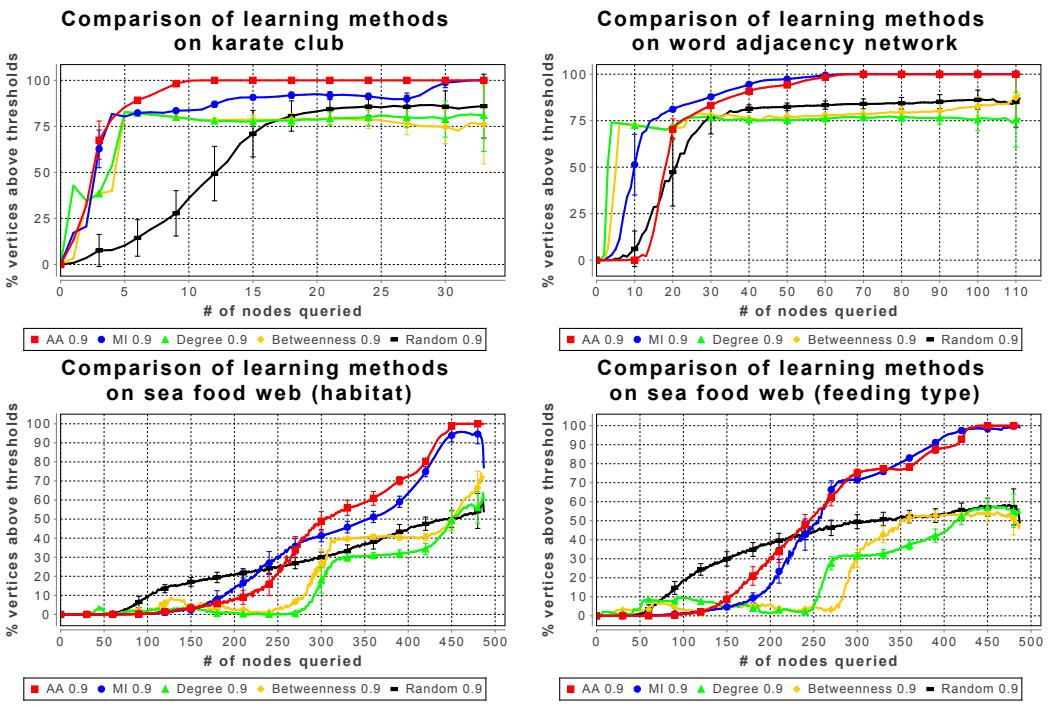


Figure 3.9: A comparison of the MI and AA learning algorithms with three simple heuristics.

We compared our active learning algorithms with several simple heuristics. These include exploring the node with the highest degree in the subgraph of unexplored nodes, exploring the node with the highest betweenness centrality (the fraction of shortest paths that go through it, see [15, 60, 62]) in the subgraph of unexplored nodes, and exploring a node chosen uniformly at random from the unexplored ones. We judge the performance of these heuristics using the same Gibbs sampling process as for MI and AA.

In Fig. 3.9, we show the results of these heuristics at the 0.9 accuracy threshold

on all three networks, including both the habitat and feeding type variables in the food web. On Zachary's Karate Club (lefg) our algorithms outperform these heuristics consistently. In the *David Copperfield* network (righg), the highest-degree and highest-betweenness heuristics enjoy an early lead, but quickly hit a ceiling and are surpassed by MI and AA.

For the Weddell Sea food web (bottom), the highest-degree and highest-betweenness heuristics perform poorly throughout the learning process. One reason for this is that many nodes with high degree or high betweenness are easy to classify from the labels of their neighbors. By exploring these nodes first, these heuristics leave themselves mainly with hard-to-classify nodes. The random-node heuristic performs surprisingly well early on, but all three heuristics are worse than MI or AA once they have explored half the nodes.

Chapter 4

Frequentist Model Selection

In this chapter, I start the statistical inquiry with Frequentist model selection techniques, whose derivations are very similar to those of AIC. In particular, the likelihood ratio test provides us a powerful hypothesis test for nested models. This classic statistical tools comes with the full controllability of margins of error and confidence intervals, however, many of the analytical results for independence data do not work properly any longer on networks. Using the variational EM framework for bootstrapping simulations, I shall investigate the likelihood ratios of two specific pairs of nested models in Poisson-SBM vs DC-SBM, and SBMs with 1 and 2 blocks. By correcting the theory according to the simulation data, I will finally propose new frequentist model selection methods for block models, including the corrected AIC.

4.1 The likelihood ratio test

In frequentist statistics, the problem of model selection between a pair of nested models can be casted as a hypothesis test. The Neyman–Pearson lemma, named after Jerzy Neyman and Egon Pearson, states that when performing a hypothesis test for point hypotheses, and in our case, nested candidate models, the likelihood ratio test (LRT) is the uniformly most powerful test [77].

To construct a LRT, we need to explicitly state our null model H_0, and the more general, nesting alternative H_1. Assuming that the likelihood function of the graph given the alternative model is $P(G \mid H_1)$, the appropriate test statistic is the log-likelihood ratio,

$$\Lambda(G) = \log \frac{\sup_{H_1 \in M_1} P(G \mid H_1)}{\sup_{H_0 \in M_0} P(G \mid H_0)}, \tag{4.1}$$

where the *Supremum function* in the denominator is taken with respect to the nested and thus smaller domain of the null model.

We reject the null model in favor of the more elaborate alternative when Λ exceeds some threshold. This threshold, in turn, is fixed by our desired error rate, and by the distribution of Λ when G is generated from the null model. When G is small, the null-model distribution of Λ can be found through parametric bootstrapping [22]: generating random graphs $\tilde{G}$ from the null model, fitting H_0 and H_1 to each graph, and evaluating $\Lambda(\tilde{G})$. When n is large, however, fitting models for each graph will take much longer. It would be helpful to replace bootstrapping with analytic calculations.

A classic result in asymptotic statistics [77] asserts that in hypothesis-testing problems like this, the large-sample null distribution of $\Lambda(G)$ approaches the chi-squared distribution $\frac{1}{2}\chi^2_\ell$, where ℓ is the number of constraints that must be imposed on H_1 to recover H_0. However, deriving the χ^2 distribution relies on second or-

der Taylor expansions. The process is very similar to those we have seen for AIC (see 2.2.1). Indeed, the same twice differentiable assumption (assumptions A.2) needs to held for this analytical result. Fortunately, we have already overcome the discrete latent states by calculating the partition functions instead.

There is another key assumption in this derivation [77, 31]: namely, that the log-likelihood of both models is well-approximated by a quadratic function in the vicinity of its maximum, so that the parameter estimates have Gaussian distributions around the true model. The most common grounds for this assumption are central limit theorems for IID data (assumptions A.3), or more generally, being in a "large data limit" (assumptions A.1). We will see that, for sparse networks, this assumption does not hold for many parameters. To avoid bootstrapping, we need to be able to correctly predict Λ's null distribution when the average degree of the graph is small, while recovering the classical χ^2 distribution in the the limit of large, dense graphs.

4.2 Model selection between SBM and DC-SBM

To reiterate the motivation behind the development of DC-SBM, vanilla and Poisson-SBMs impose real restrictions on networks; notably, the degree distribution within each block is asymptotically Poisson. This makes these block models implausible for many real-world networks, where the degrees within each community are highly inhomogeneous. Fitting these block models to such networks tends to split the high- and low- degree nodes in the same community into distinct blocks; for instance, dividing both liberal and conservative political blogs into high-degree "leaders" and low-degree "followers" [1, 48]. To avoid this effect, and allow degree inhomogeneity within blocks, there is a long history of generative models where the probability of an edge depends on node attributes θ_u as well as their group memberships (e.g. [57, 73]). Here I use the DC-SBM due to [48].

We often lack the domain knowledge to choose between the vanilla/Poisson and the degree-corrected block model, and so are faced with a classic problem of statistical model selection. Following the classic LRT introduced earlier, I do some bootstrapping experiments using the variational EM framework with the linear BP as the E-step (see 3.4). This choice of learning algorithm made it possible to gather thousands of samples at a respectable accuracy from the null distribution on networks with hundreds of thousands nodes. The simulation result show that the usual χ^2 theory for likelihood ratios relies on approximations which are invalid in our setting, because of the dependency and sparsity of network data.

I derive the correct asymptotics under certain assumptions, recovering the classic asymptotics in the limit of dense graphs, but finding that significant corrections are needed in the sparse case. Numerical experiments confirm the validity of my expressions, and I apply my method to a range of real and synthetic networks. The same corrections are applicable to AIC for the model selection problem between SBM and DC-SBM.

In the following derivations, I will focus on Poisson-SBM as it is a proper nested model within DC-SBM. However, according to section 2.3.2, the vanilla SBM and the Poisson-SBM share a lot in common, especially when the network is sparse, making the model selection criteria easily applicable for both.

4.2.1 The LRT for Poisson-SBM vs DC-SBM

Since the Poisson-SBM is nested within the DC-SBM model, any given graph G is at least as likely under the latter as under the former. Moreover, if the Poisson-SBM is the null model which generated the data, all of the parameters shared between the two should converge to the same MLEs (3.16), at least in the limit of large networks. Following the LRT construction in section 4.1, we define the null model H_0 as the

Poisson-SBM, with DC-SBM as the alternative. We have the log-likelihood ratio,

$$\Lambda_{DC}(G) \;=\; \log \frac{\sup_{H_1} \sum_g P(G, g \,|\, \theta, \omega, q)}{\sup_{H_0} \sum_g P(G, g \,|\, \omega, q)} \,. \tag{4.2}$$

with the P functions defined in (2.13) and (2.15).

The classic result asserts that the large-sample null distribution of $\Lambda_{DC}(G)$ approaches $\frac{1}{2}\chi^2_\ell$. In this case, $\ell = n-k$, as we must set all n of the $\widehat{\theta}_u$ to 1 to recover H_0 from H_1. Notice that our identifiability convention $\sum_{u:g_u=s} \theta_u = D_s$ already imposed k constraints.

However, for sparse networks, this assumption of large data limit does not hold for the parameters θ_u. Nevertheless, with some work we are able to compute the mean and variance of Λ's null distribution. While we recover the classical χ^2 distribution in the the limit of large, dense graphs, there are significant corrections when the average degree of the graph is small. In particular, χ^2 testing commits type I errors in the sparse case whenever the graph is sufficiently large, rejecting the Poisson-SBM in favor of DC-SBM even for graphs generated by the former (see Fig. 4.1). In essence, it underestimates the amount of degree inhomogeneity we would get simply from random noise, incorrectly concluding that the inhomogeneity must come from underlying properties of the nodes.

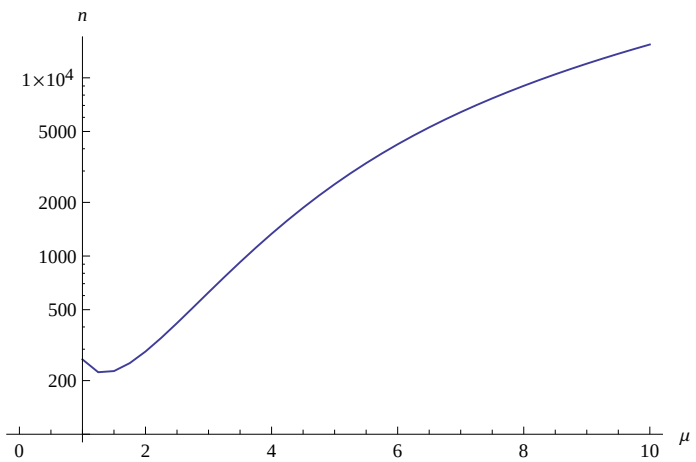


Figure 4.1: The size n, as a function of the average degree μ, above which naive χ^2 testing commits a type I error with 95% confidence
Type I error means incorrectly rejecting the SBM for graphs generated by the Poisson-SBM. For instance, for $\mu = 5$, χ^2 commits a type I error at roughly $n > 3000$, while for $\mu = 3$, it does so for $n > 700$. Here we assume the asymptotic analysis of (4.4)–(4.7) *for the mean and variance of the likelihood ratio; see Fig. 4.3 for comparison with experiment.*

To obtain theoretical estimates of the null distribution of Λ, I assume that the Gibbs distribution of both models is concentrated on the same block assignment g. This is a major assumption, but it is borne out by our experiments (Fig. 4.2 and Fig. 4.3), and the fact that under some conditions [10] the block models recovers the underlying block assignment exactly. Under this assumption, while the free energy differs from the ground state energy by an entropy term, the free energy *difference* between the two models has the same distribution as the ground state energy difference. The MLE estimates for H_0 and H_1 are then given by (3.16).

Substituting these into (4.2) gives Λ_{DC} the form of a Kullback-Leibler divergence,

$$\Lambda_{DC}(G) = \log \prod_u \left(\frac{d_u}{\overline{d}_{g_u}} \right)^{d_u} = \sum_u d_u \log \frac{d_u}{\overline{d}_{g_u}} \,. \tag{4.3}$$

Recall that $\overline{d}_s$ is the empirical mean, not the expected degree $\mu_s = \sum_t q_t \omega_{st}$ of the

true underlying Poisson-SBM.

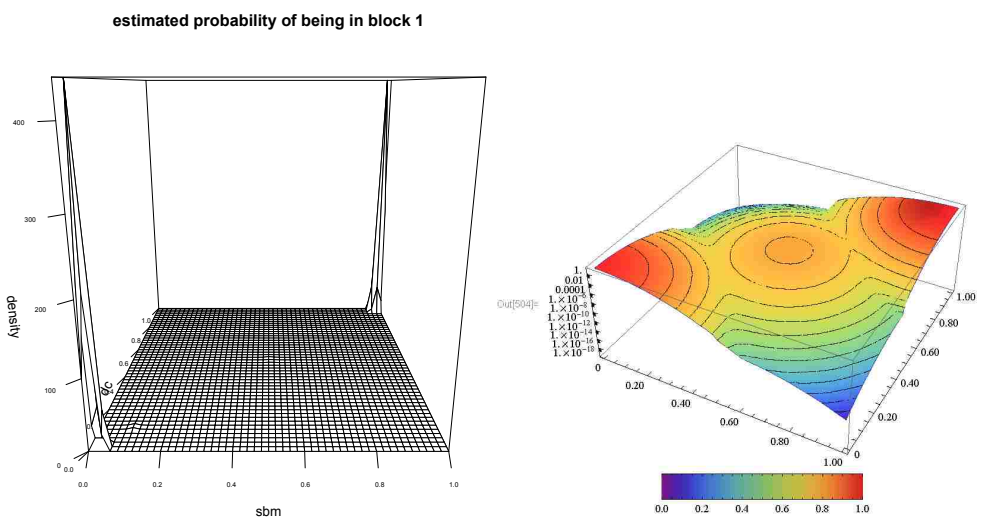
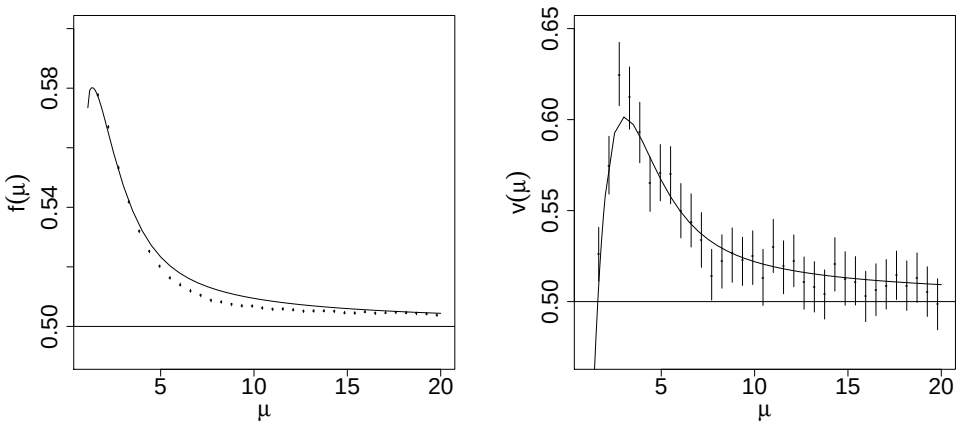


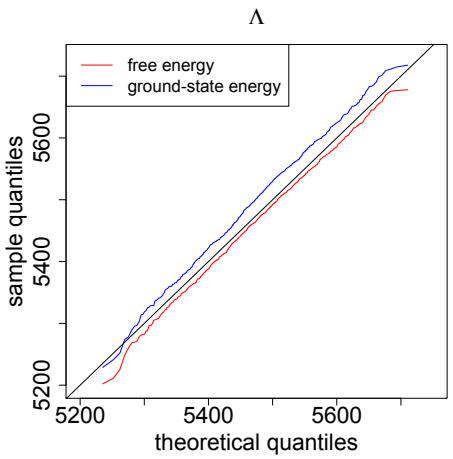
Figure 4.2: Joint density of posterior probabilities over block assignments, showing that the Poisson-SBM and the DC-SBM are concentrated around the *same* ground state

The synthetic network has $n = 10^3$*,* $k = 2$ *groups of equal size* $q_1 = q_2 = 1/2$*, average degree* $\mu_r = 11$*, and associative structure with* $\omega_{12}/\omega_{11} = \omega_{21}/\omega_{22} = 1/11$*. The* x *and* y *axes are the marginal probabilities of being in block 1 according to Poisson-SBM and DC-SBM. The left is a 3D histogram while the right is a heat map with logarithmic* z *axis.*



(a)

(b)



(c)

Figure 4.3: (a) $f(\mu)$ from (4.5), the expected log-likelihood difference per node, compared to simulation results; (b) the asymptotic variance of the log-likelihood difference per node, from (4.7), with simulation results; (c) QQ plots comparing the distribution of log-likelihood differences from 10^4 synthetic networks with $\mu = 3$ to a Gaussian with the theoretical mean and variance.
All simulations have $n = 10^4$, $k = 2$, $q_1 = q_2 = 1/2$, *and* $\omega_{12}/\omega_{11} = 0.15$, $\omega_{11}/\omega_{22} = 1$. *In (a) and (b), each point is the average over* 10^3 *networks, including 95% bootstrap confidence intervals.*

We can understand the asymptotic null distribution of Λ_{DC} by assuming that the d_u in each block r are IID and Poisson with expectation μ_r. This assumption is sound in the limit $n \to \infty$, since the correlations between node degrees are $O(1/n)$. In that case, we can compute the expectation and variance of Λ_{DC} analytically (see Appendix C.1). These results show how the behavior of Λ_{DC} differs from naive χ^2 asymptotics, as well as revealing the limits where the naive results apply. Specifically,

$$\mathbb{E}[\Lambda_{DC}] = \sum_r n_r f(\mu_r) - f(n_r \mu_r) \tag{4.4}$$

where if d is Poisson with mean μ,

$$f(\mu) = \mathbb{E}[d \log d] - \mu \log \mu = \sum_{d=1}^{\infty} \frac{\mathrm{e}^{-\mu} \mu^d}{d!} \, d \log d - \mu \log \mu \,. \tag{4.5}$$

In the limit $\mu \to \infty$, i.e., for dense graphs, both $f(\mu)$ and $f(n\mu)$ approach $1/2$, and (4.4) gives $\mathbb{E}[\Lambda_{DC}] = (n-k)/2$ just as in the standard χ^2 analysis. However, when μ is finite, $f(\mu)$ differs significantly from $1/2$.

The variance of Λ_{DC} is more complicated, but still calculable. The limiting variance per node is

$$\lim_{n\to\infty} \frac{1}{n} \mathrm{Var}[\Lambda_{DC}] = \sum_r q_r v(\mu_r) \,, \tag{4.6}$$

where, again taking d to be Poisson with mean μ,

$$v(\mu) = \mu(1 + \log \mu)^2 + \mathrm{Var}[d \log d] - 2(1 + \log \mu) \, \mathrm{Cov}[d, d \log d] \,. \tag{4.7}$$

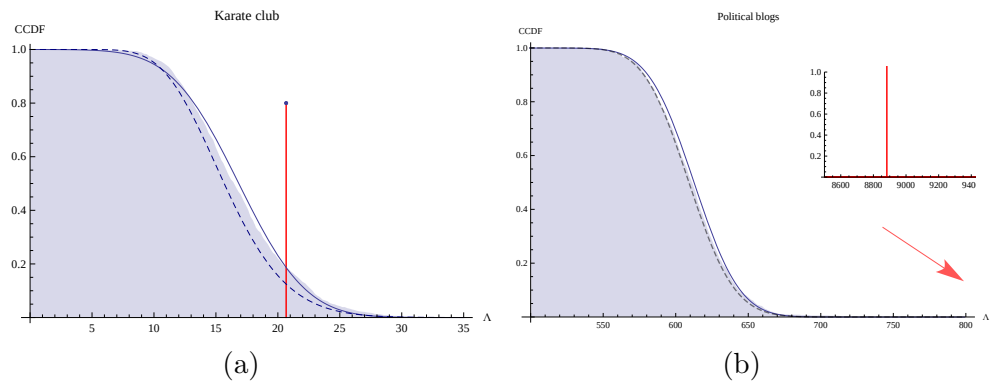
Since the variance of χ^2_ℓ is 2ℓ, the χ^2 analysis would predict $(1/n)\mathrm{Var}[\Lambda_{DC}] = 1/2$. Indeed $v(\mu)$ approaches $1/2$ in the limit $\mu \to \infty$, but like $f(\mu)$ it differs significantly from $1/2$ for finite μ. Plots of $f(\mu)$ and $v(\mu)$ can be found in Fig. 4.3(a,b). More details are available in Appendix C.1.

Why exactly does the null distribution of Λ_{DC} differ from the usual χ^2 distribution? The reason is that the parameters θ_u are in a high-dimensional regime, and

thus are not in the large data limit. We have one observation for each node, i.e., its degree d_u. If a Poisson distribution has small mean, its shape differs significantly from a Gaussian, and so does the posterior distribution of the mean based on a single sample. In particular, $P(\theta \mid d)$ follows a Gamma distribution, if the prior on θ is uninformative [94]. When the degrees are large, both the sample distribution and the posterior become Gaussian, and the χ^2 analysis takes over; but when they are small, the geometry is simply different, causing $f(\mu)$ and $v(\mu)$ to differ from 1/2. This would eventually lead to a type I error for the χ^2 testing, rejecting the Poisson-SBM for almost all graphs generated by the itself. Since χ^2 distribution is tightly peaked around $0.5n$, the situation also becomes worse with bigger n (see Fig. 4.1).

As shown in Fig. 4.3, experiments on synthetic networks generated from the Poisson-SBM show that the mean and variance of Λ_{DC} are very well fit by our theoretical results. I have not attempted to compute higher moments of Λ_{DC}. However, if we assume that d_u are independent, then the central limit theorem applies, and Λ_{DC} follows a Gaussian distribution in the limit of large n. Quantile plots from the same experiments (Fig. 4.3(c)) show that a Gaussian with mean and variance given by (4.4) and (4.6) is indeed a good fit. Moreover, the free energy difference and the ground state energy difference have similar distributions, as implied by our assumption that both Gibbs distributions are concentrated around the ground state. Interestingly, in Fig. 4.3(c), the degree is low enough that this concentration must be imperfect, but our theory still holds remarkably well. Notice that all the synthetic experiments done in this section took the simplifying assumptions that q and μ does not change across all blocks.

4.2.2 Results on real networks



(a) (b)

Figure 4.4: Hypothesis testing of real world networks
(a): Zachary's karate club [91], where $n = 34$. The CCDF (complementary cumulative distribution) of the log-likelihood ratio Λ_{DC} under the null model is estimated using bootstrapping (shaded), and is fit reasonably well by the CCDF of a Gaussian (curve) with our theoretically predicted mean and variance. The observed $\Lambda_{DC} = 20.7$ (marked with the red line) has p-values of 0.186 and 0.187 according to the bootstrap and theoretical distributions respectively. The naive χ^2 test gives a p-values of 0.125 (dashed), which is quite a bit off because of the low degree network. (b): A network of political blogs [1] where $n = 1222$. Since the network has a higher average degree, the bootstrap distribution (shaded) is very well fit by both our theory (curve) and the naive χ^2 test. The actual log-likelihood ratio is so far in the tail (see inset) that its p-value is effectively zero. Thus for the blog network, we can decisively reject the ordinary block model in favor of the degree-corrected model, while for the karate club, the evidence is less clear.

I have derived the theoretical null distribution of Λ_{DC}, and backed up our calculations with simulations. We now apply our theory to the two real world examples studied in [48], demonstrating how our methods can be applied in different situations.

The first is a social network consisting of 34 members of a karate club, where undirected edges represent friendships [91]. The network is made up of two assortative blocks centered around the instructor and the club president, each with a high degree hub and lower-degree peripheral nodes. The authors of [48] compared the perfomance of Poisson-SBM and DC-SBM on this network, and heavily favored DC-

SBM over Poisson-SBM because the former leads to a community structure agreeing with the ground truth. Our test, however, shows that the evidence is not strong enough to reject the null model with any great confidence. As shown in Fig. 4.4(a), the distribution of Λ_{DC} from bootstrap experiments is fit reasonably will by a Gaussian with our predicted mean and variance. The observed $\Lambda_{DC} = 20.7$ has a p-value of 0.187 according to the theoretical Gaussian, and 0.186 according to the bootstrap distribution. Thus a prudent statistician would think twice before embracing the additional n parameters of DC.

Indeed, under the active learning framework introduced earlier 3.6, we found that the vanilla SBM labels most of the nodes correctly if we fix the block assignment of the instructor and the president to 1 and 2 respectively. This implies that the degree inhomogeneity is not too extreme, and that only a handful of nodes are responsible for the better performance of DC.

This network is of such low degree that the naive χ^2 test cannot work, and small n hinders our Gaussian approximation. For such small networks, I suggest parametric bootstrapping using our BP algorithm to estimate the null distribution. Nonetheless, our estimation of the mean and variance remain solid, making it possible to quickly check some extreme cases.

The second example is a network of political blogs in the US assembled by Adamic and Glance [1]. As in [48], I focus on the giant component, which consists of 1222 blogs and 19087 links between them. The blogs have known political leanings, and were labeled as either liberal or conservative. The network is assortative and has a highly right-skewed degree distribution within each block. In its agreement with ground truth, DC-SBM substantially outperforms Poisson-SBM, as observed in [48]. This time around, our hypothesis testing procedure completely agrees with their choice of model. As shown in Fig. 4.4(b), the bootstrap distribution of Λ_{DC} is very well fit by a Gaussian with our theoretical prediction of the mean and variance. The

observed log-likelihood ratio $\Lambda_{DC} = 8883$ is 330 standard deviations above the mean. It is essentially impossible to produce such extreme results through mere fluctuations under the null model. Thus, for this network, introducing n extra parameters to capture the degree heterogeneity, and rejecting Poisson-SBM in favor of DC-SBM, is fully justified.

The blog network is an example for the advantages in theoretical predictions. As with many other real networks, n is large enough that bootstrapping would be too slow, but the Gaussian approximation is fairly tight. Unfortunately, since the average degree $\bar{d}$ is relatively high, our corrected theory does not do much better than the naive χ^2 approximation.

4.2.3 Corrected AIC for SBM vs DC-SBM

Deciding between the vanilla/Poisson SBM and DC-SBM models for sparse graphs presents a difficult hypothesis testing problem. The distribution of the log-likelihood ratio Λ does not follow the classic χ^2 theory, because the nuisance parameter θ, only present in the alternative, suffers from the curse of dimensionality. We have nonetheless derived Λ's mean and variance in the limit of large, sparse graphs, where node degrees become independent and Poisson. Simulations using the variational EM algorithm confirm the accuracy of our theory for moderate n, and we applied it to two real networks.

While hypothesis testing such like the LRT give us the full power of frequentist statistics with margins of error and confidence intervals, standard information criteria are much easier to interpret, and are more widely used in application domains. While we have not directly dealt with AIC, the derivations of AIC use exactly the same asymptotics as the χ^2 test 2.2.1. As a result, AIC will break down for the same reasons χ^2 theory fails for sparse graphs, and more importantly, the same correction

factor shall applied for the penalty term in AIC. This leads to the corrected AIC for the model selection problem between the vanilla/Poisson SBM and DC-SBM:

$$AIC_{DC}(M_i) = -2\ln P(G|M_i, \hat{\Pi}_i) + 2(1 + \frac{1}{6\mu} + \frac{1}{6\mu^2} + O(\frac{1}{\mu^3}))|\Pi_i|\,, \tag{4.8}$$

where Π_i is the parameter set for model M_i, and the penalty scaling term $1 + \frac{1}{6\mu} + \frac{1}{6\mu^2} + O(\frac{1}{\mu^3})$ depends on the average degree μ. This correction comes from our theory about the expected likelihood ratio, which is proportional to the expected Kullback-Leibler divergence measured by AIC. Please refer to the Appendix C.1 for details of a nontrivial analytical solution, which leads to the above asymptotic correction.

The model selection problem between vanilla/Poisson SBM and DC-SBM models might be just one example in the hierarchy of block models 2.2, many other block models have similar node attributes that participant in edge generation. Just like DC-SBMs with the θ parameters, these models are likely to suffer from the same problems in sparse networks. Similar corrections could be derived for choosing such models versus the vanilla SBM.

From a more general perspective, the work here opens the way for hypothesis testing to be applied in a wide range of network problems. With the efficient variational EM bootstrapping algorithm, we can replicate the process of doing simulations, analyzing data and correcting theories for many other model selection problems. In the next section, I will employ this exact strategy for the order selection problem.

4.3 Order selection of the vanilla SBM

Choosing the right number of blocks for various SBMs is an important problem which has attracted much attention in the literature. Here I will focus on the very simple case of choosing between the one block vanilla SBM and the two blocks vanilla SBM, under the framework of the LRT. For order selection of the SBM for any number of

blocks as well as order selection for other variants of the SBM, please refer to the next chapter.

Following the LRT construction in section 4.1, I define the null model H_0 as the one block vanilla SBM, which is by definition the Erdős–Rényi graph. With the vanilla SBM with two blocks as the alternative model H_1, we have the log-likelihood ratio,

$$\Lambda(G) = \log \frac{\sup_{H_1} \sum_g P(G, g \,|\, q, p)}{\sup_{H_0} \sum_g P(G, g \,|\, q, p)}\,, \tag{4.9}$$

where P defined as in (2.11) can be specified as

$$P(G, g \,|\, q, p) = q_1^{n_1} q_2^{n_2} \prod_{s \le t, s, t = 1}^{2} p_{st}^{m_{st}} (1 - p_{st})^{n_s n_t - m_{st}}\,,$$

under H_1. The above definition leads to multiple ways how H_0 can be nested within $H = 1$. For example, we can set $q_1 = 0$ which according to (3.16) would lead to $n_1 = 0$ and thus the additional block empty. Another way to reduce H_1 to H_0 is by forcing all the entries in the p matrix the same.

Degenerate reductions like the empty blocks poses a major challenge for statistical analysis of likelihood ratios in many other situations besides the block model. One prominent example is the classic mixture model, which for its simplicity is surprisingly hard to analyze. In the following subsections, I shall first introduce a simplified SBM model compatible with the state of art likelihood ratio test methods for mixture models. Then, I will try to generalize the result to the vanilla SBM and so on.

4.3.1 The pairwise mixture model

The classic mixture model is a natural model for data with unobserved heterogeneity. In many different disciplines, data are believed to be mixed samples from multiple subpopulations, which can be modeled by different parametric distributions. The

marginal distribution for the whole population is then a mixture model [51]. If we limit the number of subpopulation to be two as we did for the SBM, the likelihood of a finite mixture model with the parameter set X for data Y can be defined as

$$P(Y \mid X) = \prod_{\forall y_u \in Y} \left[(1-\alpha) f(y_u \mid x_1) + \alpha f(y_u \mid x_2)\right] ,$$

where α is the mixing proportion of the components, $y_u \in Y$ are individual samples and x_1, x_2 are the different parameter values for subpopulations of the same parametric form.

The above formulation has a seemingly simple form, i.e. weighted average of component likelihoods. However, it is not statistically identifiable under the one component null model, that is we can reduce the two component model to a single component by either setting $\alpha = 0$ or $X_1 = X_2$. Just like the degenerate reductions for SBMs, this identifiability problem in mixture models leads to irregularity in its likelihood ratio tests.

Overcoming this irregularity proved to be quite a challenge. The state of art likelihood ratio test techniques in the statistics literature restores the identifiability by introducing a penalty function to the likelihood. This is called the modified likelihood ratio test (MLRT), which is defined as

$$\Lambda(Y) = \log \frac{\sup_{H_1} P'(Y \mid X)}{\sup_{H_0} P'(Y \mid X)} = \log P'(Y \mid \hat{\alpha}, \hat{x_1}, \hat{x_2}) - \log P'(Y \mid (1/2), \hat{x_0}, \hat{x_0}) , \quad (4.10)$$

where H_0 is the one component null model and H_1 is the two components mixture model as the alternative. $P'(Y \mid X)$ is the modified likelihood. Under some conditions, the limiting distribution of $\Lambda(Y)$ is $(1/4)(\chi_0^2 + \chi_1^2)$ [51].

To apply these latest statistics theories to SBMs, here I will build a pairwise version of the mixture model with similar formulation as the SBM. As usual, we have an undirected graph $G = (V, E)$ with n nodes. I assume that there are 2 blocks $\{1, 2\}$. To emulate the independent data samples in mixture models, I assume that

for each pair of nodes $\{u, v\}$, there is an edge from u to v with a probability $p_{g'(uv)g'(vu)}$ that depends only on their block labels. The labels are now pairwise, and is generated independently each time a pair of nodes undergo this edge generating process. Each label generation for node u, however, follows the same distribution

$$P_{g'(u)} = \frac{q_{g'(u)}^{\frac{1}{n-1}}}{q_1^{\frac{1}{n-1}} + q_2^{\frac{1}{n-1}}}$$

with $q_1 + q_2 = 1$, regardless of the other participanting node v. Given a pairwise block assignment, i.e., a function $g' : V^2 \to \{1, 2\}^2$ assigning $n-1$ independent labels to every node for each of its edge generation, the probability of generating a given graph G in this model is

$$P_{mix}(G, g' \,|\, q, p) = \prod_{u<v} \frac{q_{g'(uv)}^{\frac{1}{n-1}} q_{g'(vu)}^{\frac{1}{n-1}}}{(q_1^{\frac{1}{n-1}} + q_2^{\frac{1}{n-1}})^2} p_{g'(uv)g'(vu)}^{A_{uv}} (1 - p_{g'(uv)g'(vu)})^{1-A_{uv}} .$$

Summing it over all pairwise block assignments, we get the partition function

$$\begin{aligned} P_{mix}(G \,|\, q, p) =& \sum_{g'} P_{mix}(G, g' \,|\, q, p) \\ =& \frac{1}{(q_1^{\frac{1}{n-1}} + q_2^{\frac{1}{n-1}})^{n(n-1)}} \prod_{u<v} \sum_{s,t=1}^{2} q_s^{\frac{1}{n-1}} q_t^{\frac{1}{n-1}} p_{st}^{A_{uv}} (1 - p_{st})^{1-A_{uv}} , \quad (4.11) \end{aligned}$$

where we are able to factor it into local terms thanks to the pair-wise independence between all edges. Notice that each local term is a weighted average of component likelihoods, i.e. we have a mixture model with each edge being a sample from the 2×2 components. We shall call the above model the pairwise mixture model.

Just like data under the classic mixture model is independent and identically distributed (i.i.d.), the above partition function is just a fancy formulation of the

one block vanilla SBM. To see this, recall the likelihood of a one block vanilla SBM,

$$\begin{aligned} P(G \,|\, p) &= \prod_{(u,v)\in E} p \prod_{(u,v)\notin E} (1-p) \\ &= \prod_{(u,v)\in E} \sum_{s,t=1}^{2} \frac{q_s^{\frac{1}{n-1}} q_t^{\frac{1}{n-1}}}{(q_1^{\frac{1}{n-1}} + q_2^{\frac{1}{n-1}})^2} p_{st} \prod_{(u,v)\notin E} \sum_{s,t=1}^{2} \frac{q_s^{\frac{1}{n-1}} q_t^{\frac{1}{n-1}}}{(q_1^{\frac{1}{n-1}} + q_2^{\frac{1}{n-1}})^2} (1-p_{st}) \,, \end{aligned} \tag{4.12}$$

where p is a mixtures of p_{st} entries.

If we take the logarithm of the partition function (4.11),

$$\begin{aligned} &\log P_{mix}(G \,|\, q, p) \\ &= \sum_{u<v} \log \mathbb{E}_{c_{st}^{uv}} \left[\frac{q_s^{\frac{1}{n-1}} q_t^{\frac{1}{n-1}} p_{st}^{A_{uv}} (1-p_{st})^{1-A_{uv}}}{c_{st}^{uv}} \right] - n(n-1) \log(q_1^{\frac{1}{n-1}} + q_2^{\frac{1}{n-1}}) \\ &= \sum_{u<v} \mathbb{E}_{c_{st}^{uv}} \left[\log(q_s^{\frac{1}{n-1}} q_t^{\frac{1}{n-1}} p_{st}^{A_{uv}} (1-p_{st})^{1-A_{uv}}) - \log c_{st}^{uv} \right] \\ &\quad + \sum_{u<v} \mathbb{D}_{KL}(c_{st}^{uv} \parallel \hat{P}_{st}^{uv}) - n(n-1) \log(q_1^{\frac{1}{n-1}} + q_2^{\frac{1}{n-1}}) \,. \end{aligned} \tag{4.13}$$

Here we have used the variational trick to switch the logarithm and expectation operators, just as we did in 3.3. This time, however, we constructed a variational distribution c_{st}^{uv} for the local pairwise distribution $P(uv)$ ($\forall u, v, \sum_{st} c_{st}^{uv} = 1$), instead of the global Boltzman distribution. Again, thanks to the pair-wise independence, we can actually recover the most likely local pairwise distribution $\hat{P}_{st}^{uv}$ exactly by setting:

$$\hat{c}_{st}^{uv} = \frac{q_s^{\frac{1}{n-1}} q_t^{\frac{1}{n-1}} p_{st}^{A_{uv}} (1-p_{st})^{1-A_{uv}}}{\sum_{st} q_s^{\frac{1}{n-1}} q_t^{\frac{1}{n-1}} p_{st}^{A_{uv}} (1-p_{st})^{1-A_{uv}}} \,. \tag{4.14}$$

No longer an approximation, the Kullback-Leibler divergence goes to zero, and

we have the following formulation of free energy for the pairwise mixture model,

$$\log P_{mix}(G \mid q, p) = \sum_{u<v} \left[\sum_{st} \left[\hat{c}^{uv}_{st} \log(q_s^{\frac{1}{n-1}} q_t^{\frac{1}{n-1}} p_{st}^{A_{uv}} (1-p_{st})^{1-A_{uv}}) \right] + \mathbb{S}[\hat{c}^{uv}_{st}] \right] - n(n-1) \log(q_1^{\frac{1}{n-1}} + q_2^{\frac{1}{n-1}}) \,.$$

Applying the same EM algorithm of the variational framework introduced in 3.3, we have the MLEs:

$$\widehat{q}_s^{\frac{1}{n-1}} \propto \bar{n}_s = \sum_u \hat{c}^u_s \,, \quad \widehat{p}_{st} = \frac{\overline{m}_{st}}{n_s n_t} = \frac{\sum_{(u,v)\in E} \hat{c}^{uv}_{st}}{(\sum_u \hat{c}^u_s)(\sum_u \hat{c}^u_t)} \,.$$

If we take the weighted average of $\hat{p}_{st}$ entries,

$$< \widehat{p} >= \sum_{st} \frac{\bar{n}_s \bar{n}_t}{n^2} \widehat{p}_{st} = \sum_{s,t=1}^{2} \frac{\hat{q}_s^{\frac{1}{n-1}} \hat{q}_t^{\frac{1}{n-1}}}{(\hat{q}_1^{\frac{1}{n-1}} + \hat{q}_2^{\frac{1}{n-1}})^2} \widehat{p}_{st} \,,$$

we get exactly the same mixture as we did in (4.12).

4.3.2 Order selection of the vanilla SBM

Now let us go back to the problem of order selection for the vanilla SBM. Recall under the variational framework introduced in 3.3, we can rewrite the partition function of the two blocks vanilla SBM as:

$$\log P(G \mid q, p) = \sum_{u<v} \mathbb{E}_{b^{uv}_{st}} \left[\log(q_s^{\frac{1}{n-1}} q_t^{\frac{1}{n-1}} p_{st}^{A_{uv}} (1-p_{st})^{1-A_{uv}}) - \log b^{uv}_{st} \right] + \mathbb{D}_{KL}(B \parallel P^*) \,, \tag{4.15}$$

where the variational distribution B is an approximation to the true Boltzman distribution P^*. In 3.3, we considered the following Bethe formulation of it:

$$B(g) = \frac{\prod_{u<v} b^{uv}_{g_u g_v}}{\prod_u (b^u_{g_u})^{n-2}} \,,$$

with $b_s^u = \sum_{t,\forall v} b_{st}^{uv}$.

As I have shown earlier, local marginals optimized using the BP algorithm lead to exact recovery of P^* on trees. Even if the graph has a lot of short loops, we can still get very close to it, thus making the Kullback-Leibler divergence $\mathbb{D}_{KL}(B \parallel P^*)$ negligible. Now (4.15) becomes the first term in the pairwise mixture model, if we plug in the BP marginals into (4.13):

$$\begin{aligned}\log P_{mix}(G \mid q,p) &= \sum_{u<v} \mathbb{E}_{b_{st}^{uv}} \left[\log(q_s^{\frac{1}{n-1}} q_t^{\frac{1}{n-1}} p_{st}^{A_{uv}} (1-p_{st})^{1-A_{uv}}) - \log b_{st}^{uv}\right] \\ &\quad + \sum_{u<v} \mathbb{D}_{KL}(b_{st}^{uv} \parallel \hat{P}_{st}^{uv}) - n(n-1)\log(q_1^{\frac{1}{n-1}} + q_2^{\frac{1}{n-1}}) \\ &= \log P(G \mid q,p) + \sum_{u<v} \mathbb{E}_{b_{st}^{uv}} \left[\log \frac{b_{st}^{uv}}{\hat{P}_{st}^{uv}}\right] - n(n-1)\log(q_1^{\frac{1}{n-1}} + q_2^{\frac{1}{n-1}})\,.\end{aligned} \tag{4.16}$$

Notice that b_{st}^{uv} terms are optimized for the Bethe approximation of the vanilla SBM, not for the pairwise mixture model. By simply plugging in the BP marginals, we effectively constrained the the function $g' : V^2 \to \{1,2\}^2$ to assign the same label for each node when generating its $n-1$ independent edges/non-edges, recovering the function $g : V \to \{1,2\}$. As a result, the Kullback-Leibler divergence $\mathbb{D}_{KL}(b_{st}^{uv} \parallel \hat{P}_{st}^{uv})$ is no longer zero.

Recall that $P_{mix}(G \mid q,p)$ is just the likelihood of the one block vanilla SBM. If we also use the MLEs of the vanilla SBM (3.16), the weighted average of $\hat{p}_{st}$ entries is

$$< \widehat{p} >= \sum_{s,t=1}^{2} q_s q_t \widehat{p}_{st} = \sum_{s,t=1}^{2} \frac{(\sum_u b_s^u)(\sum_v b_t^v)}{n^2} \frac{\sum_{(u,v)\in E} b_{st}^{uv}}{(\sum_u b_s^u)(\sum_v b_t^v)} = \frac{m}{n^2}\,.$$

Although this is a different mixture when compared with (4.12), $P_{mix}(G \mid q,p)$ remains the likelihood of a one block vanilla SBM. In fact, it has now becomes the maximum likelihood of the one block vanilla SBM. We can now rewrite (4.16) in the

form of (4.10),

$$\begin{aligned}\Lambda(G) =& \log \frac{\sup_{H_1} \sum_g P(G, g \,|\, q, p)}{\sup_{H_0} \sum_g P(G, g \,|\, q, p)} \\ =& \log P(G \,|\, \hat{q}_1, \hat{p}_{11}, \hat{p}_{12}, \hat{p}_{21}, \hat{p}_{22}) - \log P(G \,|\, \hat{q}_1, \frac{m}{n^2}, \frac{m}{n^2}, \frac{m}{n^2}, \frac{m}{n^2}) \\ =& n(n-1) \log(\hat{q_1}^{\frac{1}{n-1}} + \hat{q_2}^{\frac{1}{n-1}}) - \sum_{u<v} \sum_{st} b_{st}^{uv} \log \frac{b_{st}^{uv}}{\hat{P}_{st}^{uv}} \,.\end{aligned}$$

where we have plugged in the MLEs of the vanilla SBM, and

$$\hat{P}_{st}^{uv} = \frac{q_s^{\frac{1}{n-1}} q_t^{\frac{1}{n-1}} p_{st}^{A_{uv}} (1-p_{st})^{1-A_{uv}}}{\sum_{st} q_s^{\frac{1}{n-1}} q_t^{\frac{1}{n-1}} p_{st}^{A_{uv}} (1-p_{st})^{1-A_{uv}}} \,.$$

Chapter 5

Bayesian Model Selection

In this chapter, I will shift my attention to Bayesian model selection techniques, which are the statistical foundations directly lead to BIC. Bayesian approaches take the whole posterior distribution into account rather than just point estimates, thus achieving the trade-off between bias and variance. Using the efficient algorithms introduced in Chapter 3, I shall again investigate model selection problems through both theoretical and empirical studies. By adapting the theory for network data, I will finally propose new Bayesian model selection methods for block models, including the corrected BIC.

5.1 The Bayesian integration

Frequentist model selection approaches like the LRT introduced in the last chapter, usually relies on finding some point estimates of the parameters. Bayesian approaches, on the other hand, are based on the posterior of the models, which takes the whole distribution of the parameters into account [20]. These posteriors distri-

butions, by the Bayes' theorem, have an intuitive interpretation for model selection,

$$P(M_i \mid G) = \frac{P(M_i)}{P(G)} P(G \mid M_i) \propto \iiint_0^1 P(G \mid M_i, \Pi_i) P(\Pi_i \mid M_i) \mathrm{d}\Pi_i \,, \tag{5.1}$$

where I have assumed that the prior probability of models $P(M_i)$ is uniform, and the total evidence of data $P(G)$ is constant.

These are exactly the same assumptions we made in the derivation of BIC 2.2.2 (assumptions B.1). The posterior $P(M_i \mid G)$ is thus proportional to the total likelihood $P(G \mid M_i)$, which is intuitively the integrated conditional likelihood given parameter values $P(G \mid M_i, \Pi_i)$ over the prior $P(\Pi_i \mid M_i)$ of the parameter set Π_i. This total likelihood term will be the target for Bayesian model selection in the following sections.

Ironically, model selection is inherently an contradiction to the Bayesian philosophy of not making point estimates. A full Bayesian approach would give a posterior distribution for all candidate models, rather a single preferred choice. However, compared with the frequentist methods in last chapter, it still makes sense to calling these maximum a posteriori (MAP) methods Bayesian.

5.1.1 Bayes factor

The Bayes factor is a Bayesian extension to the classical LRT (4.1). Instead of taking the supremum of conditional likelihood functions, the Bayes factor uses the whole posterior for the ratio test:

$$\Lambda_{Bayes}(G) = \log \frac{\iiint_0^1 P(G \mid M_1, \Pi_1) P(\Pi_1 \mid M_1) \mathrm{d}\Pi_1}{\iiint_0^1 P(G \mid M_0, \Pi_0) P(\Pi_0 \mid M_0) \mathrm{d}\Pi_0} \,. \tag{5.2}$$

Here the null model M_0 and its alternative M_1 does not need to be nested like the LRT. In fact, the Bayes factor works for any pair of models as long as both have valid total likelihoods, even if they do not share any parameters and take totally different forms of likelihood functions [20].

The Bayes factor works just like the LRT. We reject the null model in favor of the alternative when Λ_{Bayes} exceeds some threshold. This threshold, again, depends on your tolerance of error, as well as the null distribution of Λ_{Bayes}.

Just like the LRT, one criticism of Bayes Factors is that it only works for pairwise model comparisons. Another problem with Bayes inference in general is that it depends heavily on the choice of prior $P(\Pi_i \mid M_i)$, and most of the priors lead to intractable integrals.

5.1.2 Bayesian information criterion

One key advantage of Bayesian model selection is its flexibility for models of any form, provided that the posterior can be calculated. This makes it a good framework for universal model selection. To compare among any number of arbitrary models, however, we need to choose a common confidence interval for all models, sacrificing our ability in setting the margins of error. Furthermore, to ensure the tractability of posteriors, I shall restrict our choice of prior to conjugate priors of the likelihood functions.

One such Bayesian model selection method is the BIC we have seen previously 2.2.2:

$$BIC(M_i) = -2 \ln P(Y \mid M_i, \hat{\Pi}_i) + |\Pi_i| \ln n \,, \tag{5.3}$$

where $|\Pi_i|$ is the degree of freedom of the model M_i with a parameter set $|\Pi_i|$, and n is number of i.i.d. samples in the data.

The above simple formulation with a maximized likelihood and a penalty term for model complexity, deceivingly, is a large sample approximation to twice the logarithm of the total likelihood (5.1). However, as we have already seen in the previous chapter, the assumption of large data limit does not always hold for sparse networks. The next

few sections will revisit the derivation of BIC for network data, and investigate its connection with another model selection criterion we have seen earlier: the Minimum Description Length principle (MDL).

5.2 BIC for SBM and its connection to MDL

In this section I will derive the correct approximation to (5.1) for the vanilla SBM. Following the formulation of BIC, we shall arrive at a maximized likelihood plus a penalty term. I will also investigate its connection with MDL using a special coding scheme.

For mathematical convenience, I represent our network as the directed SBM (2.17). By Bayes' theorem, we have the posterior of a SBM M_i with the parameters $\{q, p\}$:

$$\begin{aligned} P(M_i \,|\, G) =& \frac{P(M_i)}{P(G)} P(G \,|\, M_i) \\ \propto& \sum_g \iiint_0^1 \mathrm{d}\{p_{st}\}\mathrm{d}\{q_s\}\, P(G, g \,|\, q, p)\ , \end{aligned} \tag{5.4}$$

where I have followed the same assumptions as before (assumptions B.1). In this full Bayesian framework, the total likelihood $P(G \,|\, M_i)$ is integrated over the prior distributions on both p and q entries, as well as summed over all possible latent states g.

One key design problem for Bayesian model selection is how to balance between bias and variance. The trade-off is achieved by carefully choosing the parameters that are to be integrated over. Depending on the bias in the learning task, as well as the variance in application domains, some of the parameters might not need to be integrated. For example, if the learning task is to find the SBM with the most likely parametrization, regardless of any specific block assignments, the integral over parameters is not necessary. This is the approach I took in the previous chapter.

Alternatively, if the learning task is to find the SBM with the most likely ground state, we do not need the sum over the latent state g, and benefit from the smaller variance because of the bias we are willing to assume. However, if we plan to apply the learned SBM to other networks with variance in p and q entries, the integral over them remains essential. This corresponds to the idea of *Universal Coding* in MDL 2.4.2, where a code has to achieve good compression for any data generated by the same model with different parameters. I used this model for the active learning algorithm 3.6, and I will keep it same here in this chapter for the connection with existing MDL methods for block models.

For block models, a key assumption during the derivation of BIC (assumptions B.1 2.2.2) is violated by the discrete latent state g. One solution I have proposed in the previous chapter is to sum it over using the variational EM framework with the linear BP as the E-step. Please refer to Appendix B.2 for an example of this full Bayesian approach. In this chapter, however, with the sum forgone, we need to pay extra attention in deriving the correct BIC for block models.

If I assume that the p_{st} and q_s entries are independent, with the sole constrain $\sum_s q_s = 1$, and they follows their respective conjugate Dirichlet and Beta priors, we have the Bayesian posterior of a SBM given the graph G and the block assignment g:

$$
\begin{aligned}
P(M_i \,|\, G, g) \propto P(G, g \,|\, M_i) &= \iiint_0^1 \mathrm{d}\{p_{st}\}\mathrm{d}\{q_s\}\, P(G, g \,|\, q, p) \\
&= \left(\int_{\triangle} \mathrm{d}q \mathrm{Dirichlet}(\vec{q}|\vec{\delta}) \prod_{s=1}^{k} q_s^{n_s} \right) \\
&\quad \left(\prod_{s,t=1}^{k} \int_0^1 \mathrm{d}p_{st}\, \mathrm{Beta}(p_{st}|\alpha,\beta)\, p_{st}^{m_{st}} (1-p_{st})^{n_s n_t - m_{st}} \right)
\end{aligned}
$$

$$
\begin{aligned}
&= \left(\frac{\Gamma(\sum_{s=1}^{k} \delta_s)}{\prod_{s=1}^{k} \Gamma(\delta_s)} \int_{\triangle} \mathrm{d}q \prod_{s=1}^{k} q_s^{n_s+\delta_s-1} \right) \\
&\quad \left(\prod_{s,t=1}^{k} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \int_0^1 \mathrm{d}p_{st}\, p_{st}^{m_{st}+\alpha-1} (1-p_{st})^{n_s n_t - m_{st}+\beta-1} \right) \\
&= \left(\frac{\Gamma(\sum_{s=1}^{k} \delta_s)}{\prod_{s=1}^{k} \Gamma(\delta_s)} \frac{\prod_{s=1}^{k} \Gamma(n_s+\delta_s)}{\Gamma(\sum_{s=1}^{k}(n_s+\delta_s))} \right) \\
&\quad \left(\prod_{s,t=1}^{k} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\,\Gamma(\beta)} \frac{\Gamma(m_{st}+\alpha)\,\Gamma(n_s n_t - m_{st}+\beta)}{\Gamma(n_s n_t+\alpha+\beta)} \right) \\
&= P(V, g \,|\, M_i) \times P(E, g \,|\, M_i)\ , \qquad (5.5)
\end{aligned}
$$

where I assumed the same beta prior $\{\alpha, \beta\}$ for all the p_{st} entries, and applied the Euler integral of the first kind, including its multinomial generalization on the simplex $\sum_s q_s = 1$. As equation (5.5) shows, the total likelihood factors into terms for nodes and edges.

To get an idea of the posterior distribution $P(M_i \,|\, G, g)$, I assume the data g follows a uniform prior over random graphs generated by a SBM with 5 prescribed blocks. This is a generalization to the assumptions we have made (assumptions B.1), but the total likelihood remains proportional to the posterior. For simplicity, I have also plugged in the uniform priors (i.e., $\delta_{\forall s} = 1, \alpha = \beta = 1$) for the parameters, just like I did it in Section 3.6,

$$
\begin{aligned}
&P(M_i \,|\, G, g) \propto P(G, g \,|\, M_i) \\
&= \left((k-1)! \frac{\prod_{s=1}^{k} n_s!}{(n+k-1)!} \right) \left(\prod_{s,t=1}^{k} \frac{m_{st}!(n_s n_t - m_{st})!}{(n_s n_t+1)!} \right)\ . \qquad (5.6)
\end{aligned}
$$

The distributions of the posterior with different number of blocks are shown in Figure 5.1. While the SBM with correct number of blocks (red) does has slightly higher likelihood in average, it overlaps quite heavily with SBMs with fewer blocks (green) or more blocks (blue). But further investigation reveals that most of the variance came from the variance in data prior. Once we enforce the constant data

assumption, i.e. fixing the input graph for all the candidate models, the correct SBM always has a higher likelihood than the others, as illustrated in Figure 5.2.

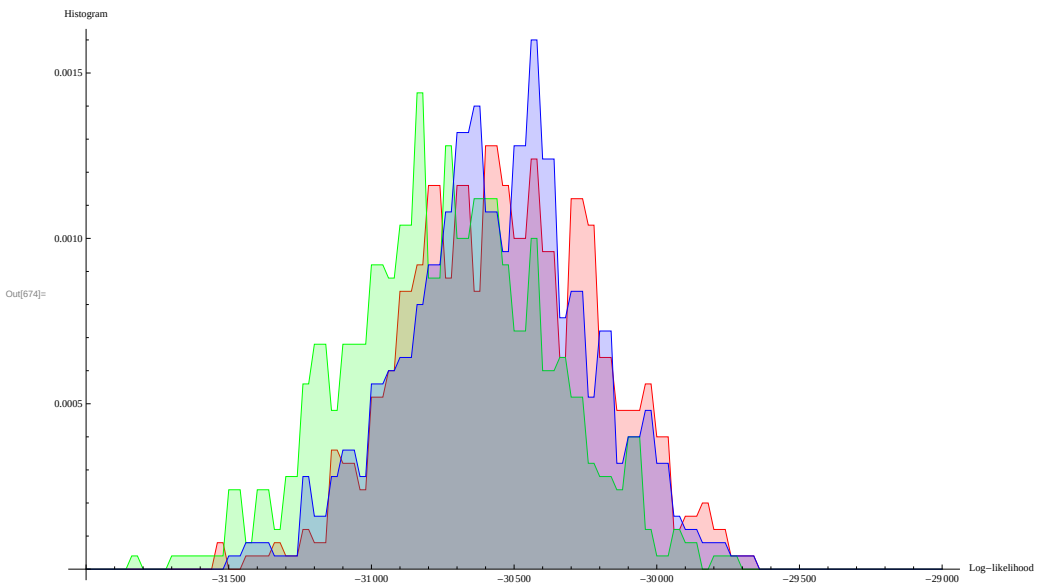


Figure 5.1: The distribution of log-likelihoods of the Bayesian model (equation (5.5)) *The distributions are gathered from randomly generated SBMs with 1000 nodes and 5 prescribed blocks. SBMs with different number of blocks k are fitted to the data. Specifically, the green distribution is from a SBM with $k = 1$, the red with $k = 5$ and the blue with $k = 10$. The experiment is done using a Monte Carlo sampling method, with 500 samples for each SBM.*

5.2.1 Bayesian code for SBM

According to Grünwald [35], Bayesian approach for model selection has a close relation to the minimum description length principle. In particular, if we choose the Jeffreys priors for p_{st} and q_s entries (i.e., $\alpha = \beta = \delta_{\forall s} = 1/2$), the coding length according to the Bayesian model is asymptotically the same as the optimal minimax coding. Grünwald also pointed out in [35], while the Jeffreys priors lead to the shortest coding, other priors and their corresponding non-optimal coding can still produce description with length of the same asymptotic order, as long as the prior

is dominated by the evidence.

Knowing prior choice is flexible, I will use the simple form with uniform priors (5.6), we have:

$$\begin{aligned} & P(G, g \,|\, M_i) = P(V, g \,|\, M_i) \times P(E, g \,|\, M_i) \\ &= \left((k-1)! \frac{\prod_{s=1}^{k} n_s!}{(n+k-1)!} \right) \left(\prod_{s,t=1}^{k} \frac{m_{st}!(n_s n_t - m_{st})!}{(n_s n_t + 1)!} \right) \\ &= \left(\frac{1}{\binom{n+k-1}{k-1}} \frac{1}{\binom{n}{(n_1, n_2, \ldots, n_k)}} \right) \left(\prod_{s,t=1}^{k} \frac{1}{\binom{n_s n_t}{m_{st}} (n_s n_t + 1)} \right) . \end{aligned} \tag{5.7}$$

The leading combinatorial terms in equation (5.7) lead to a Bayesian universal code for a graph G consists of the following parts:

1. number of blocks k ($\log k$ bits, implicit)
2. code for the partition of n into n_s terms ($\log \binom{n+k-1}{k-1}$ bits)
3. code for the block assignment given the n_s terms ($\log \binom{n}{(n_1, n_2, \ldots, n_k)}$ bits)
4. for each pair of blocks s, t, the number of edges m_{st} between them ($\log m_{st} < \log(n_s n_t + 1)$ bits)
5. for each pair of blocks s, t, code for the edge allocations given m_{st} ($\log \binom{n_s n_t}{m_{st}}$ bits)

According to [35], there is a correspondence between probability distributions and prefix codes. In the above coding scheme, the distribution of possible realizations in part i ($i > 1$) conditioned on all previous code parts are all uniform, therefore the optimal code length for part i can be quantified by the negative logarithm of the corresponding combinatorial terms in equation (5.7).

While the above coding scheme gives an intuitive connection to the Bayesian integration, popular MDLs are usually defined in terms of the most likely estimators

(MLEs) or equivalently, the entropy minimizers of the likelihood functions. In the following subsections, I shall prove the mathematical equivalence between the BIC measure based on (5.5) and the MDL for SBMs as defined in [67].

5.2.2 Corrected BIC for order selection in SBM

The keys to transform the integrals in (5.5) to the BIC formulation (5.3) are the uniform priors and Laplace's approximation. If the integrals are tightly peaked around their most likely value, by applying Laplace's method, or equivalently by using Stirling's formula for the factorials in (5.7), we should have a close approximation:

$$
\begin{aligned}
& P(G, g \,|\, M_i) = P(V, g \,|\, M_i) \times P(E, g \,|\, M_i) \\
=& \frac{(k-1)!n!}{(n+k-1)!} \frac{\prod_{s=1}^{k} n_s!}{n!} \prod_{s,t=1}^{k} \frac{m_{st}!(n_s n_t - m_{st})!}{n_s n_t!(n_s n_t + 1)} \\
\approx& \frac{\prod_{s=1}^{k} \sqrt{2\pi n_s}}{\binom{(n+k-1)}{n} \sqrt{2\pi n}} \prod_{u} \frac{n_{g(u)}}{n} \\
& \prod_{s,t=1}^{k} \frac{2\pi \sqrt{m_{st}(n_s n_t - m_{st})}}{\sqrt{2\pi (n_s n_t)}(n_s n_t + 1)} \prod_{u<v,(u,v)\in E} \frac{m_{g(u)g(v)}}{n_{g(u)} n_{g(v)}} \prod_{u<v,(u,v)\notin E} (1 - \frac{m_{g(u)g(v)}}{n_{g(u)} n_{g(v)}}) \\
\approx& P(V, g \,|\, \hat{q}) \frac{\prod_{s=1}^{k} \sqrt{2\pi n_s}}{\binom{(n+k-1)}{n} \sqrt{2\pi n}} \times P(E, g \,|\, \hat{p}) \prod_{s,t=1}^{k} \frac{\sqrt{2\pi}}{\sqrt{\frac{n_s^3 n_t^3}{m_{st}(n_s n_t - m_{st})}}} \,, \qquad (5.8)
\end{aligned}
$$

where I plugged in the MLEs $\hat{q_s} = \dfrac{n_s}{n}$ and $\hat{p_{st}} = \frac{m_{st}}{n_s n_t}$.

If we take the negative log of (5.8). The factor associated with E becomes the term:

$$
\begin{aligned}
-\ln P(E, g \,|\, M_i) &\approx -\ln P(E, g \,|\, \hat{p}) - \sum_{s,t=1}^{k} \frac{1}{2} \ln \frac{2\pi m_{st}(n_s n_t - m_{st})}{n_s^3 n_t^3} \\
&\approx -\ln P(E, g \,|\, \hat{p}) + \frac{k^2}{2} \ln \frac{n^6}{2\pi |E|(n^2 - |E|)} - C \,, \qquad (5.9)
\end{aligned}
$$

where I made a mean-field assumption about m_{st} under constant number of blocks k. If the graph is sparse, as $|E| = \rho n$, we have

$$\begin{aligned} -\ln P(E, g \,|\, M_i) &\approx -\ln P(E, g \,|\, \hat{p}) + \frac{k^2}{2} \ln \frac{n^6}{2\pi\rho n^2 (n-\rho)} \\ &\approx -\ln P(E, g \,|\, \hat{p}) + \frac{k^2}{2} \ln \Theta(n^3) \ . \end{aligned}$$

If the graph is dense, as $|E| = \rho n^2$, we have

$$\begin{aligned} -\ln P(E, g \,|\, M_i) &\approx -\ln P(E, g \,|\, \hat{p}) + \frac{k^2}{2} \ln \frac{n^6}{2\pi\rho n^4 (1-\rho)} \\ &\approx -\ln P(E, g \,|\, \hat{p}) + \frac{k^2}{2} \ln \Theta(n^2) \ . \end{aligned}$$

Putting it together with the term associated with V, in which I again assumed mean-field n_s terms, we get

$$\begin{aligned} & -\ln P(G, g \,|\, M_i) = -\ln P(V, g \,|\, M_i) - \ln P(E, g \,|\, M_i) \\ \approx & -\ln P(V, g \,|\, \hat{q}) + \Theta(k \ln n) - \ln P(E, g \,|\, \hat{p}) + \frac{k^2}{2} \ln \Theta(n^2) \\ = & -\ln P(G, g \,|\, \hat{p}, \hat{q}) + \frac{k^2}{2} \ln \Theta(n^2) \ . \end{aligned} \tag{5.10}$$

Finally, multiply it by 2, we have the BIC for order selection in dense SBMs:

$$BIC_{SBM}(M_i) = -2 \ln P(G, g | M_i, \hat{\Pi}_i) + k^2 \ln \Theta(n^2) \ , \tag{5.11}$$

which is simply the direct application of BIC to directed block models, with k^2 specifying the number of parameters in the block affinity matrix p and n^2 represent the sample size as pairwise edge/non-edge interactions.

Similarly, in sparse SBMs, the BIC for order selection is:

$$BIC^*_{SBM}(M_i) = -2 \ln P(G, g | M_i, \hat{\Pi}_i) + k^2 \ln \Theta(n^3) \ , \tag{5.12}$$

where the penalty term becomes even greater, favoring simpler models to compensate for fewer data samples in sparse networks.

5.2.3 Mathematical comparison with MDL

Many MDL measures has been proposed for networks [74, 75, 67]. In [67], the authors adopted a universal code for the vanilla SBM corresponds to the Bayesian integration in (5.5), and designed a code with description length of Σ_t :

$$\Sigma_t \approx -\ln P(E, g \,|\, \hat{p}) + \frac{(k+1)k}{2} \ln |E| + n \ln k \tag{5.13}$$

where I have made the simplifying assumption that $|E| \gg k^2$, and replaced notations according to our convention.

This is asymptotically the same as the BIC defined in (5.11). To see this, rewrite (5.10) as

$$\begin{aligned}
&\frac{1}{2} BIC_{SBM}(M_i) \approx -\ln P(V, g \,|\, M_i) - \ln P(E, g \,|\, M_i) \\
\approx &-\ln P(V, g \,|\, \hat{q}) + \Theta(k \ln n) - \ln P(E, g \,|\, \hat{p}) + \frac{k^2}{2} \ln \Theta(n^2) \\
= &-\ln P(E, g \,|\, \hat{p}) + \frac{k^2}{2} \ln \Theta(n^2) + \Theta(n \ln k) \ .
\end{aligned}$$

Specifically, the first term corresponds to the description length of the graph given the parameters. The second term accounts for the description length of the p_{st} matrix. The third term of $n \ln k$, which accounts for the code length for block assignment, appears from the terms $-\ln P(V, g \,|\, \hat{q})$ with a mean-field assumption.

However, if the mean-field assumptions are violated, the optimal coding scheme according to BIC, i.e. a Bayesian model with uniform priors, should leads to shorter descriptions. This is confirmed by the experiment result shown in Figure5.2, where I have intentionally made one block bigger than others.

Although by theory the optimal code with Jeffreys priors could produce even better information compression, Grünwald claimed that all reasonable priors should produce description length of the same asymptotic order [35]. This means our Bayesian

measure with uniform priors, as well as the MDL in [67] are still practically sound criteria for SBM model selection.

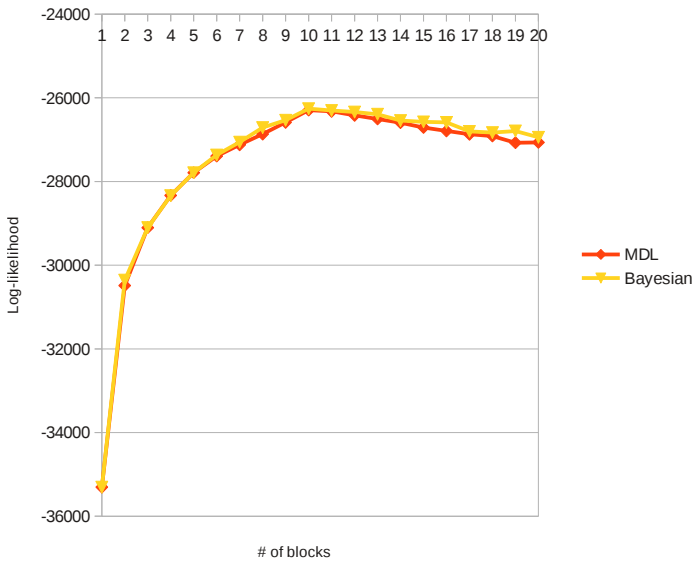


Figure 5.2: Log-likelihood (or negative description length) of the Bayesian model (equation (5.10)) compared with those of the MDL model in [67] (equation (5.13)) *The experiment is setup the same way as those in Figure.1 of [67]. The algorithms run on the same SBM with 10000 nodes and 10 prescribed blocks. To demonstrate that the Bayesian model and its corresponding coding scheme has a slight advantage, I did intentionally make one of the block 4 times as big as the rest. Both the Bayesian and MDL model achieves the highest likelihood, or shortest description length at the right number of blocks. The experiment is done using a Monte Carlo sampling method.*

5.3 BIC for DC-SBM

Now I will generalize the BIC formulations to DC-SBMs, and compared it to the frequentist method introduced in 4.2. Through the inherent connection between BIC and MDL, this would also lead to a coding scheme for these models.

For mathematical convenience, I use the directed DC-SBM (2.19) as the model. By Bayes' theorem, we have the posterior of a DC-SBM M_i with the parameters

$\{\theta, \omega, q\}$:

$$\begin{aligned} P_{DC}(M_i \,|\, G) =& \frac{P(M_i)}{P(G)} P_{DC}(G \,|\, M_i) \\ \propto& \sum_g \iiint_0^1 \mathrm{d}\{\theta_u\}\mathrm{d}\{\omega_{st}\}\mathrm{d}\{q_s\}\, P(G, g \,|\, \theta, \omega, q) \;, \end{aligned} \tag{5.14}$$

where I again assumed that the prior probability of models $P(M_i)$ is uniform, and the total evidence of data $P(G)$ is constant. The total Bayesian likelihood integrates over the prior distribution on θ, p and q entries, as well as summed over all possible latent states g. For the connection to MDL, here I simply forgo the sum and only focus on a given block assignment g.

If I assume that the θ_u, p_{st} and q_s entries are independent, with the constrains $\sum_{u:g_u=s} \theta_u = n_s$ and $\sum_i q_s = 1$, we have the Bayesian posterior of a DC-SBM given the graph G and the block assignment g :

$$\begin{aligned} P_{DC}(M_i \,|\, G, g) \propto & P_{DC}(G, g \,|\, M_i) = \iiint_0^1 \mathrm{d}\{\theta_u\}\mathrm{d}\{\omega_{st}\}\mathrm{d}\{q_s\}\, P(G, g \,|\, \theta, \omega, q) \\ =& \left(\int_{\triangle} \mathrm{d}\{\theta_u\} \prod_u \theta_u^{d_u} \right) \iiint_0^1 \mathrm{d}\{\omega_{st}\}\mathrm{d}\{q_s\}\, P(G, g \,|\, \omega, q) \\ =& P(\Theta, g \,|\, M_i) \times P_{Poisson}(G, g \,|\, M_i) \\ \approx& P(\Theta, g \,|\, M_i) \times P(G, g \,|\, M_i) \;, \end{aligned} \tag{5.15}$$

where I have made the approximation that the Poisson-SBM is asymptotically the same as the vanilla SBM (see sections 2.3.2). As equation (5.15) shows, the total likelihood of the DC-SBM has one additional factor compared with the vanilla (Poisson) SBM.

To prepare the extra factor $P(\Theta, g \,|\, M_i)$ for Bayesian treatments, I first change the variables $\theta_u = n_{g(u)}\eta_u$ in the first integral, making the integrand a proper multinomial

distribution. Now if the new parameters η_u follow their Dirichlet conjugate priors,

$$\begin{aligned}
P(\Theta, g \,|\, M_i) =& \prod_{s=1}^{k}\left(\int_{\triangle} \mathrm{d}\{\theta_u\} \prod_{g(u)=s} (\frac{\theta_u}{n_s})^{d_u}\right) \times \prod_u n_{g(u)}^{d_u} \\
\approx& \prod_{s=1}^{k}\left(\int_{\triangle} \mathrm{d}\{\eta_u\} \prod_{g(u)=s} \eta_u^{d_u}\right) \times \prod_u n_{g(u)}^{d_u+1} \\
=& \prod_{s=1}^{k}\left(\int_{\triangle} \mathrm{d}\eta \mathrm{Dirichlet}(\vec{\eta_s}|\vec{\gamma_s}) \prod_{g(u)=s} \eta_u^{d_u}\right) \times \prod_u n_{g(u)}^{d_u+1} \\
=& \prod_{s=1}^{k}\left(\frac{\Gamma(\sum_{g(u)=s}\gamma_u)}{\prod_{g(u)=s}\Gamma(\gamma_u)} \int_{\triangle} \mathrm{d}\eta_s \prod_{g(u)=s} \eta_u^{d_u+\gamma_u-1}\right) \times \prod_u n_{g(u)}^{d_u+1} \\
=& \prod_{s=1}^{k}\left(\frac{\Gamma(\sum_{g(u)=s}\gamma_u)}{\prod_{g(u)=s}\Gamma(\gamma_u)} \frac{\prod_{g(u)=s}\Gamma(d_u+\gamma_u)}{\Gamma(\sum_{g(u)=s}(d_u+\gamma_u))}\right) \times \prod_u n_{g(u)}^{d_u+1}\,, \qquad (5.16)
\end{aligned}$$

where I applied the multinomial Euler integral on the simplex $\sum_{u:g_u=s}\eta_u = 1$. Now, if I again assume the priors are uniform (i.e., $\gamma_{\forall u} = 1$), we have:

$$\begin{aligned}
P(\Theta, g \,|\, M_i) =& \prod_{s=1}^{k}\left((n_s-1)!\frac{\prod_{g(u)=s} d_u!}{(D_s+n_s-1)!}\right) \times \prod_u n_{g(u)}^{d_u+1} \\
=& \prod_{s=1}^{k}\left(\frac{(n_s-1)!D_s!}{(D_s+n_s-1)!}\frac{\prod_{g(u)=s} d_u!}{D_s!}\right) \times \prod_u n_{g(u)}^{d_u+1} \\
\approx& \prod_{s=1}^{k}\left(\frac{\prod_{g(u)=s}\sqrt{2\pi d_u}}{\binom{(D_s+n_s-1)}{D_s}\sqrt{2\pi D_s}} \prod_{g(u)=s} (\frac{d_u}{D_s})^{d_u}\right) \times \prod_u n_{g(u)}^{d_u+1} \\
=& P(\Theta, g \,|\, \hat{\eta}) \prod_{s=1}^{k}\left(\frac{\prod_{g(u)=s}\sqrt{2\pi d_u}}{\binom{(D_s+n_s-1)}{D_s}\sqrt{2\pi D_s}}\right) \times \prod_u n_{g(u)}^{d_u+1}\,, \qquad (5.17)
\end{aligned}$$

where $D_s = \sum_{g(u)=s} d_u$ is the total degree of nodes in block s. Here I applied the Stirlings formula for factorials, and plugged in the MLEs $\eta_u = \frac{\theta_u}{n_{gu}} = \frac{d_u}{D_{g(u)}}$.

Now we put back the factors from the vanilla SBM (5.10), and take the logarithm

of it, we have the BIC formulation for DC-SBM:

$$\begin{aligned}
&-\frac{1}{2}BIC_{DC}M_i = \ln P_{DC}(G, g \,|\, M_i) = \ln P(\Theta, g \,|\, M_i) + \ln P(G, g \,|\, M_i) \\
\approx &\ln P(\Theta, g \,|\, \hat{\eta}) + \ln P(E, g \,|\, \hat{p}) - \frac{k^2}{2}\ln\Theta(|E|) + \ln P(V, g \,|\, \hat{q}) - \Theta(k\ln k) \\
&+ O(k\ln n) + \sum_{s=1}^{k}\left(\Theta(\frac{n_s}{2}\ln(\frac{1}{n_s})) - \Theta(n_s\ln(D_s + n_s)\right) + \sum_u (d_u+1)\ln n_{g(u)} \\
= &\ln P(G, g \,|\, \hat{\eta}, \hat{q}, \hat{p}) - \frac{k^2}{2}\ln\Theta(|E|) - \Theta(n\ln n) - \Theta(n\ln|E|) + \Theta(|E|\ln n)\,.
\end{aligned} \tag{5.18}$$

5.3.1 Mathematical comparison with LRT

To confirm its correctness, I compare the BIC measure under the model selection problem of SBM vs DC-SBM, with the LRT in section 4.2. We can rewrite (5.10) and (5.18) as:

$$\begin{aligned}
\ln P(G, g \,|\, 1, \hat{q}, \hat{p}) &\approx \ln P_{SBM}(G, g \,|\, M_i) + \frac{k^2}{2}\ln\Theta(|E|)\,, \\
\ln P(G, g \,|\, \hat{\theta}, \hat{q}, \hat{p}) &\approx \ln P_{DC}(G, g \,|\, M_i) + \frac{k^2}{2}\ln\Theta(|E|) \\
&\quad + \Theta(n\ln n) + \Theta(n\ln|E|)\,.
\end{aligned}$$

Therefore, following the construction of Bayes factors 5.1.1, we have the Bayesian version of the log-likelihood ratio,

$$\begin{aligned}
\Lambda_{DC}(G, g) &= \ln P(G, g \,|\, \hat{\theta}, \hat{q}, \hat{p}) - \ln P(G, g \,|\, 1, \hat{q}, \hat{p}) \\
&\approx \ln P(\Theta, g \,|\, M_i) + \Theta(n\ln n) + \Theta(n\ln|E|) \\
&\approx \ln P(\Theta, g \,|\, \hat{\theta})\,,
\end{aligned}$$

which is the same as the log-likelihood ratio test statistics we have in (4.3). The agreement between Bayesian and Frequentist methods is not a coincident, because we have used uniform priors in our derivation. This is also very similar to the code

length function used in [67]. Again, an intuitive Bayesian code in the next subsection will reveal that it is not merely a mathematical equivalence.

5.3.2 Bayesian code for DC-SBM

As we did in 5.2.1, here I will propose a intuitive Bayesian code for DC-SBM. Since the total likelihood of the DC-SBM factors (5.15) and we already the code for factor $P(G, g \,|\, M_i)$ corresponding to the vanilla SBM, the following code is only for the extra factor $P(\Theta, g \,|\, M_i)$.

Again assuming uniform priors (i.e., $\gamma_{\forall u} = 1$), we have:

$$\begin{aligned} P(\Theta, g \,|\, M_i) =& \prod_{s=1}^{k} \left(\frac{(n_s - 1)! D_s!}{(D_s + n_s - 1)!} \frac{\prod_{g(u)=s} d_u!}{D_s!} \right) \times \prod_u n_{g(u)}^{d_u+1} \\ =& \prod_{s=1}^{k} \left(\frac{1}{\binom{D_s+n_s-1}{n_s-1}} \frac{1}{\binom{D_s}{(d_{u_1^s}, d_{u_2^s}, \ldots, d_{u_{n_s}^s})}} \right) \times \prod_s n_s^{D_s+n_s} . \end{aligned} \tag{5.19}$$

The leading combinatorial terms above lead to a Bayesian universal code for $P(\Theta, g \,|\, M_i)$:

1. number of blocks k ($\log k$ bits, implicit)
2. for each block s, the total degree D_s ($\log D_s$ bits, implicit)
3. for each block s, code for the partition of D_s into n_s terms ($\log \binom{D_s+n_s-1}{n_s-1}$ bits)
4. for each block s, code for the degree assignment given the degree sequence ($\log \binom{D_s}{(d_{u_1^s}, d_{u_2^s}, \ldots, d_{u_{n_s}^s})}$ bits)
5. For each block s, the negative code for uniformly randomly assigning each degree to each node ($-(D_s + n_s) \log n_s$ bits)

Notice that the last part of the code is negative, and it is effectively the code needed if the degree sequence in each block is totally random, being totally ignorant of degrees just as the vanilla SBM does. We can interpret the last part of the code as the canceling factor going from the vanilla SBM to DC-SBM. As a result, if we just use the same Bayesian code for the vanilla SBM 5.2.1 in addition to the above code, we will have the Bayesian code for DC-SBM.

Chapter 6

Conclusions and Future Work

Model selection is very important for building better models with both efficient learning processes and accurate generalization performances. For stochastic block models, however, it remains an open problem because traditional model selection methods no longer work properly in the realm of network data, and many classic statistical tools need to be corrected for sparse graphs.

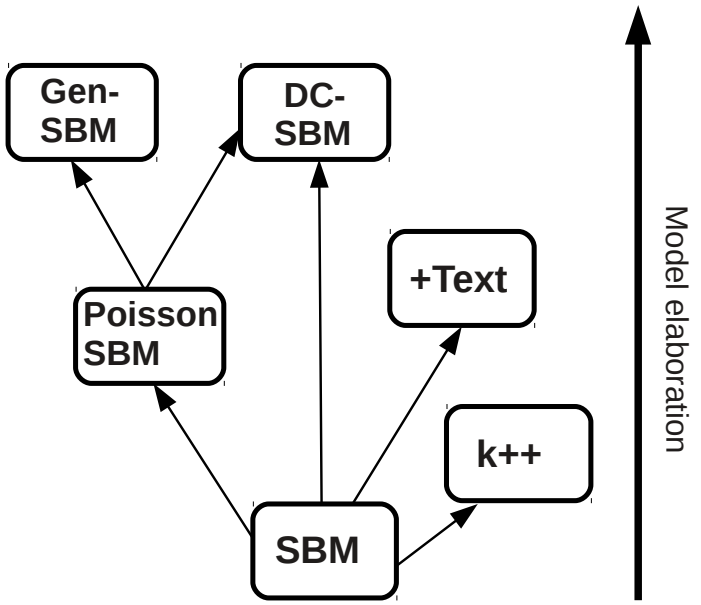


Figure 6.1: The network of complexity hierarchy of variants of block models
The model on the origin of an edge in this network is strictly a special case of the model on the target. They form a partial order with simpler models at the bottom. $k++$ means a SBM with more blocks, which can also be applied to all the other variants. For details of the +Text SBM, please refer to our paper [93]. Gen-SBM is defined in the paper [2].

In this dissertation, I targeted two specific model selection problem in the hierarchy of block models (Figure 6.1). The first is to choose between the Poisson-SBM and DC-SBM models. I approached this pair of nested models using classic frequentist statistics. Based on simulations made possible by my scalable algorithms, I corrected the classic theory, and proposed a corrected AIC for this particular model selection problem:

$$AIC_{DC}(M_i) = -2\ln P(G|M_i, \hat{\Pi}_i) + 2(1 + \frac{1}{6\mu} + \frac{1}{6\mu^2} + O(\frac{1}{\mu^3}))|\Pi_i|\,,$$

where Π_i is the parameter set for model M_i, and μ is the average degree of the network. I later confirmed this result using a Bayesian approach.

The second problem is choosing the right number of blocks for the vanilla SBM.

To tackle this important but surprisingly difficult problem, I adopted Bayesian model selection techniques. By careful choices of assumptions and priors, I not only uncovered the deep connection between BIC and MDL, but also analytically solved this order selection problem, ultimately leading to the corrected BIC for the task

$$BIC_{SBM}(M_i) = -2 \ln P(G, g|M_i, \hat{\Pi}_i) + k^2 \ln \Theta(n^2) \ ,$$

with k being the number of blocks and n being the size of the network. Again, sparse networks have a slight bigger penalty term:

$$BIC^*_{SBM}(M_i) = -2 \ln P(G, g|M_i, \hat{\Pi}_i) + k^2 \ln \Theta(n^3) \ . \tag{6.1}$$

These problems serve as examples for two groups of very different model selection problems in the hierarchy. One group is for choosing between the vanilla SBM and those like the DC-SBM, with node attributes that participant in edge generation. The other is the order selection problem for each block model variant. I expect the statistical approaches I used here can be respectively generalized to these similar situations. From a more general perspective, my work here opens the way for applying these statistical tools in a wide range of network problems. With an efficient algorithmic framework like the variational EM with BP, one can replicate the process of doing bootstrapping simulations, analyzing data and correcting theories for many other model selection problems.

Armed with the knowledge of model selection for networks in general, I hope to build more sophisticated models in the future that not only are capable of generating complex network structures, but at the same time are statistically well-defined with low risk of over-fitting. As the size and quality of network data sets keep to grow, and our domain knowledge improves, I look forward to a flexible framework armed with the full statistical arsenal, capable of selecting or even automatically generating appropriate models given the data and inference task.

Appendices

Appendix A

Other constructions of the BP algorithm

A.1 BP as a partition function construction

Besides the Bethe approximation, we can construct the partition function by recursive addition of nodes [23]. Starting with an existing graph G, with a partition function:

$$Z_G = \sum_{\{g(G)\}} P(G, g|\theta)$$

Let $G^- = G \setminus \{u\}$ be the graph without the node u. By assuming conditional independence among the neighbors, we can write Z_G in terms of Z_{G^-}:

$$\begin{aligned} Z_G^t &= \gamma_t \sum_{\{g(G^-)\}} \prod_{w \neq u} f(g(w), t) Z_{G^-}^{g(w)} \\ &= \gamma_t \prod_{w \neq u} \sum_s f(s, t) Z_{G^-}^s \end{aligned} \tag{A.1}$$

where Z_G^t is the partial partition function conditioned on node u being type t.

If you compare (A.1) with (3.13), you will find them to be of a very similar form. Indeed, the messages defined in our BP algorithm can be interpreted as a local partition function constructed based on messages received except the message from the target. In fact, (A.1) has the exact same form as the approximate non-edge messages. Apply the same message normalization, we have

$$\begin{aligned}\mu_G^t &= \frac{Z_G^t}{\sum_t Z_G^t} = \frac{\gamma_t \prod_{w\neq u}\sum_s f(s,t) Z_{G^-}^s}{\sum_t \gamma_t \prod_{w\neq u}\sum_s f(s,t) Z_{G^-}^s} \\ &= \frac{\prod_{w\neq u}\sum_s Z_{G^-}^s}{\sum_t \gamma_t \prod_{w\neq u}\sum_s f(s,t) Z_{G^-}^s} \frac{\gamma_t \prod_{w\neq u}\sum_s f(s,t) Z_{G^-}^s}{\prod_{w\neq u}\sum_s Z_{G^-}^s} \\ &= \frac{1}{Z_u}\gamma_t \prod_{w\neq u}\sum_s f(s,t)\mu_{G^-}^s \end{aligned} \tag{A.2}$$

where

$$Z_u = \frac{\sum_t \gamma_t \prod_{w\neq u}\sum_s f(s,t) Z_{G^-}^s}{\prod_{w\neq u}\sum_s Z_{G^-}^s} = \frac{Z_G}{Z_{G^-}}$$

In other words, the normalizing term in non-edge messages is the growing ratio of the parition function if node u and its induced edges are added. We can estimate the complete partition function as the product of

$$\begin{aligned} Z_G &= \prod_u Z_u = \prod_u \left[\sum_t \gamma_t \prod_{w\neq u}\sum_s f(s,t)\mu_{G_u}^s\right] \\ &= \prod_u \left[\sum_t \xi_t^u(G_u)\right] \end{aligned}$$

where G_u is the growing graph before node u is added.

The above construction, however, is less accurate than the Bethe free energy mentioned in in Section 3.2. With $b_s^u = \mu_s^u$, and a close comparison with (B.4), we can see that it is in fact a mean-field approximation of the total free energy, which is estimated as the sum of node free energies (including its induced edges). To go beyond the fist order of Kikuchi, we need a better formulation of the BP algorithm.

A.2 The factor graph (sum-product) formulation

Suppose a stochastic model with a joint probability distribution that factors into a product of n *local functions* λ_i, each having X_i, some subset of $\{x_1, ..., x_m\}$ as arguments:

$$P(\{x_1, ..., x_m\}) = \prod_{i=1}^{n} \lambda_i(X_i)$$

A *factor graph* is a bipartite graph that express the structure of such a factorization. It has a *variable node* for each variable x_j, and a factor node for each local function λ_i. An Edges would connect a variable node x_j to a factor node λ_i if and only if the former is an argument of the latter.

Many other popular graphical models such as Bayesian networks can be translated into this representation. Furthermore, various famous algorithms on these graphical models can also be reduced to the belief propagation algorithm on the corresponding factor graph [49]. Under the factor graph formulation, we will shortly see why belief propagation is also called the sum-product algorithm.

Accroding to (2.11), our block model has a natural factor graph representation with edge/node generating functions $f(g(u)g(v))$ and $q(g(u))$ as factor nodes and the node type assignment $g(u)$ as the variable nodes. With the orginal graph of Figure 3.1, we would have a corresponding factor graph shown as Figure A.1:

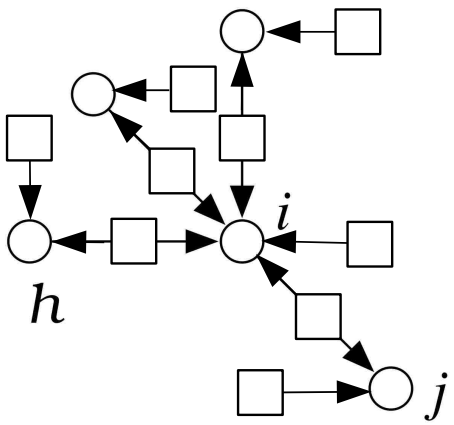


Figure A.1: The factor graph representation of Figure 3.1
Factor nodes are represented as squares while the variable nodes circles. Note that all edge generating factors have 2 variables, whereas all node generating factors have just 1.

The message update equation, originally (3.13), can now be break into the "sum" step and the "product" step which leads to the alternative name of belief propagation.

$$\xi_t^{\lambda\to u} = \begin{cases} \sum_s f(st)\mu_s^{w\to\lambda} & \lambda \in f(g(u)g(v))\&w : n(\lambda)\backslash\{x\} \\ q_t & \lambda \in q(g(u)) \end{cases}$$

$$\xi_t^{u\to\lambda} = \prod_{\eta\in n(x)\backslash\{\lambda\}} \xi_t^{\eta\to u}$$

where $n(x)\backslash\{\lambda\}$ indecates all neighboring factor nodes of variable node $g(u)$ except the target, while $n(\lambda)\backslash\{x\}$ only has one element because the edge/nonedge factors all have 2 variables.

After the messages has converged, we have the marginals of factors:

$$P[\lambda_i(X_i)] = \begin{cases} \sum_{st} f(st) \prod_w \mu_s^{w\to\lambda} & \lambda \in f(g(u)g(v))\&w : n(\lambda) \\ \sum_t q_t\mu_t^{w\to\lambda} & \lambda \in q(g(u))\&w : n(\lambda) \end{cases} \tag{A.3}$$

This local partition function corresponds to F^u_{Bethe} in (3.11).

The normalizing term of the edge generating factors can be converted to:

$$Z^*_{uv} = \sum_{st} f(st) \prod_{w:n(\lambda)} \mu_s^{w\to\lambda} = \frac{\sum_{st} f(st) Z_{G_1}^{g(u)=s} Z_{G_2}^{g(j)=t}}{(\sum_s Z_{G_1}^{g(u)=s})(\sum_t Z_{G_2}^{g(j)=t})} = \frac{Z_{G_1+G_2+(i,j)}}{Z_{G_1+G_2}}$$

Therefore, Z_{uv} is the growing ratio of the partition function if the edge/nonedge (u, v) is added, *excluding* any of its end points. This local partition function corresponds to F^{uv}_{Bethe} in (3.11).

With both local partition functions readily available from convergent messages, BP on factor graphs provides us an easy way of obtaining an estimate of the partition function:

$$Z_G = \prod_{u\neq v} Z^*_{uv} \times \prod_u Z^*_u \tag{A.4}$$

(A.4) follows the intuition of the factor graph formulation. The total probability by definition factors into the product of factor nodes. It might seems odd that the total partition function, which is a sum of such products over states, factors into a product of local sums. However, the conditional independence assumption has made such sum/product swaps possible.

Although (A.4) is intuitive and easy to obtain, it still fails to leverage all the information provided by the second order beliefs b^{uv}_{st}. If we compare it to (3.11),

$$\begin{aligned}\ln Z_G &= \sum_{u\neq v} F^{uv}_{Bethe} + \sum_u F^u_{Bethe} \\ &= \sum_{u\neq v} \sum_{st} b^{uv}_{st}(\ln b^{uv}_{st} - \ln f(s,t)) + \sum_u \sum_s b^u_s(\ln b^u_s - \ln q_s)\end{aligned}$$

we can see that it differs from the full Bethe estimate in the first order terms. The above construction actually assumes a joint belief of

$$B^*(g \,|\, G, q, p) = \prod_{u\neq v} b^{uv}_{st} \prod_u b^u_s$$

This is different from the Bethe assumption (B.5). By using the BP fixed point values and thus the Bethe minimizers instead of its true minimizers, the estimated Z_G in this form is not the optimal approximation to the partition function. In practice, however, since the partition function is usually dominated by second order terms, Z_G would not be too far off either.

Appendix B

Variational EM algorithms for recommendation systems

Please notice that this appendix follows the notations from the paper [36], which is not consist with main body of this dissertation.

B.1 The partition function under given parameters

Here we assume the Q parameters are fixed. Instead of integrating over them, we simply maximize the likelihood with respect to the Q parameter,

$$\begin{aligned}\hat{p}_{SBM}(r_{ui} = r|R^O) =& \frac{1}{Z(\tau)} \sum_{P_U,P_I} \hat{q}_r(\sigma_u, \sigma_i) P(P_U, P_I, R^O|\hat{Q}, \gamma, \eta) \\ =& \frac{1}{Z(\tau)} \sum_{P_U,P_I} P(P_U, P_I, R^O_+|\hat{Q}, \gamma, \eta) \end{aligned} \tag{B.1}$$

where $P(P_U, P_I, R^O|\hat{Q}, \gamma, \eta) = \prod_{\alpha\in P_U} \prod_{\beta\in P_I} \gamma_\alpha^{n_\alpha} \eta_\beta^{n_\beta} \prod_{i=1}^K \hat{q}_i(\alpha, \beta)^{n^i_{\alpha,\beta}}$, and R^O_+ represents the new graph with the r label on the missing link. This would be a close

approximation of (3.23) if the probability density as a function of Q is dominated by the most likely values $\hat{Q}$.

In statistical physics terms the normalization in (B.1) corresponds to the partition function $Z(\tau)$ of the observed graph, at $\tau = 1$ of the Boltzman distribution from which we sample the discrete group assignment variables. The probability density of $P(P_U, P_I|R^O, \hat{Q}, \gamma, \eta)$ under the Boltzman distribution is given by:

$$P(P_U, P_I|R^O, \hat{Q}, \gamma, \eta) = \frac{e^{\tau \ln P(P_U, P_I, R^O|\hat{Q}, \gamma, \eta)}}{\sum_{P_U, P_I} e^{\tau \ln P(P_U, P_I, R^O|\hat{Q}, \gamma, \eta)}} \tag{B.2}$$

Where the denominator is the partition function

$$Z(\tau) = \sum_{P_U, P_I} P(P_U, P_I, R^O|\hat{Q}, \gamma, \eta)^\tau$$

The sum in (B.1) on the other hand corresponds to the partition function $Z_+(\tau)$ of the new graph with the r label on the missing link.

$$Z_+(\tau) = \sum_{P_U, P_I} P(P_U, P_I, R^O_+|\hat{Q}, \gamma, \eta)^\tau$$

To deal with these exponential sums, we shall introduce some MCMC sampling methods to estimate this partition function. In the following sections, we shall focus our attention to the calculation of $Z(\tau)$. The results can be easily applied to $Z_+(\tau)$ as the latter can be viewed just as a graph with one additional rating.

B.1.1 Variational approximations

As mentioned earlier, besides the sampling techniques, we can use variational approaches to approximate the true Boltzman distribution $P(P_U, P_I|R^O, Q, \gamma, \eta)$, using a belief distribution $B(P_U, P_I)$. Since we are only interested in $Z(1)$, we shall assume $\tau = 1$, and write $Z = Z(1)$.

Recall the partition function,

$$Z = \sum_{P_U, P_I} P(P_U, P_I, R^O | \hat{Q}, \gamma, \eta) \frac{B(P_U, P_I)}{B(P_U, P_I)}$$

We take the logarithm of it, and by Jensen's inequality

$$\begin{aligned} \ln Z \geq & E_{B(P_U,P_I)} \left[\ln \frac{P(P_U, P_I, R^O | \hat{Q}, \gamma, \eta)}{B(P_U, P_I)} \right] \\ = & E_{B(P_U,P_I)} [\ln P(P_U, P_I, R^O | \hat{Q}, \gamma, \eta)] + S_{B(P_U,P_I)} = -F_{B(P_U,P_I)} \end{aligned} \tag{B.3}$$

It would become a equality if and only if $B(P_U, P_I)$ is exactly $P(P_U, P_I | R^O, Q, \gamma, \eta)$. As a result, by maximizing $-F_{B(P_U,P_I)}$ (called the negative Gibbs free energy in statistical physics) with respect to $B(P_U, P_I)$, we can approach the log partition function from below.

The key here is to use a $B(P_U, P_I)$ that is simple for inference, but yet flexible enough to fit the data closely. One class of such distributions is the cluster variational approximation.

Mean-field approximation

The cluster variational method, approximates the joint Boltzman distribution as a product of localized factors. In statistical physics, it corresponds to the Kikuchi approximations [89] where the free energy is the sum of local energy terms (see Appendix). The first order cluster approximation, also known as the mean-field approximation, has defined local belief only at the single node level, with a particularly simple form of the joint belief:

$$B_{MF}(P_U, P_I) = \prod_{u \in U} b^u_{\delta_u} \prod_{i \in I} b^i_{\delta_i}$$

For each vertex u and a type s, define b^s_u as the marginal belief that vertices $u \in U$ is of type s. They should obey the normalization conditions $\sum_s b^u_s = 1$. Similarly

we have b_i^t as the marginal belief that vertices $i \in I$ is of type t. It follows that the two-node beliefs are simply $b_{st}^{ui} = b_s^u \times b_t^i$. Now we can define the log partition function in terms of local factors:

$$\begin{aligned} -F_{MF} =& E_{B(P_U,P_I)}[\ln P(P_U, P_I, R^O|\hat{Q}, \gamma, \eta)] + S_{B(P_U,P_I)} \\ =& \sum_{u\in U}\sum_{s} b_s^u(\ln\gamma_s - \ln b_s^u) + \sum_{i\in I}\sum_{t} b_t^i(\ln\eta_t - \ln b_t^i) \\ &+ \sum_{(ui)\in R^O}\sum_{st} b_s^u b_t^i(\ln f(u,i,s,t)) \end{aligned} \tag{B.4}$$

where the function of $f(u,i,s,t)$ in our case is defined as:

$$f(u,i,s,t) = q_r(s,t) \quad if\ (u,i) \in R^r \subseteq R^O$$

Assuming the parameters and the beliefs are conditional independent, we can solve for the MLEs,

$$\hat{\gamma}_s = \frac{\sum_{u\in U} b_s^u}{n_s}, \quad \hat{\eta}_t = \frac{\sum_{i\in I} b_t^i}{n_t}, \quad \hat{q}_r(s,t) = \sum_{(ui)\in R^O} \frac{b_s^u b_t^i R_{ui}^r}{b_s^u b_t^i},$$

$$\hat{b}_s^u \propto exp[\sum_{t,i}(\ln f(u,i,s,t) b_t^i) + \ln\gamma_s - 1],$$

$$\hat{b}_t^i \propto exp[\sum_{s,u}(\ln f(u,i,s,t) b_s^u) + \ln\eta_t - 1].$$

Using numeric techniques like the EM algorithm, one can approximately solve the above problem analytically.

Bethe approximation

While the first order mean-field approximation is simple and fast [9], it does not work well for block models where correlations are the key factors. The second order Cluster variational approximation, also known as the Bethe approximation, expands the range of local factors to pairs of nodes,

It approximates the true Boltzman distribution using one-node beliefs b_s^u, as well as two-node beliefs b_{st}^{ui} [90]. For each pair of vertices u, i and pair of types s, t, define b_{st}^{ui} as the pairwise marginal belief that vertices u and i are of type s and t respectively. They should obey the marginalization conditions $\sum_t b_{st}^{ui} = b_s^u$. The Bethe estimate of the joint belief is

$$B_{Bethe}(P_U, P_I) = \frac{\prod_{u,i} b_{st}^{ui}}{\prod_u (b_s^u)^{d_u - 1} \prod_i (b_t^i)^{d_i - 1}} \tag{B.5}$$

As we can see, higher order cluster variational methods follows the inclusion –exclusion principle when summing up the local factors (see Figure 3.1 for the second order case). In general, the accuracy improves as the order increases, and it is exact when the largest local component becomes the whole graph itself.

For block models on simple graphs, the Bethe estimate of the first term in (B.3) is exact:

$$\begin{aligned} &E_{B_{Bethe}(P_U,P_I)}[\ln P(P_U, P_I, R^O | \hat{Q}, \gamma, \eta)] \\ =&\sum_{u \in U} \sum_s b_s^u \ln \gamma_s + \sum_{i \in I} \sum_t b_t^i \ln \eta_t + \sum_{(ui) \in R^O} \sum_{st} b_{st}^{ui} \ln f(u, i, s, t) \end{aligned} \tag{B.6}$$

The Bethe estimate of the entropy, on the other hand, is only exact when the graph is singly-connected.

$$\begin{aligned} &S_{Bethe}(B(P_U, P_I)) = -\sum_{P_U, P_I} B_{Bethe}(P_U, P_I) \ln B_{Bethe}(P_U, P_I) \\ =& - E_{B_{Bethe}(P_U,P_I)} \left[\sum_{(ui) \in R^O} \ln b_{st}^{ui} - \sum_u \ln (b_s^u)^{d_u - 1} - \sum_i \ln (b_t^i)^{d_i - 1} \right] \\ =& - \sum_{(ui) \in R^O} \sum_{st} b_{st}^{ui} \ln b_{st}^{ui} + \sum_{u \in U} (d_u - 1) \sum_s b_s^u \ln (b_s^u) + \sum_{i \in I} (d_i - 1) \sum_t b_t^i \ln (b_t^i) \end{aligned} \tag{B.7}$$

This is because (B.5) is not exact on graphs with loops.

Putting both together, with some rearrangement,

$$-F_{Bethe} = \sum_{(ui)\in R^O} \sum_{st} b_{st}^{ui}(\ln f(u,i,s,t) - \ln b_{st}^{ui}) + \sum_{u\in U}\sum_{s} b_s^u(\ln(b_s^u)^{d_u-1} + \ln\gamma_s) + \sum_{i\in I}\sum_{t} b_t^i(\ln(b_t^i)^{d_i-1} + \ln\eta_t) \quad \text{(B.8)}$$

If the graph is sparse in cycles, we could get a good approximation of the partition function by maximizing (B.8) instead. Assuming the parameters and the beliefs are conditional independent, we can solve for the MLEs,

$$\hat{\gamma}_s = \frac{\sum_{u\in U} b_s^u}{n_s}, \quad \hat{\eta}_t = \frac{\sum_{i\in I} b_t^i}{n_t}, \quad \hat{q}_r(s,t) = \sum_{(ui)\in R^O} \frac{b_{st}^{ui} R_{ui}^r}{b_{st}^{ui}}.$$

However, the MLEs of the Bethe beliefs are not as easy to obtain as the simple mean-field approximation. The pair-wise beliefs have introduced non-trivial dependencies, and more sophisticated optimizing tools are required. In the paper [90], the authors proved that the Belief Propagation algorithm converges to the same fixed points as the Bethe maximizing process. It makes a efficient message passing implementation possible, when analytical solution is not there.

The BP message passing algorithm

The idea of belief propagation is each vertex u sends a "message" to each of its neighbors v, consisting of the marginal distribution that u would have if v were not in the network. We denote this $\mu_t^{u\to v}$, the probability that u would be of type t if v were absent. We update $\mu_{u\to v}$ according to the messages that u receives from its *other* neighbors w.

Finally, we assume that these neighbors are independent. In other words, we ignore the effect of paths that don't go through u. This assumption holds, for instance, if the graph is locally treelike and correlations decay.

In our model, we have the following update rule for type s message from a user u to an item i,

$$\mu_s^{u\to i} = \frac{\xi_s^{u\to i}}{\sum_{t=1}^{k} \xi_s^{u\to i}},$$

The numerator $\xi_s^{u\to i}$ is the un-normalized versions of the $\mu_s^{u\to i}$, which is defined as

$$\xi_s^{u\to i} = \gamma_s \left(\prod_{\substack{j:(j,u)\in R^O \\ j\neq i}} \sum_t \mu_t^{j\to u} f(j,u,t,s) \right) \tag{B.9}$$

where the function:

$$f(j,u,t,s) = q_r(t,s) \quad if\ (j,u) \in R^r \subseteq R^O$$

Similarly, we have the type t messages from an item i to a user u,

$$\mu_t^{i\to u} \propto \eta_t \left(\prod_{\substack{v:(v,i)\in R^O \\ v\neq u}} \sum_s \mu_s^{v\to i} f(v,i,s,t) \right) \tag{B.10}$$

Once we reach a fixed point in the messages, they can be used to estimate the beliefs in the Bethe formula (B.8),

$$b_s^u \propto \gamma_s \prod_{j:(j,u)\in R^O} \sum_t \mu_t^{j\to u} f(j,u,t,s), \tag{B.11}$$

$$b_t^i \propto \eta_t \prod_{v:(v,i)\in R^O} \sum_s \mu_s^{v\to i} f(v,i,s,t), \tag{B.12}$$

$$b_{st}^{ui} \propto \mu_s^{u\to i} \mu_t^{i\to u} f(u,i,s,t) \quad v:(v,i) \in R^O\,. \tag{B.13}$$

where we normalize each of these by summing over s,t. Notice that the above messages and beliefs are only defined on observed edges, that is $(u,i) \in R^r \subseteq R^O$.

B.2 The partition function under full integration

If we take the full Bayesian approach, we have a particularly simple result with respect to the integration over Q [36]:

$$\begin{aligned} p_{SBM}(r_{ui} = r|R^O) =& \frac{1}{Z} \sum_{P_U,P_I} \frac{n^r_{\delta_u\delta_i} + 1}{n_{\delta_u\delta_i} + K} \prod_{\alpha,\beta} \frac{\prod_{k=1}^{K}(n^k_{\alpha\beta})!}{(n_{\alpha\beta} + K - 1)!} \\ =& \frac{1}{Z} \sum_{P_U,P_I} P(P_U, P_I, R^O_+) \end{aligned} \tag{B.14}$$

where $P(P_U, P_I, R^O_+)$ is the marginalized (over parameters Q) likelihood of the new graph with the r label on the missing link. Notice that the model is based on the simplified SBM as defined in (3.26), in which the first order priors γ, η are absent.

In statistical physics terms the normalization Z in (B.14) corresponds to the partition function of the observed graph. The probability density of the Boltzman distribution is given by:

$$P(P_U, P_I|R^O) = \frac{e^{\tau \ln P(P_U,P_I,R^O)}}{\sum_{P_U,P_I} e^{\tau \ln P(P_U,P_I,R^O)}} \tag{B.15}$$

Where the denominator is the partition function

$$Z = \sum_{P_U,P_I} P(P_U, P_I, R^O)$$

The sum in (B.14) on the other hand corresponds to the partition function Z_+ of the new graph with the r label on the missing link.

$$Z_+ = \sum_{P_U,P_I} P(P_U, P_I, R^O_+)$$

To deal with these exponential sums, we shall extend variational methods to the full Bayesian case. In the following sections, we shall focus our attention to the calculation of Z. The results can be easily applied to Z_+ as the latter can be viewed just as a graph with one additional rating.

B.2.1 Variational Bayesian approximation

Besides the sampling techniques, we can use variational approaches to approximate the Boltzman distribution $P(P_U, P_I|R^O)$, using a belief distribution $B(P_U, P_I)$

$$Z = \sum_{P_U, P_I} P(P_U, P_I, R^O) \frac{B(P_U, P_I)}{B(P_U, P_I)}$$

We take the logarithm of it, and by Jensen's inequality

$$\begin{aligned} \ln Z \geq & E_{B(P_U,P_I)} \left[\ln \frac{P(P_U, P_I, R^O)}{B(P_U, P_I)} \right] \\ = & E_{B(P_U,P_I)}[\ln P(P_U, P_I, R^O)] + S_{B(P_U,P_I)} = -F_{B(P_U,P_I)} \end{aligned} \tag{B.16}$$

It would become a equality if and only if $B(P_U, P_I)$ is exactly $P(P_U, P_I|R^O)$. As a result, by maximizing $-F_{B(P_U,P_I)}$ (called the negative Gibbs free energy in statistical physics) with respect to $B(P_U, P_I)$, we can approach the log partition function from below.

The key here is to use a $B(P_U, P_I)$ that is simple for inference, but yet flexible enough to fit the data closely. One class of such distributions is the cluster variational approximation which approximates the joint Boltzman distribution as a product of localized factors.

Before we can define the partition function in terms of local factors, we need to rewrite the likelihood in terms of local factors as well

$$\begin{aligned} P(P_U, P_I, R^O) = & \prod_{\alpha,\beta} \frac{\prod_{k=1}^K (n_{\alpha\beta}^k)!}{(n_{\alpha\beta} + K - 1)!} \\ \approx & \prod_{\alpha,\beta} \frac{\prod_{k=1}^K \sqrt{2\pi n_{\alpha\beta}^k}}{\sqrt{2\pi(n_{\alpha\beta} + K - 1)} (\frac{n_{\alpha\beta} + K - 1}{e})^{K-1}} \prod_{\substack{(u,i)\in R^O \\ k:(u,i)\in R^k}} \frac{n_{\delta_u\delta_i}^k}{n_{\delta_u\delta_i}} \end{aligned} \tag{B.17}$$

where we used the Stirling's approximation for factorials. This is equivalent to using Laplace's method to approximate the integral over Q, which would be quite

accurate if parameters Q follows some common regularity conditions. To see this, let us rewrite the likelihood as an integral:

$$\begin{aligned}
&P(P_U, P_I, R^O) = \int_0^1 \prod_{\alpha\in P_U}\prod_{\beta\in P_I}\prod_{i=1}^{K} q_i(\alpha,\beta)^{n^i_{\alpha,\beta}} dQ \\
=&\int_0^1 \exp\left[\frac{M}{M}\ln P(P_U, P_I, R^O|Q)\right] dQ \\
\approx& P(P_U, P_I, R^O|\hat{Q})(\frac{2\pi}{M})^{\frac{|Q|}{2}}\left|\frac{P(P_U, P_I, R^O|Q)}{\partial^2 Q}\Big|_{\hat{Q}}\right|^{-\frac{1}{2}} \\
=&\prod_{\substack{(u,i)\in R^O \\ k:(u,i)\in R^k}} \frac{n^k_{\delta_u\delta_i}}{n_{\delta_u\delta_i}} \prod_{\alpha,\beta} \frac{\sqrt{2\pi}}{\sqrt{M\times\prod_{i=1}^K n^i_{\alpha,\beta}(n^i_{\alpha,\beta}-1))}}
\end{aligned}$$

Now we take the log of (B.17), and rewrite the leading global term as $C(P_U, P_I)$,

$$\ln P(P_U, P_I, R^O) = C(P_U, P_I) + \sum_{k=1}^{K}\sum_{(u,i)\in R^k} \ln n^k_{\delta_u\delta_i} - \sum_{(u,i)\in R^O} \ln n_{\delta_u\delta_i} \tag{B.18}$$

Mean-field approximation

The first order cluster approximation, or the mean-field approximation, has defined local belief only at the single node level, with a particularly simple form of the joint belief:

$$B_{MF}(P_U, P_I) = \prod_{u\in U} b^u_{\delta_u} \prod_{i\in I} b^i_{\delta_i}$$

For each vertex u and a type s, define b^s_u as the marginal belief that vertices $u \in U$ is of type s. They should obey the normalization conditions $\sum_s b^u_s = 1$. Similarly we have b^t_i as the marginal belief that vertices $i \in I$ is of type t. It follows that the two-node beliefs are simply $b^{ui}_{st} = b^u_s \times b^i_t$.

Now we can define the log partition function in terms of node factors:

$$\begin{aligned}
-F_{MF} =& E_{B(P_U,P_I)}[\ln P(P_U, P_I, R^O)] + S_{B(P_U,P_I)} \\
=& E_{B(P_U,P_I)}[C(P_U, P_I)] + \sum_{k=1}^{K} \sum_{(u,i)\in R^k} \sum_{st} b_s^u b_t^i (\ln n_{st}^k) \\
& - \sum_{(ui)\in R^O} \sum_{st} b_s^u b_t^i (\ln n_{st}) - \sum_{u\in U} \sum_{s} b_s^u (\ln b_s^u) - \sum_{i\in I} \sum_{t} b_t^i (\ln b_t^i) \quad \text{(B.19)}
\end{aligned}$$

Unlike (B.4), the variables in (B.19) are solely determined by the beliefs $B(P_U, P_I)$, which in turn maximizes (B.19):

$$\begin{aligned}
& n_{st}^k = \sum_{(u,i)\in R^k} b_s^u b_t^i, \quad n_{st} = \sum_{k=1}^{K} n_{st}^k = \sum_{(u,i)\in R^O} b_s^u b_t^i, \\
& \hat{b}_s^u \propto exp[\sum_{k=1}^{K} \sum_{i:(u,i)\in R^k} \sum_{t} b_t^i (\ln n_{st}^k) - \sum_{i:(ui)\in R^O} \sum_{t} b_t^i (\ln n_{st}) - 1], \\
& \hat{b}_t^i \propto exp[\sum_{k=1}^{K} \sum_{u:(u,i)\in R^k} \sum_{s} b_s^u (\ln n_{st}^k) - \sum_{u:(ui)\in R^O} \sum_{s} b_s^u (\ln n_{st}) - 1].
\end{aligned}$$

Using numeric techniques like the EM algorithm, one can approximately solve the above problem analytically.

Bethe approximation

While the first order mean-field approximation is simple and fast [9], it does not work well for block models where correlations are the key factors. The second order Cluster variational approximation, also known as the Bethe approximation, expands the range of local factors to pairs of nodes,

It approximates the true Boltzman distribution using one-node beliefs b_s^u, as well as two-node beliefs b_{st}^{ui} [90]. For each pair of vertices u, i and pair of types s, t, define b_{st}^{ui} as the pairwise marginal belief that vertices u and i are of type s and t respectively.

They should obey the marginalization conditions $\sum_t b_{st}^{ui} = b_s^u$. The Bethe estimate of the joint belief is

$$B_{Bethe}(P_U, P_I) = \frac{\prod_{u,i} b_{st}^{ui}}{\prod_u (b_s^u)^{d_u - 1} \prod_i (b_t^i)^{d_i - 1}} \tag{B.20}$$

As we can see, higher order cluster variational methods follows the inclusion –exclusion principle when summing up the local factors (see Figure 3.1 for the second order case). In general, the accuracy improves as the order increases, and it is exact when the largest local component becomes the whole graph itself.

For block models on simple graphs, the Bethe estimate of the first term in (B.16) is exact:

$$\begin{aligned} & E_{B_{Bethe}(P_U,P_I)}[\ln P(P_U, P_I, R^O)] \\ =& E_{B(P_U,P_I)}[C(P_U, P_I)] + \sum_{k=1}^{K} \sum_{(u,i)\in R^k} \sum_{st} b_{st}^{ui} (\ln n_{st}^k) - \sum_{(ui)\in R^O} \sum_{st} b_{st}^{ui} (\ln n_{st}) \end{aligned} \tag{B.21}$$

The Bethe estimate of the entropy, on the other hand, is only exact when the graph is singly-connected.

$$\begin{aligned} & S_{Bethe}(B(P_U, P_I)) = -\sum_{P_U, P_I} B_{Bethe}(P_U, P_I) \ln B_{Bethe}(P_U, P_I) \\ =& - E_{B_{Bethe}(P_U,P_I)} \left[\sum_{(ui)\in R^O} \ln b_{st}^{ui} - \sum_u \ln (b_s^u)^{d_u - 1} - \sum_i \ln (b_t^i)^{d_i - 1} \right] \\ =& - \sum_{(ui)\in R^O} \sum_{st} b_{st}^{ui} \ln b_{st}^{ui} + \sum_{u\in U} (d_u - 1) \sum_s b_s^u \ln (b_s^u) + \sum_{i\in I} (d_i - 1) \sum_t b_t^i \ln (b_t^i) \end{aligned} \tag{B.22}$$

This is because (B.20) is not exact on graphs with loops.

Putting both together, with some rearrangement,

$$\begin{aligned}-F_{Bethe} =& E_{B(P_U,P_I)}[C(P_U,P_I)] \\ &+ \sum_{k=1}^{K} \sum_{(u,i)\in R^k} \sum_{st} b_{st}^{ui}(\ln n_{st}^k) - \sum_{(ui)\in R^O} \sum_{st} b_{st}^{ui}(\ln n_{st} + \ln b_{st}^{ui}) \\ &+ \sum_{u\in U}(d_u - 1) \sum_{s} b_s^u \ln(b_s^u) + \sum_{i\in I}(d_i - 1) \sum_{t} b_t^i \ln(b_t^i) \end{aligned} \tag{B.23}$$

If the graph is sparse in cycles, we could get a good approximation of the partition function by maximizing (B.23) with respect to $B(P_U, P_I)$. Given the the beliefs $B(P_U, P_I)$, we would get an analytical solution to (B.23), by plugging in following values

$$n_{st}^k = \sum_{(u,i)\in R^k} b_{st}^{ui}, \quad n_{st} = \sum_{k=1}^{K} n_{st}^k = \sum_{(u,i)\in R^O} b_{st}^{ui}.$$

Here we shall employ the EM framework again. We will solve for the MLEs of $B(P_U, P_I)$ while fixing the above variables (E-step), and iteratively update both back and forth. However, the MLEs of the Bethe beliefs are not as easy to obtain even with everything else fixed. The pair-wise beliefs have introduced non-trivial dependencies, and more sophisticated optimizing tools are required. In the paper [90], the authors proved that the Belief Propagation algorithm converges to the same fixed points as the Bethe maximizing process. It makes a efficient message passing implementation possible, when analytical solution is not there.

The BP message passing algorithm

The idea of belief propagation is each vertex u sends a "message" to each of its neighbors v, consisting of the marginal distribution that u would have if v were not in the network. We denote this $\mu_t^{u\to v}$, the probability that u would be of type t if v were absent. We update $\mu_{u\to v}$ according to the messages that u receives from its *other* neighbors w.

Finally, we assume that these neighbors are independent. In other words, we ignore the effect of paths that don't go through u. This assumption holds, for instance, if the graph is locally treelike and correlations decay.

In our model, we have the following update rule for type s message from a user u to an item i,

$$\mu_s^{u\to i} = \frac{\xi_s^{u\to i}}{\sum_{t=1}^{k} \xi_s^{u\to i}},$$

The numerator $\xi_s^{u\to i}$ is the un-normalized versions of the $\mu_s^{u\to i}$, which is defined as:

$$\xi_s^{u\to i} = \prod_{\substack{j:(j,u)\in R^O \\ j\neq i}} \sum_t \mu_t^{j\to u} \frac{\prod_{k=1}^{K} n_{st}^k}{n_{st}} \tag{B.24}$$

Similarly, we have the type t messages from an item i to a user u,

$$\xi_t^{i\to u} = \prod_{\substack{v:(v,i)\in R^O \\ v\neq u}} \sum_s \mu_t^{v\to i} \frac{\prod_{k=1}^{K} n_{st}^k}{n_{st}} \tag{B.25}$$

Once we reach a fixed point in the messages, they can be used to estimate the beliefs in the Bethe formula (B.8),

$$b_s^u \propto \prod_{j:(u,j)\in R^O} \sum_t \mu_t^{j\to u} \frac{\prod_{k=1}^{K} n_{st}^k}{n_{st}}, \tag{B.26}$$

$$b_t^i \propto \prod_{v:(v,i)\in R^O} \sum_s \mu_t^{v\to i} \frac{\prod_{k=1}^{K} n_{st}^k}{n_{st}}, \tag{B.27}$$

$$b_{st}^{ui} \propto \mu_s^{u\to i} \mu_t^{i\to u} \frac{\prod_{k=1}^{K} n_{st}^k}{n_{st}} \quad v:(v,i)\in R^O\,. \tag{B.28}$$

where we normalize each of these by summing over s, t. Notice that the above messages and beliefs are only defined on observed edges, that is $(u,i)\in R^r \subseteq R^O$.

Appendix C

Theoratic Derivation of Likelihood ratios

C.1 LRT for SBM vs DC-SBM

For simplicity we focus on one group with expected degree μ. Assuming independence between the groups will then recover the expressions (4.4) and (4.6) where the mean and variance of Λ is a weighted sum over groups. We have

$$\begin{aligned}\Lambda &= \sum_{i=1}^{n} d_i \log \frac{d_i}{\overline{d}} \\ &= \sum_i d_i \log d_i - \left(\sum_i d_i\right) \log \left(\sum_i d_i\right) + \left(\sum_i d_i\right) \log n \,, \end{aligned} \tag{C.1}$$

where $\overline{d} = (1/n) \sum_i d_i$ is the sample mean. We wish to compute the mean and expectation of $\log L$ if the data is generated by the null model.

If d is Poisson-distributed with mean μ, let $f(\mu)$ denote the difference between

the expectation of $d \log d$ and its most likely value $\mu \log \mu$:

$$f(\mu) = \left(\sum_{d=1}^{\infty} \frac{\mathrm{e}^{-\mu} \mu^d}{d!} \, d \log d \right) - \mu \log \mu \,. \tag{C.2}$$

Assume that the d_i are independent and Poisson with mean μ; this is reasonable in a large sparse graph, since the correlations between degrees of different nodes is $O(1/n)$. Then $\sum_i d_i$ is Poisson with mean $n\mu$, and (C.1) gives

$$\mathbb{E}[\Lambda] = n f(\mu) - f(n\mu) \,. \tag{C.3}$$

To understand this asymptotically, note that $f(\mu)$ converges to $1/2$ when μ is large. Thus in the limit of large n,

$$\mathbb{E}[\Lambda] = n f(\mu) - \frac{1}{2} \,.$$

When μ is large, this gives $\mathbb{E}[\Lambda] = (n-1)/2$, just as χ^2 hypothesis testing would suggest. However, as Fig. C.1 shows, $f(\mu)$ deviates significantly from $1/2$ for finite μ. We can obtain the leading corrections as a power series in $1/\mu$ by approximating (C.2) with the Taylor series of $d \log d$ around $d = \mu$, giving

$$f(\mu) = \frac{1}{2} + \frac{1}{12\mu} + \frac{1}{12\mu^2} + O(1/\mu^3) \,.$$

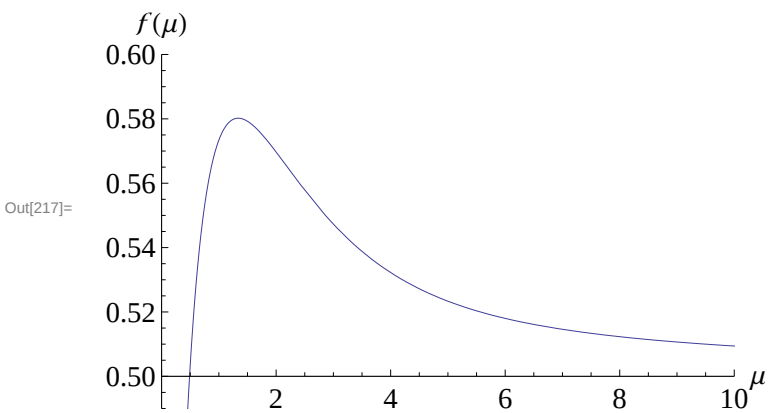


Figure C.1: The function $f(\mu)$ defined in (C.2), or equivalently the expected log-likelihood difference divided by n. We compare this with experiment in Fig. 4.3(a).

Computing the variance is harder, but still possible. It will be convenient to define several functions. If d is Poisson with mean μ, let $\phi(\mu)$ denote the variance of $d \log d$:

$$\begin{aligned}\phi(\mu) = \mathrm{Var}[d \log d] &= \mathbb{E}\left[(d \log d)^2\right] - \mathbb{E}[d \log d]^2 \\ &= \sum_{d=0}^{\infty} \frac{\mathrm{e}^{-\mu}\mu^d}{d!}(d \log d)^2 - \left(f(\mu) + \mu \log \mu\right)^2 \ . \end{aligned} \tag{C.4}$$

We will also use

$$\begin{aligned}c(\mu) = \mathrm{Cov}[d, d \log d] &= \mathbb{E}\left[d^2 \log d\right] - \mu \mathbb{E}[d \log d] \\ &= \sum_{d=1}^{\infty} \frac{\mathrm{e}^{-\mu}\mu^d}{d!}\, d^2 \log d - \mu \left(f(\mu) + \mu \log \mu\right) \ . \end{aligned} \tag{C.5}$$

Finally, let $\lambda \geq \mu$, and let d and u be independent and Poisson with mean μ and $\lambda - \mu$ respectively. Then let

$$\begin{aligned}r(\mu, \lambda) &= \mathrm{Cov}[d \log d, (d+u) \log(d+u)] \\ &= \mathbb{E}[(d \log d)((d+u) \log(d+u))] - \mathbb{E}[d \log d]\, \mathbb{E}[(d+u) \log(d+u)] \\ &= \sum_{d,u=1}^{\infty} \frac{\mathrm{e}^{-\lambda}\mu^d(\lambda-\mu)^u}{d!u!}(d \log d)((d+u) \log(d+u)) \\ &\quad - \left(f(\mu) + \mu \log \mu\right)\left(f(\lambda) + \lambda \log \lambda\right) \ , \end{aligned} \tag{C.6}$$

where we used the fact that $d + u$ is Poisson with mean λ.

Then again assuming that the d_i are independent, we have the following terms

and cross-terms for the variance of (C.1):

$$\begin{aligned}
\operatorname{Var}\left[\sum_i d_i \log d_i\right] &= n\phi(\mu) \\
\operatorname{Var}\left[\left(\sum_i d_i\right) \log\left(\sum_i d_i\right)\right] &= \phi(n\mu) \\
\operatorname{Var}\left[\sum_i d_i\right] &= n\mu \\
\operatorname{Cov}\left[\sum_i d_i \log d_i,\ \left(\sum_i d_i\right) \log\left(\sum_i d_i\right)\right] &= nr(\mu, n\mu) \\
\operatorname{Cov}\left[\sum_i d_i \log d_i,\ \sum_i d_i\right] &= nc(\mu) \\
\operatorname{Cov}\left[\left(\sum_i d_i\right) \log\left(\sum_i d_i\right),\ \sum_i d_i\right] &= c(n\mu)
\end{aligned}$$

Putting this all together, we have

$$\operatorname{Var}[\Lambda] = n\phi(\mu) + \phi(n\mu) + n\mu \log^2 n - 2nr(\mu, n\mu) + 2\big(nc(\mu) - c(n\mu)\big) \log n\,. \quad \text{(C.7)}$$

In the limit of large μ, using Taylor series to expand the summands of (C.4) and (C.5) gives the following simplifications:

$$\begin{aligned}
\phi(\mu) &= \mu \log^2 \mu + 2\mu \log \mu + \mu + \frac{1}{2} + O\left(\frac{\log \mu}{\mu}\right) \\
c(\mu) &= \mu \log \mu + \mu + O(1/\mu)\,.
\end{aligned}$$

Also, when $\lambda \gg \mu$ and $\mu = O(1)$, using $\log(d+u) \approx \log u + d/u$ lets us separate the double sum in (C.6), giving

$$\begin{aligned}
r(\mu, \lambda) = \mathbb{E}\big[d^2 \log d\big]\,(1 + \log \lambda) + \mathbb{E}[d \log d]\,\mathbb{E}[u \log u] \\
- \mathbb{E}[d \log d]\,\mathbb{E}[(d+u)\log(d+u)]) + O(1/\lambda)\,.
\end{aligned}$$

In particular, setting $\lambda = n\mu$ gives

$$r(\mu, n\mu) = c(\mu)(1 + \log n\mu) + O(1/n)\,.$$

Finally, keeping $O(n)$ terms in (C.7) and defining $v(\mu)$ as in (4.6) gives

$$v(\mu) = \lim_{n\to\infty} \frac{1}{n}\mathrm{Var}[\Lambda] = \phi(\mu) + \mu(1+\log\mu)^2 - 2c(\mu)(1+\log\mu)\,. \tag{C.8}$$

Using the definitions of ϕ and c, we can write this more explicitly as (where Var and Cov denote the variance and covariance in the Poisson distribution with mean μ)

$$\begin{aligned} v(\mu) &= \mu(1+\log\mu)^2 + \mathrm{Var}[d\log d] - 2(1+\log\mu)\,\mathrm{Cov}[d, d\log d] \\ &= \mu(1+\log\mu)^2 \\ &\quad + \sum_{d=1}^{\infty} \frac{\mathrm{e}^{-\mu}\mu^d}{d!}(d\log d)\big(d\log d - 2(1+\log\mu)(d-\mu)\big) \\ &\quad - \left(\sum_{d=1}^{\infty} \frac{\mathrm{e}^{-\mu}\mu^d}{d!}\, d\log d\right)^2 . \end{aligned} \tag{C.9}$$

We plot this function in Fig. C.2. It converges to 1/2 in the limit of large μ, but it is significantly larger for finite μ.

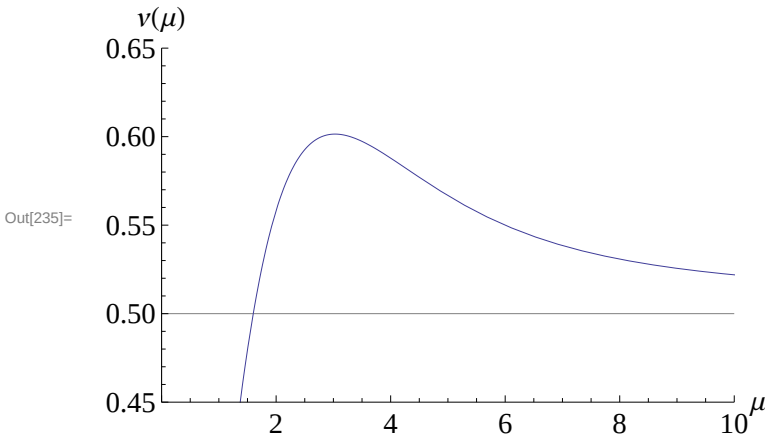


Figure C.2: The asymptotic variance of the log-likelihood difference, divided by n, given in (C.8). We compare this with experiment in Fig. 4.3(b).

References

[1] L. Adamic and N. Glance. The political blogosphere and the 2004 US Election: Divided They Blog. In *Proc 3rd Intl Workshop on Link Discovery.*, 2005.

[2] C. Aicher, A. Z. Jacobs, and A. Clauset. Adapting the Stochastic Block Model to Edge-Weighted Networks. *ArXiv e-prints*, May 2013.

[3] E. Airoldi, S. Fienberg, C. Joutard, and T. Love. Discovery of latent patterns with hierarchical bayesian mixed-membership models and the issue of model choice. *Data mining patterns: new methods and applications*, page 240, 2008.

[4] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *J. Machine Learning Research*, 9:1981–2014, 2008.

[5] H. Akaike. *Information theory and an extension of the maximum likelihood principle*, volume 1, pages 267–281. Akademiai Kiado, 1973.

[6] D. Alderson, L. Li, W. Willinger, and J. C. Doyle. Understanding internet topology: principles, models, and validation. *IEEE/ACM Trans. Networks*, 13(6):1205–1218, 2005.

[7] S. Allesina and M. Pascual. Food web models: a plea for groups. *Ecology letters*, 12(7):652–662, 2009.

[8] D. R. Anderson and K. P. Burnham. Avoiding Pitfalls When Using Information-Theoretic Methods. *The Journal of Wildlife Management*, 66(3):pp. 912–918, 2002.

[9] M. Beal and Z. Ghahramani. Variational Bayesian learning of directed graphical models with hidden variables. *Bayesian Analysis*, 1(4):793–832, 2006.

[10] P. J. Bickel and A. Chen. A nonparametric view of network models and Newman-Girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106:21068–21073, 2009.

[11] M. Bilgic and L. Getoor. Link-based Active Learning. In *NIPS Workshop on Analyzing Networks and Learning with Graphs*, 2009.

[12] M. Bilgic, L. Mihalkova, and L. Getoor. Active Learning for Networked Data. In *Proc. Intl. Conf. on Machine Learning*, 2010.

[13] P. Billingsley. Statistical Methods in Markov Chains. *The Annals of Mathematical Statistics*, 32:12–40, Mar. 1961.

[14] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[15] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[16] U. Brose, L. Cushing, E. L. Berlow, T. Jonsson, C. Banasek-Richter, L. F. Bersier, J. L. Blanchard, T. Brey, S. R. Carpenter, M. F. Blandenier, et al. Body sizes of consumers and their resources. *Ecology*, 86(9):2545–2545, 2005.

[17] P. Burman, E. Chow, and D. Nolan. A cross-validatory method for dependent data. *Biometrika*, 81(2):351–358, 1994.

[18] K. Burnham. *Model selection and multi-model inference : a practical information-theoretic approach*. Springer, New York NY, 2. ed., [Repr.]. edition, 2010.

[19] H. Chipman, E. I. George, R. E. McCulloch, M. Clyde, D. P. Foster, and R. A. Stine. The practical implementation of Bayesian model selection. *Lecture Notes-Monograph Series*, 38:65–134, 2001.

[20] G. Claeskens and N. L. Hjort. *Model Selection and Model Averaging*. Cambridge University Press, Cambridge, England, 2008.

[21] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.

[22] A. C. Davison and D. V. Hinkley. *Bootstrap Methods and their Applications*. Cambridge University Press, Cambridge, England, 1997.

[23] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Asymptotic analysis of the stochastic block model for networks and its algorithmic applications. *Physical Review E*, 84, 2011.

[24] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Inference and Phase Transitions in the Detection of Modules in Sparse Networks. *Phys. Rev. Lett.*, 107:065701, Aug 2011.

[25] U. B. et al. Consumer-resource body-size relationships in natural food webs. *Ecology*, 87(10):2411–2417, 2006.

[26] S. Fortmann-Roe. Understanding the Bias-Variance Tradeoff.

[27] S. Fortunato. Community detection in graphs. *Physics Reports*, 2009.

[28] N. Friel and A. Pettitt. Marginal likelihood estimation via power posteriors. *JOURNAL-ROYAL STATISTICAL SOCIETY. SERIES B STATISTICAL METHODOLOGY*, 70(3):589, 2008.

[29] L. Gao and J. Rexford. Stable internet routing without global coordination. *IEEE/ACM Trans. Networks*, 9(6):681–692, 2001.

[30] L. Getoor and B. Taskar. *Introduction to statistical relational learning*. The MIT Press, 2007.

[31] C. J. Geyer. Le Cam Made Simple: Asymptotics of Maximum Likelihood without the LLN or CLT or Sample Size Going to Infinity. Technical Report 643, School of Statistics, University of Minnesota, 2005.

[32] A. B. Goldberg, X. Zhu, and S. Wright. Dissimilarity in graph-based semi-supervised classification. *J. Machine Learning Research W&P*, 2:155–162, 2007.

[33] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A Survey of Statistical Network Models. *Foundations and Trends in Machine Learning*, 2(2):129–233, 2009.

[34] T. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. 2005.

[35] P. Grünwald. *The minimum description length principle*. MIT press, 2007.

[36] R. Guimerà, A. Llorente, E. Moro, and M. Sales-Pardo. Predicting Human Preferences Using the Block Structure of Complex Social Networks. *PloS one*, 7(9):e44620, 2012.

[37] R. Guimera and M. Sales-Pardo. Missing and spurious interactions and the reconstruction of complex networks. *PNAS*, 106:22073–22078, 2009.

[38] Y. Guo and R. Greiner. Optimistic active learning using mutual information. In *Proc. Intl. Joint Conf. on Artificial Intelligence*, 2007.

[39] M. Handcock, A. Raftery, and J. Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2):301–354, 2007.

[40] M. S. Handcock, A. E. Raftery, and J. M. Tantrum. Model-based clustering for social networks. *J. Royal Statist. Soc. A*, 170(2):1–22, 2007.

[41] M. B. Hastings. Community Detection as an Inference Problem. *Physical Review E*, 74(3):035102, 2006.

[42] D. Heckerman. A tutorial on learning with Bayesian networks. *Innovations in Bayesian Networks*, pages 33–82, 2008.

[43] J. M. Hofman and C. H. Wiggins. Bayesian approach to network modularity. *Physical Review Letters*, 100(25):258701, 2008.

[44] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: first steps. *Social networks*, 5:109–137, 1983.

[45] M. Houseman and D. R. White. *Taking Sides: Marriage Networks and Dravidian Kinship in Lowland South America*, pages 214–243. Transformations of Kinship. Smithsonian Institution Press, 1998.

[46] U. Jacob. *Trophic Dynamics of Antarctic Shelf Ecosystems—Food Webs and Energy Flow Budgets*. PhD thesis, University of Bremen, 2005.

[47] A. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14:841, 2002.

[48] B. Karrer and M. E. J. Newman. *Stochastic blockmodels and community structure in networks*, Jan. 2011.

[49] F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519, 2001.

[50] J. A. Lasserre, C. M. Bishop, and T. P. Minka. Principled hybrids of generative and discriminative models. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 87–94, 2006.

[51] P. Li. *Hypothesis testing in finite mixture models*. PhD thesis, University of Waterloo, 2007.

[52] S. Lin, B. Sturmfels, and Z. Xu. Marginal likelihood integrals for mixtures of independence models. *The Journal of Machine Learning Research*, 10:1611–1631, 2009.

[53] D. V. Lindley. On a measure of the information provided by an experiment. *Ann. Math. Statist.*, 27(4):986–1005, 1956.

[54] J. Machta and R. S. Ellis. Monte Carlo Methods for Rough Free Energy Landscapes: Population Annealing and Parallel Tempering. *Journal of Statistical Physics*, pages 1–13, 2011.

[55] D. J. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[56] C. Moore, X. Yan, Y. Zhu, J. Rouquier, and T. Lane. Active learning for node classification in assortative and disassortative networks. In *KDD '11 Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 841. ACM Press, 2011.

[57] M. Mørup and L. K. Hansen. Learning latent structure in complex networks. In *NIPS Workshop on Analyzing Networks and Learning with Graphs*, 2009.

[58] E. Mossel, J. Neeman, and A. Sly. Stochastic Block Models and Reconstruction. *ArXiv e-prints*, Feb. 2012.

[59] R. M. Neal and G. E. Hinton. A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368, Dordrecht, 1998. Kluwer Academic.

[60] M. Newman. Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Physical Review E*, 64(1), 2001.

[61] M. E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[62] M. E. J. Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 27(1):39–54, 2005.

[63] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):36104, 2006.

[64] M. E. J. Newman and E. A. Leicht. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, 104:9564–9569, 2006.

[65] A. S. Patterson, Y. Park, and J. S. Bader. Degree-corrected block models. Manuscript.

[66] T. Peixoto. Entropy of stochastic blockmodel ensembles. *PRE*, 85(5):056122, May 2012.

[67] T. Peixoto. Parsimonious module inference in large networks. *arXiv preprint arXiv:1212.4794*, 2012.

[68] T. Pierce. Inference of large-scale structure in networks. Master's thesis, University of New Mexico, 2008.

[69] J. Pitman. *Combinatorial stochastic processes*, volume 1875. Springer-Verlag, 2006.

[70] M. A. Porter, J. P. Onnela, and P. J. Mucha. Communities in networks. *Notices of the American Mathematical Society*, 56(9):1082–1097, 2009.

[71] M. Qi and G. Zhang. An investigation of model selection criteria for neural network time series forecasting. *European Journal of Operational Research*, 132(3):666–680, 2001.

[72] J. Racine. Consistent cross-validatory model-selection for dependent data:¡ i¿ hv-block cross-validation. *Journal of econometrics*, 99(1):39–61, 2000.

[73] J. Reichardt, R. Alamino, and D. Saad. The Interplay between Microscopic and Mesoscopic Structures in Complex Networks. *PLoS ONE*, 6:e21282, 2011.

[74] M. Rosvall and C. T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences*, 104(18):7327, 2007.

[75] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[76] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proc. 18th Intl. Conf. on Machine Learning*, pages 441–448, 2001.

[77] M. J. Schervish. *Theory of Statistics*. Springer Series in Statistics. Springer-Verlag, Berlin, 1995.

[78] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[79] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective Classification in Network Data. *AI Magazine*, 29(3):93–106, 2008.

[80] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 111–147, 1974.

[81] M. R. E. Symonds and A. Moussalli. A brief guide to model selection, multimodel inference and model averaging in behavioural ecology using Akaike's information criterion. *Behavioral Ecology and Sociobiology*, 65:13–21, Aug. 2010.

[82] W. Tong and R. Jin. Semi-supervised learning by mixed label propagation. In *Proc. 22nd Intl. Conf. on Artificial intelligence*, volume 1, pages 651–656, 2007.

[83] J. Čopič, M. O. Jackson, and A. Kirman. Identifying Community Structures from Network Data. *B.E. Press Journal of Theoretical Economics*, 9(1):Article 30, 2009.

[84] U. von Luxburg. Clustering stability: an overview. *Arxiv preprint arXiv:1007.1075*, 2010.

[85] Y. J. Wang and G. Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19, 1987.

[86] S. Wasserman and C. Anderson. Stochastic a posteriori blockmodels: Construction and assessment. *Social Networks*, 9:1–36, 1987.

[87] R. J. Williams, A. Anandanadesan, and D. Purves. The Probabilistic Niche Model Reveals the Niche Structure and Role of Body Size in a Complex Food Web. *PLoS One*, 5(8):e1209, 2010.

[88] R. J. Williams and N. D. Martinez. Simple rules yield complex food webs. *Nature*, 404(6774):180–183, 2000.

[89] J. Yedidia, W. Freeman, and Y. Weiss. Constructing Free-Energy Approximations and Generalized Belief Propagation Algorithms. *IEEE Transactions on Information Theory*, 51:2282–2312, July 2005.

[90] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. pages 239–269, 2003.

[91] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.

[92] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining Active Learning and Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In *Proc. ICML-2003 Workshop on the Continuum from Labeled to Unlabeled Data*, 2003.

[93] Y. Zhu, X. Yan, L. Getoor, and C. Moore. Scalable Text and Link Analysis with Mixed-Topic Link Models. *ArXiv e-prints*, Mar. 2013.

[94] Y. Zhu, X. Yan, and C. Moore. Generating and Inferring Communities with Inhomogeneous Degree Distributions. *arXiv:1205.7009v1*, 2012.