VARIATIONAL APPROXIMATION FOR IMPORTANCE SAMPLING AND STATISTICAL INFERENCE ON SOCIAL INFLUENCE

BY

XIAO SU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Statistics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Doctoral Committee:

Professor Yuguo Chen, Chair and Director of Research
Professor Feng Liang
Professor Naveen Narisetty
Professor Douglas Simpson

Abstract

Monte Carlo methods are widely used in statistical computing area to solve different problems. Social network analysis plays an importance role in many fields. In this dissertation, we focus on improving the efficiency of importance sampling, detecting the degrees of influence in networks, and exploring properties of generalized Erdős-Rényi model.

In the first part of the thesis, we propose an importance sampling algorithm with proposal distribution obtained from variational approximation. This method combines the strength of both importance sampling and the variational method. On one hand, this method avoids the bias from variational approximation. On the other hand, variational approximation provides a way to design the proposal distribution for the importance sampling algorithm. Theoretical justification of the proposed method is provided. Numerical results show that using variational approximation as the proposal can improve the performance of importance sampling and sequential importance sampling.

In the second part of the thesis, we propose a sequential hypothesis testing procedure to detect the degrees of influence in a network. We build a multivariate Bernoulli model to represent the status of each node in the network with different degrees of influence. A double bootstrap strategy is used to resolve the uncertainty from by estimating nuisance parameters in hypothesis testing. Theoretical justification of the proposed method is provided to show that the hypothesis testing is powerful for larger networks. Simulation studies show that our method can preserve the levels and improve the powers in hypothesis testing. We also apply our proposed method on two real network data to explore the degree of influence for various features.

In the third part of the thesis, we propose a random graph model for undirected networks with small-world properties, namely with a high clustering coefficient and a low average path length. We generalize the regular Erdős-Rényi dyadic random graph by considering higher-order motif, which is triadic graph. We show some properties of our proposed model, analyze the probability of multi-edges, and compare the local clustering coefficient with ER model. In addition, we also provide some conditions about phase transition including connectivity threshold and the existence of giant components.

To my family.

Acknowledgments

First, I would like to express my sincerest appreciation to my advisor, Professor Yuguo Chen, for his guidance, enthusiasm, encouragement and patience. He gives me a lot of great ideas for my research, and encourages me when I feel frustrated. My research work could not be done without his help and support.

Second, I would like to give special thanks to my thesis committee members Professor Feng Liang, Professor Naveen Narisetty and Professor Douglas Simpson for their valuable suggestions to my thesis research. I would also like to thank Professor Partha Dey from the Department of Mathematics at University of Illinois Urbana-Champaign for his generous help to my research.

In addition, I would share my thanks to the faculty and staff members in Department of Statistics at University of Illinois Urbana-Champaign for providing a friendly environment for study and research. I am also very grateful to all fellow students in the Statistics Department who have shared their experiences in life and study.

Last but not the least, I would like to thank my parents for their unconditional love, continuous encouragement and support to my study and life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Bayesian inference is a popular statistical method for the posterior distribution of parameters or latent variables. There are many applications of it in a wide range of disciplines, such as engineering, biology, pharmacy, sociology etc. In decision theory, we are more concerned with the posterior mean or median, which can provide a good estimation in the sense of minimizing the risk with respect to some particular loss functions. Bayesian framework also indicates the underlying structures of some complex models, which means it plays an important role when we are making decisions. In addition, Bayesian inference is sometimes used in graphical models, social network analysis and variable selection procedures.

If the analytical computation on the posterior distribution is difficult to implement, we can consider using the Monte Carlo methods as an alternative way to generate samples from the posterior distribution. However, the posterior distributions are hard to sample for some complex statistical models with both unknown parameters and latent variables. Markov chain Monte Carlo (MCMC) (Hastings, 1970; Geman and Geman, 1984) and importance sampling (IS) are widely used for handling such problems. Importance sampling draws samples from an easy-to-sample proposal distribution, and then correct the bias by introducing the importance weights. Choosing a good proposal distribution is essential to improve the efficiency of importance sampling algorithm. When the target distribution is hard to sample directly, we often try to find a proposal distribution which is close to the target distribution to reduce the variance of the importance weight. For high dimensional problems, sequential importance sampling (SIS) (Liu and Chen, 1998; Doucet et al., 2000) gives a way to construct the proposal sequentially.

Variational Bayes (VB) (Jordan et al., 1999) tackles the problem in a different way by deriving a tractable approximation to the posterior distribution. It minimizes the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) between the posterior and the variational approximation, and uses the variational approximation to make inference. In the optimization part, VB algorithm usually uses stochastic optimization (Robbins and Monro, 1951) or coordinate optimization strategy. This method is also related to EM algorithm (Dempster et al., 1977; Neal and Hinton, 1998), and we can also treat the optimization procedure as two steps, which is just like how EM works. There are a lot of fields that variational method can be applied, such as computational biology (Sanguinetti et al., 2006), network data analysis (Hofman and Wiggins, 2008), natural language processing (Blei et al., 2003), building statistical model (Armagan and Dunson, 2011) etc.

An advantage of VB is the variational approximation can be obtained quickly, and it usually takes less time than Monte Carlo sampling algorithms, such as MCMC. However, one issue for the variation method is that the gap between the variation approximation and the true posterior distribution may lead to biased inference based on the variational approximation. In many problems, the estimate obtained from the variation approximation may not be consistent. Also the uncertainty of the VB estimate is not available. In order to resolve the bias associated with VB, we will use the variational approximation as the proposal distribution for importance sampling, and then use the importance weight to correct the bias. Since the importance sampling estimate is consistent under mild conditions, the bias issue of VB is resolved. In the meantime, since the variational approximation is close to the true posterior distribution and is usually easy to sample, it is an ideal choice for the importance sampling proposal distribution. So, this idea combines the strength of these two methods. We will provide theoretical justification of the proposed methods using the f-divergence (Ali and Silvey, 1966). We will also implement the proposed methods on several models to demonstrate its performance in practice.

In chapter 2, we consider using the variational approximation as the proposal distribution

for importance sampling, and then using the importance weight to correct the bias. Since the importance sampling estimate is consistent under mild conditions, the bias issue of VB is resolved. The uncertainty of the importance sampling estimate is also relatively easy to obtain. In the meantime, since the variational approximation is close to the true posterior distribution and is usually easy to sample, it is a good choice for the importance sampling proposal distribution. So this idea combines the strength of these two methods. We will provide theoretical justification of the proposed method using the f-divergence (Ali and Silvey, 1966), and implement the proposed method on univariate normal model, Gaussian mixture model, Bayesian linear regression model, hidden Markov model and Dirichlet process mixture model to demonstrate its performance in practice.

Social network analysis plays an importance role in many fields, including sociology, psychology, biology etc. Many new methods have been developed in recent years to analyze the network data, and statistical technique sometimes gets involved when implementing the analysis procedure. Usually, we are interested in figuring out if the individuals traits can spread from one person to another through a process, which is usually known as social influence or peer effect. Another purpose to analyze the network data is to explore the internal structure and do the community detection to find clusters or groups of people who are friends with a higher probability.

Mathematically, we use a graph to denote the whole network, and each person will be presented by a vertex or node in the graph. Also, the friendship can be depicted by the edges between each pair of the nodes. This abstract notation provides us a very intuitive way to describe the network, and is also convenient to build the statistical models and do the estimation.

However, most people are concerned about the exploration of community structure in the network analysis, and they ignore the covariates of each person and build the model without considering the peer effects. For example, medical research is centered on individual health outcomes, or we say people smoke or not. The researchers should also care about if

the behavior of one person has any influences to another one, or how long can the effects being lasted. Here, we are interested in finding if the individuals traits can spread from one person to another, which is usually known as social influence or social diffusion. We use the term degrees of influence to denote how long the social influence can last in a network. If the degrees of influence is zero, then the behaviors of all people in the network are independent, and can not be influenced by other people. On the other hand, if the degrees of influence is a positive integer, then people's behaviors or opinions can be influenced by their friends or even friends' friends.

In chapter 3, we focus on exploring the degrees of influence in an observed network. We build a multivariate Bernoulli model to specify the correlation structure of the people's behaviors in the whole network. In order to detect the true degrees of influence, we propose a sequential hypothesis testing procedure and overcome the issue of nuisance parameters by introducing double bootstrap (Beran, 1988). In addition, we show that under certain conditions, the power of our proposed hypothesis testing goes to one when the network is large. We also do some simulation studies and real data analyses to illustrate the performance of our proposed method.

When describing a network, people usually assume each possible edge only contains two vertices. In graph theory, the Erdős-Rényi (ER) model is one of the most popular method for generating random graphs. There are two related generating procedures for the ER model. The first one is the $G(n, M)$ model, we randomly select one from the collection of all graphs which contains n vertices and M edges. The other one is the $G(n, p)$ model, we randomly connect nodes in a graph with n vertices, and each edge appears in the graph with the same probability p. However, in some complex real-world examples, the original graph setting for the ER model is not enough to describe the model. So, a new type of graph with hyper-edges is introduced to represent the structure of those networks, and some people also call it hyper-graph.

In Chapter 4, we propose a random graph model for undirected networks with small-

world properties. We generalize the regular Erdős-Rényi dyadic random graph by considering higher-order motif, which is triadic graph. We show some properties of our proposed model, including analyzing the probability of multi-edges and comparing the local clustering coefficient with ER model. In addition, we also provide some conditions about phase transition including connectivity threshold and the existence of giant components.

Chapter 2

Variational Approximation for Importance Sampling

2.1 Introduction

Monte Carlo methods, such as importance sampling (IS) and Markov chain Monte Carlo (MCMC), are widely used in Bayesian inference when analytical computation based on the posterior distribution is difficult. The posterior distributions are sometimes hard to sample directly, especially for complex statistical models with both unknown parameters and latent variables. In that case, importance sampling draws samples from an easy-to-sample proposal distribution, and then corrects the bias by the importance weights. Choosing a good proposal distribution is essential to the efficiency of importance sampling algorithms. We often try to find a proposal distribution that is close to the target distribution to reduce the variance of the importance weight. For high dimensional problems, sequential importance sampling (SIS) (Liu and Chen, 1998; Doucet et al., 2000) gives a way to construct the proposal distribution sequentially.

Variational Bayes (VB) (Jordan et al., 1999) tackles the problem in a different way by deriving a tractable approximation to the posterior distribution. It minimizes the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) between the posterior and the variational approximation, and uses the variational approximation to make inference. In the optimization part, VB algorithm usually uses stochastic optimization (Robbins and Monro, 1951) or coordinate optimization strategy. This method is also related to the EM algorithm (Dempster et al., 1977; Neal and Hinton, 1998). VB, IS and MCMC can be used for general computational problems, but in this chapter we focus on the application in Bayesian settings

to make the discussion more concrete.

An advantage of VB is the variational approximation can be obtained quickly, and it usually runs faster than Monte Carlo sampling algorithms such as MCMC. The variational method has been applied in many fields, such as computational biology (Sanguinetti et al., 2006), network data analysis (Hofman and Wiggins, 2008), natural language processing (Blei et al., 2003), and statistical inference (Armagan and Dunson, 2011). However, one issue with the variation method is that the gap between the variational approximation and the true posterior distribution may lead to biased inference based on variational approximation. In many problems, the estimate based on variation approximation may not be consistent. Also the uncertainty of the VB estimate is not available.

In this chapter, we consider using the variational approximation as the proposal distribution for importance sampling, and then using the importance weight to correct the bias. Since the importance sampling estimate is consistent under mild conditions, the bias issue of VB is resolved. The uncertainty of the importance sampling estimate is also relatively easy to obtain. In the meantime, since the variational approximation is close to the true posterior distribution and is usually easy to sample, it is a good choice for the importance sampling proposal distribution. So this idea combines the strength of these two methods. We will provide theoretical justification of the proposed method using the f-divergence (Ali and Silvey, 1966), and implement the proposed methods on several models to demonstrate its performance in practice.

This chapter is organized as follows. We first review importance sampling and variational approximation in Section 2.2, and introduce the new method in Section 2.3. Then, we provide theoretical justification in Section 2.4, and give numerical results of the new method on several examples in Section 2.5. Section 2.6 concludes with a discussion.

2.2 Literature Review

2.2.1 Importance sampling

Suppose $\mathbf{Z}$ is a random vector with probability density function $p(\mathbf{z})$, and we want to estimate the expectation of some function $h(\mathbf{Z})$:

$$\boldsymbol{\mu} = E_p(h(\mathbf{Z})) = \int h(\mathbf{z})p(\mathbf{z})d\mathbf{z}.$$

If $p(\mathbf{z})$ is hard to sample directly, we may consider importance sampling (IS) to generate samples from a proposal distribution $q(\mathbf{z})$. Then the expectation $\boldsymbol{\mu}$ can be estimated by the weighted average

$$\tilde{\boldsymbol{\mu}} = \frac{w(\mathbf{z}^{(1)})h(\mathbf{z}^{(1)}) + \cdots + w(\mathbf{z}^{(m)})h(\mathbf{z}^{(m)})}{w(\mathbf{z}^{(1)}) + \cdots + w(\mathbf{z}^{(m)})}, \tag{2.1}$$

where $w(\mathbf{z}^{(i)}) = p(\mathbf{z}^{(i)})/q(\mathbf{z}^{(i)})$ are the importance weights. The estimate $\tilde{\boldsymbol{\mu}}$ is consistent, and it can also handle densities that are only known up to normalizing constants.

The standard error of $\tilde{\boldsymbol{\mu}}$ can be used to measure the efficiency of the IS algorithm. Another criterion is the effective sample size (ESS) (Kong et al., 1994; Kong, 1992; Martino et al., 2017):

$$\text{ESS} = \frac{m}{1 + cv^2},$$

where the coefficient of variation (cv) is defined as:

$$cv^2 = \frac{Var_q[w(\mathbf{Z})]}{E_q^2[w(\mathbf{Z})]}.$$

The ESS roughly approximates the number of independent and identically distributed (i.i.d.) samples these m importance samples are equivalent to. Thus, a smaller cv^2 indicates that the IS algorithm is more effective in terms of the ESS. In addition, the cv^2 is also the χ^2

distance between the proposal distribution $q(\mathbf{z})$ and the target distribution $p(\mathbf{z})$, defined as

$$\chi^2(p||q) = \int \frac{(p-q)^2}{q} d\mathbf{z},$$

and this will be used later in our theoretical justification.

For high dimensional problems, it is often hard to find a good proposal for IS. To overcome this difficulty, Liu and Chen (1998) and Doucet et al. (2000) provided the general framework of sequential importance sampling (SIS) to build up the proposal $q(\mathbf{z})$ sequentially. For a d-dimensional vector $\mathbf{z} = (z_1, \ldots, z_d)$, the proposal distribution can be decomposed as:

$$q(\mathbf{z}) = q_1(z_1) q_2(z_2|z_1) \cdots q_d(z_d|z_1, \ldots, z_{d-1}).$$

Each proposal distribution in the decomposition is for a low dimensional component, so it is relatively easier to design a good proposal. The target distribution $p(\mathbf{z})$ can be decomposed in a similar way by using auxiliary distributions to guide the choice of the proposal distribution (Liu and Chen, 1998). The importance weight can also be computed recursively based on the decomposition. SIS has been successfully applied to many problems, including the filtering problem in hidden Markov models (or state space models).

Another variation of IS is adaptive importance sampling (AIS) (Cappé et al., 2004, 2008; Bugallo et al., 2017), which provides a scheme to find a good proposal distribution adaptively based on samples in previous steps. For multi-modal distributions, Owen (2013) suggested using mixture importance sampling as a way to carry out AIS. However, AIS does not work well for high dimensional distributions without incorporating an additional MCMC layer, and the computation time of AIS is usually much longer than importance sampling (Bugallo et al., 2017).

2.2.2 Variational approximation

Variational Bayesian method (Jordan et al., 1999) is a technique for approximating the intractable integrals in Bayesian inference. It is typically useful when the statistical models are relatively complex with a lot of parameters and latent variables. In Bayesian inference, suppose we have a set of n i.i.d. data $\mathbf{x}$, and all latent variables and parameters are denoted by $\mathbf{Z}$. We need to find an approximation to the posterior distribution $p(\mathbf{z}|\mathbf{x})$ that can minimize the KL divergence, i.e.,

$$q^*(\mathbf{z}) = \underset{q(\mathbf{z})\in\mathcal{D}}{\operatorname{argmin}}\, \mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})),$$

where $\mathcal{D}$ is a restricted distribution family. Here $\mathcal{D}$ is usually a simpler family of distributions to make the optimization and inference tractable.

Xing et al. (2002) assumed the variational distribution $q(\mathbf{z})$ can be factorized over some partitions of the latent variables as follows:

$$q(\mathbf{z}) = \prod_{j=1}^{M} q_j(z_j),$$

where M is the number of parameters and latent variables. The best distribution q_j^* for each factor that solves the optimization problem can be expressed as:

$$q_j^*(z_j) = \frac{e^{E_{-j}[\log p(\mathbf{z},\mathbf{x})]}}{\int e^{E_{-j}[\log p(\mathbf{z},\mathbf{x})]} dz_j} \propto e^{E_{-j}[\log p(z_j,\mathbf{z}_{-j},\mathbf{x})]}. \tag{2.2}$$

Here $E_{-j}[\cdot]$ means the expectation with respect to all $q_i(z_i)$ with $i \neq j$ and $\mathbf{z}_{-j}$ means all the elements in the vector $\mathbf{z}$ except z_j. However, the optimal mean-field variational approximations $q_j^*(z_j)$ can not be computed directly because $E_{-j}[z_i]$ $(i \neq j)$ are involved in the right hand side of (2.2). Thus, an iterative method is often used to obtain the best solution, and such mean-field variational algorithm can only guarantee to converge to a local minimum of $\mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}))$ (Blei et al., 2017).

Beal and Ghahramani (2003) proposed a variational Bayesian EM algorithm to estimate the marginal likelihood of probabilistic models with latent variables or incomplete data. They also compared the variational bound with a sampling-based method known as annealed importance sampling (Neal, 2001). Dieng et al. (2017) proposed another variational algorithm by minimizing the χ-divergence between the variational approximation and the posterior distribution.

2.3 VB Approximation for Importance Sampling

Although obtaining variational approximation is faster than some sampling based methods, such as MCMC, and it learns the approximate probability density functions through optimization, the inference based on the approximation is biased due to the gap between the variational approximation and the true posterior distribution. On the other hand, IS provides a consistent estimate, but the proposal distribution is hard to design. Here we combine VB with IS by using variational approximation $q(\mathbf{z})$ as the proposal distribution for IS. It avoids the bias from VB approximation and also provides a good way to construct the proposal distribution for IS.

Suppose we have a model with prior $p(\mathbf{z})$ and likelihood function $p(\mathbf{x}|\mathbf{z})$, where $\mathbf{z}$ contains both parameters and latent variables, then the posterior distribution is

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{z})p(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})} \propto p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = p(\mathbf{x}, \mathbf{z}).$$

By the mean-field variational algorithm, we can obtain the variational approximation $q(\mathbf{z})$ to the posterior $p(\mathbf{z}|\mathbf{x})$. If the support of $q(\mathbf{z})$ includes the support of $p(\mathbf{z}|\mathbf{x})$, then the expectation of the function $h(\mathbf{Z})$ with respect to $p(\mathbf{z}|\mathbf{x})$ can be estimated by importance sampling as in (2.1), with $w(\mathbf{z}^{(i)}) = p(\mathbf{z}^{(i)}|\mathbf{x}^{(i)})/q(\mathbf{z}^{(i)})$. The variational importance sampling algorithm is summarized in Algorithm 1.

Algorithm 1 Variational importance sampling

1. Obtain the analytical expression of $p(\mathbf{z}|\mathbf{x})$ (up to a normalizing constant)
2. Derive the variational approximation $q(\mathbf{z}) = \prod_{j=1}^{M} q_j(z_j)$ to $p(\mathbf{z}|\mathbf{x})$
3. For $i \in \{1, \ldots, m\}$
4. Draw $\mathbf{z}^{(i)}$ from the proposal distribution $q(\mathbf{z})$
5. Calculate importance weight $w(\mathbf{z}^{(i)}) = p(\mathbf{z}^{(i)}|\mathbf{x}^{(i)})/q(\mathbf{z}^{(i)})$
6. Estimate the expectation of $h(\mathbf{Z})$ with respect to $p(\mathbf{z}|\mathbf{x})$ by (2.1).

Dowling et al. (2018) used the modes of the variational distributions to initialize the location parameters of the proposal distributions in adaptive importance sampling, which is applicable when the variational approximation is in the location scale family. Our proposed method uses the variational approximation itself as the proposal distribution for importance sampling. It does not put restrictions on the proposal distribution, and it can be extended to sequential importance sampling as shown in the next section.

2.3.1 VB approximation for sequential importance sampling

If the dimension of the parameters and latent variables is high, or if the data arrive sequentially, SIS is often used. VB can be combined with SIS as well by constructing the proposal with VB sequentially.

Let $\mathbf{z}$ be all the hidden variables, and $\mathbf{z}_{1:t} = \{z_1, \ldots, z_t\}$ be the first t components. Let $\mathbf{x} = \{x_1, \ldots, x_T\}$ be the data which arrive sequentially. The posterior distribution of interest is $p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$, $t = 1, \ldots, T$. In variational approximation, we assume the approximation $q(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$ can be factorized in the following way:

$$q(\mathbf{z}_{1:t}|\mathbf{x}_{1:t}) = \prod_{k=1}^{t} q(z_k|\mathbf{x}_{1:t}),\ t = 1, \ldots, T.$$

We consider two different approaches for constructing the proposal distribution sequentially.

VB-SIS1. In the first method, at each time $t = 1, \ldots, T$, we minimize the KL divergence

between $q(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$ and the true posterior distribution $p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$, and obtain the variational distributions as follows:

$$\begin{aligned}
q(\mathbf{z}_{1:1}|\mathbf{x}_{1:1}) &= q_1(\mathbf{z}_{1:1}) = q_{11}(z_1), \\
q(\mathbf{z}_{1:2}|\mathbf{x}_{1:2}) &= q_2(\mathbf{z}_{1:2}) = q_{21}(z_1)\, q_{22}(z_2), \\
&\vdots \\
q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) &= q_T(\mathbf{z}_{1:T}) = q_{T1}(z_1)\, q_{T2}(z_2) \cdots q_{TT}(z_T).
\end{aligned}$$

We will use $q_{tt}(z_t)$, $t = 1, 2, \ldots, T$, as the proposal distributions in SIS, and we call this method VB-SIS1 with general procedure given in Algorithm 2.

Algorithm 2 Variational sequential importance sampling 1 (VB-SIS1)

1. Set $w_0(\mathbf{z}_{1:0}^{(i)}) = 1$, $i = 1, \ldots, m$
2. For $t \in \{1, \ldots, T\}$
3. Obtain the analytical expression of $p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$
4. Derive the variational approximation to $p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$ using VB-SIS1:

$$q(\mathbf{z}_{1:t}|\mathbf{x}_{1:t}) = q_{t1}(z_1) q_{t2}(z_2) \cdots q_{tt}(z_t)$$

5. For $i \in \{1, \ldots, m\}$
6. Draw $z_t^{(i)}$ from the proposal distribution $q_{tt}(z_t)$
7. Update importance weight $w_t(\mathbf{z}_{1:t}^{(i)}) = w_{t-1}(\mathbf{z}_{1:t-1}^{(i)}) \frac{p(\mathbf{z}_{1:t}^{(i)}|\mathbf{x}_{1:t})}{p(\mathbf{z}_{1:t-1}^{(i)}|\mathbf{x}_{1:t-1}) q_{tt}(z_t^{(i)})}$
8. Using the sample $\mathbf{z}_{1:t}^{(i)}$, $i = 1, \ldots, m$, and importance weights $w_t(\mathbf{z}_{1:t}^{(i)})$ to estimate the expectation of $h(\mathbf{z}_{1:t})$ with respect to $p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$

VB-SIS2. Another method is to obtain the proposal distribution in the current step t

by reusing the proposals in previous steps. This procedure can be represented as follows:

$$\begin{aligned}
\tilde{q}(\mathbf{z}_{1:1}|\mathbf{x}_{1:1}) &= \tilde{q}_1(\mathbf{z}_{1:1}) = \tilde{q}_1(z_1), \\
\tilde{q}(\mathbf{z}_{1:2}|\mathbf{x}_{1:2}) &= \tilde{q}_2(\mathbf{z}_{1:2}) = \tilde{q}_1(z_1)\,\tilde{q}_2(z_2), \\
&\vdots \\
\tilde{q}(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) &= \tilde{q}_T(\mathbf{z}_{1:T}) = \tilde{q}_1(z_1)\,\tilde{q}_2(z_2)\cdots\tilde{q}_T(z_T).
\end{aligned}$$

At time t, in order to obtain $\tilde{q}_t(\mathbf{z}_{1:t})$, we fix the proposals from previous steps $\tilde{q}_1(z_1), \ldots, \tilde{q}_{t-1}(z_{t-1})$, and obtain $\tilde{q}_t(z_t)$ by minimizing the KL divergence between $\tilde{q}(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$ and the true posterior distribution $p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$. Since we only need to determine the variational distribution for the last latent variable at each step, the running time will be shorter than VB-SIS1. We will use $\tilde{q}_t(z_t)$, $t = 1, \ldots, T$, as the proposal distribution, and we call this method VB-SIS2 with general procedure given in Algorithm 3.

Algorithm 3 Variational sequential importance sampling 2 (VB-SIS2)

1. Set $w_0(\mathbf{z}_{1:0}^{(i)}) = 1$, $i = 1, \ldots, m$
2. For $t \in \{1, \ldots, T\}$
3. Obtain the analytical expression of $p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$
4. Derive the variational approximation to $p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$ using VB-SIS2:

$$q(\mathbf{z}_{1:t}|\mathbf{x}_{1:t}) = \tilde{q}_1(z_1)\tilde{q}_2(z_2)\cdots\tilde{q}_t(z_t)$$

5. For $i \in \{1, \ldots, m\}$
6. Draw $z_t^{(i)}$ from the proposal distribution $\tilde{q}_t(z_t)$
7. Update importance weight $w_t(\mathbf{z}_{1:t}^{(i)}) = w_{t-1}(\mathbf{z}_{1:t-1}^{(i)}) \frac{p(\mathbf{z}_{1:t}^{(i)}|\mathbf{x}_{1:t})}{p(\mathbf{z}_{1:t-1}^{(i)}|\mathbf{x}_{1:t-1})\tilde{q}_t(z_t^{(i)})}$
8. Using the sample $\mathbf{z}_{1:t}^{(i)}$, $i = 1, \ldots, m$, and importance weights $w_t(\mathbf{z}_{1:t}^{(i)})$ to estimate the expectation of $h(\mathbf{z}_{1:t})$ with respect to $p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$

In some cases (such as the hidden Markov model example in Section 2.5.4), we use the

following approximation to further simplify the variational approximation:

$$p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t}) \approx p(\mathbf{z}_{1:t}|\mathbf{x}_{\max(1,t-\Delta+1):t})$$

where Δ is a tuning parameter. This approximation assumes that the observations at time $k < t - \Delta$ almost provide no additional information to $\mathbf{z}_{1:t}$. Under this assumption, we can obtain the variational approximation at step t only based on the observations $\mathbf{x}_{\max(1,t-\Delta+1):t}$, that is,

$$q(\mathbf{z}_{1:t}|\mathbf{x}_{1:t}) = q(\mathbf{z}_{1:t}|\mathbf{x}_{\max(1,t-\Delta+1):t}).$$

Naesseth et al. (2018) considered approximating the posterior distribution for the state space model by introducing variational parameters and resampling procedures. The variational SIS algorithms we proposed are different because we obtain the proposal distribution at each step by deriving variational approximation sequentially. Our variational SIS can be used for general computation based on SIS, including state space models. Adding the resampling procedure can further improve the efficiency of SIS. We will not consider it here because we would like to compare the VB proposal with the standard proposal to evaluate the efficiency gain from VB proposal. Adding resampling steps will make it hard to distinguish where the efficiency gain is coming from. In practice, users can always combine resampling with variational SIS to make it more effective in high dimensional problems.

2.4 Theoretical Justification

To simplify the notation, we will use p and q to denote the true posterior distribution $p(\mathbf{z}|\mathbf{x})$ and the variational distribution $q(\mathbf{z})$ in this section. In variational inference, we minimize the KL divergence between q and p:

$$KL(q||p) = \int q \log \frac{q}{p} d\mathbf{z}.$$

In importance sampling, the cv^2 is the χ^2 distance between p and q:

$$\chi^2(p||q) = \int \frac{(p-q)^2}{q} d\mathbf{z},$$

and we hope to find a proposal distribution q with a relatively small cv^2.

In order to make connections between these two distances, we introduce a more general f-divergence (Ali and Silvey, 1966) between p and q as:

$$D_f(p||q) = E_q\left[f\left(\frac{p}{q}\right)\right] = \int f\left(\frac{p}{q}\right) \cdot q \, d\mathbf{z},$$

where $f(\cdot)$ satisfies the following three conditions:

(i) $f(1) = 0$.

(ii) $f(x)$ is a convex function.

(iii) $f(x)$ is continuous at $x = 1$.

Let $u = p/q$, $f_1(u) = -\log u$ and $f_2(u) = (u-1)^2$, then we can see that the two distances can be written as:

$$KL(q||p) = D_{f_1}(p||q) \qquad \text{and} \qquad \chi^2(p||q) = D_{f_2}(p||q).$$

The Taylor expansion for $f_1(u)$ at $u = 1$ is :

$$f_1(u) = -\log u = -\log(1 + (u-1)) = -(u-1) + \frac{(u-1)^2}{2} - \frac{(u-1)^3}{3} + \cdots.$$

Taking expectation on both sides with respect to q and using the fact $E_q[u] = E_q[p/q] = 1$, we obtain the following equations

$$KL(q||p) = \frac{1}{2}\chi^2(p||q) + o((u-1)^2).$$

This indicates that when u is close to 1, these two distances are equivalent, i.e.,

$$KL(q||p) \asymp \frac{1}{2}\chi^2(p||q).$$

In order to quantify the value of u, we introduce two quantities β_1 and β_2 as follows (Sason and Verdú, 2016):

$$\beta_1 = \operatorname{ess\,inf} \frac{q}{p}, \quad \beta_2 = \operatorname{ess\,inf} \frac{p}{q}. \tag{2.3}$$

The essential infimum and the essential supremum are defined as:

$$\operatorname{ess\,inf} \frac{p}{q} = \sup\{b \in \mathbb{R} : \mu(\{x : p(x)/q(x) < b\}) = 0\},$$

$$\operatorname{ess\,sup} \frac{p}{q} = \inf\{a \in \mathbb{R} : \mu(\{x : p(x)/q(x) > a\}) = 0\},$$

where $\mu(\cdot)$ denotes the Lebesgue measure.

Since $\int q(\mathbf{z})\, d\mathbf{z} = 1$ and $\int p(\mathbf{z}|\mathbf{x})\, d\mathbf{z} = 1$, we have $0 \leq \beta_1, \beta_2 \leq 1$, and $\beta_1 = 1 \Leftrightarrow \beta_2 = 1 \Leftrightarrow p = q$. Suppose $0 < \beta_1 < 1$ and $0 < \beta_2 < 1$. We say a sequence of probability measures with densities p_n converge to q if

$$\lim_{n\to\infty} \operatorname{ess\,inf} \frac{p_n}{q} = 1. \tag{2.4}$$

Lemma 2.1. *Suppose f is a function satisfying Conditions (i)-(iii), and a sequence of probability measures with densities p_n converge to q in the sense of (2.4). Let*

$$\beta_{1,n}^{-1} = ess\sup \frac{p_n}{q}, \quad \beta_{2,n} = ess\inf \frac{p_n}{q}.$$

Then we have

$$\lim_{n\to\infty} \beta_{1,n} = \lim_{n\to\infty} \beta_{2,n} = 1,$$

and

$$\lim_{n\to\infty} D_f(p_n||q) = 0.$$

The proof of the lemma as well as the proof of the following theorem are in Section 2.7. Define a function:

$$\kappa(t) = \frac{t \log t + (1-t)}{(t-1) - \log t}, \qquad 0 < t < 1, \tag{2.5}$$

which is increasing for $0 < t < 1$. Then from Sason and Verdú (2016), the following inequalities hold:

$$\kappa(\beta_{2,n}) \leq \frac{KL(p_n||q)}{KL(q||p_n)} \leq \kappa(\beta_{1,n}^{-1}), \tag{2.6}$$

$$\frac{1}{2}\beta_{1,n} \leq \frac{KL(p_n||q)}{\chi^2(p_n||q)} \leq \frac{1}{2}\beta_{2,n}^{-1}, \tag{2.7}$$

where p_n, $\beta_{1,n}$, and $\beta_{2,n}$ are defined in Lemma 2.1. The following theorem gives the limit of the ratios in (2.6) and (2.7).

Theorem 2.2. *Suppose a sequence of probability measures with densities p_n converge to q in the sense of (2.4). For KL divergence and χ^2 distance, we have*

$$\lim_{n\to\infty} \frac{KL(p_n||q)}{KL(q||p_n)} = 1, \quad \lim_{n\to\infty} \frac{KL(p_n||q)}{\chi^2(p_n||q)} = \frac{1}{2}.$$

From the above theorem, we immediately have the following corollary.

Corollary 2.3. *Suppose a sequence of probability measures with densities p_n converge to q in the sense of (2.4). For KL divergence and χ^2 distance, we have*

$$\lim_{n\to\infty} \frac{KL(q||p_n)}{\chi^2(p_n||q)} = \frac{1}{2}.$$

When we consider a proposal distribution q in importance sampling, we have from (2.6) and (2.7) that

$$2\beta_2\kappa(\beta_2) \leq \frac{\chi^2(p||q)}{KL(q||p)} \leq 2\beta_1^{-1}\kappa(\beta_1^{-1}). \tag{2.8}$$

Therefore the upper and lower bounds for χ^2 distances are

$$2\beta_2\kappa(\beta_2)KL(q||p) \leq \chi^2(p||q) \leq 2\beta_1^{-1}\kappa(\beta_1^{-1})KL(q||p). \tag{2.9}$$

Our goal is to find a proposal distribution q close to the target distribution p in terms of the χ^2 distance $\chi^2(p||q)$. The relation in (2.8) indicates that it is reasonable to use the distribution q that minimizes the KL divergence $KL(q||p)$ as the proposal distribution. This justifies the use of VB solution as the proposal distribution for importance sampling. A smaller upper bound in (2.9) often indicates that the corresponding proposal distribution has better performance in importance sampling, so the upper bound can give us an intuitive way to evaluate the choice of the proposal distribution. This idea is illustrated in the example in Section 2.5.1 by computing β_1 and β_2 explicitly. However, the exact values of β_1 and β_2 are hard to calculate in some complex models.

2.5 Numerical Results

All examples in this section were coded in R and run on a MacBook Pro with 2.3 GHz Intel Core i7 processor.

2.5.1 Univariate normal

This toy example is on Bayesian inference for a univariate normal distribution. Suppose our observed data $\mathbf{x} = \{x_1, \ldots, x_N\}$ is a random sample from a normal distribution with mean μ and precision τ. We use the normal-gamma conjugate prior for μ and τ as follows:

$$p(\mu|\tau) = \mathcal{N}(\mu_0, (\lambda_0\tau)^{-1}), \qquad p(\tau) = \text{Gamma}(a_0, b_0).$$

We consider a factorized variational approximation to the posterior distribution $q(\mu, \tau) = q_\mu(\mu)q_\tau(\tau)$. The variational approximation algorithm gives $q_\mu(\mu) \sim \mathcal{N}(\nu, \lambda^{-1})$ with the mean and precision:

$$\nu = \frac{\lambda_0\mu_0 + N\bar{x}}{\lambda_0 + N} \quad \text{and} \quad \lambda = (\lambda_0 + N)E[\tau],$$

and $q_\tau(\tau) \sim \text{Gamma}(a, b)$ with two parameters:

$$a = a_0 + \frac{N}{2}, \quad b = b_0 + \frac{1}{2} E_\mu \left[\sum_{i=1}^{N} (x_n - \mu)^2 + \lambda_0 (\mu - \mu_0)^2 \right].$$

If we follow the updating rules and compute the expectation with the parameter values from the previous step, we can obtain the variational distribution $q(\mu, \tau)$ as in Algorithm 4.

Algorithm 4 Variational algorithm for univariate normal

1. Initialize $b = 1$, $\lambda = 1$
2. Calculate $\nu = \frac{\lambda_0 \mu_0 + N\bar{x}}{\lambda_0 + N}$ and $a = a_0 + \frac{N}{2}$
3. Repeat the following until convergence
4. $\quad \lambda = (\lambda_0 + N)\frac{a}{b}$
5. $\quad b = b_0 + \frac{1}{2}\left[(\sum_{i=1}^{N} x_n^2 + \lambda_0 \mu_0^2) - (2\sum_{i=1}^{N} x_n + 2\lambda_0 \mu_0)\nu + (\lambda_0 + N)(\nu^2 + \frac{1}{\lambda}) \right]$

We set the hyperparameters $\mu_0 = 1$, $\lambda_0 = 1$, $a_0 = 1$, $b_0 = 1$, and generated $N = 50$ data points from $N(1, 1)$. For this simple example, the true posterior distribution $p(\mu, \tau | \mathbf{x})$ can be derived as

$$\begin{aligned} p(\mu | \tau, \mathbf{x}) &= \mathcal{N}\left(\frac{\lambda_0 \mu_0 + N\bar{x}}{\lambda_0 + N}, (\lambda_0 + N)^{-1} \right), \\ p(\tau | \mathbf{x}) &= \text{Gamma}\left(a_0 + \frac{N}{2}, b_0 + \frac{1}{2}\left[\sum_{i=1}^{N} (x_i - \bar{x})^2 + \frac{\lambda_0 N (\bar{x} - \mu_0)^2}{\lambda_0 + N} \right] \right). \end{aligned}$$

The contour plots in Figure 2.1 show some resemblance between the true posterior distribution and the VB approximation.

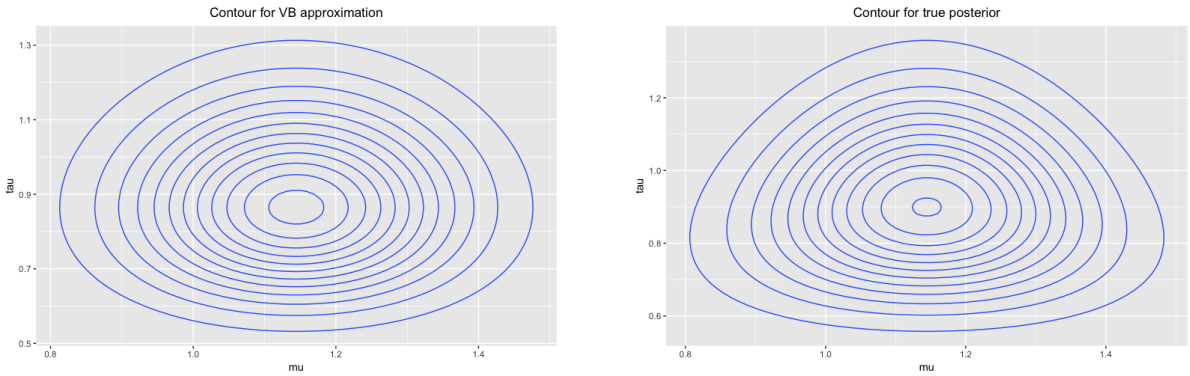


Figure 2.1: Contour plots for the true posterior and the VB approximation

We compared the performance of different methods in Table 2.1, including the variational Bayes method (denoted by "VB"), IS with variational distribution as the proposal (denoted by "VB as proposal"), IS with the prior as the proposal (denoted by "Prior as proposal"), and adaptive importance sampling (denoted by "AIS") (Bugallo et al., 2017). The variational distributions are well-known standard distributions in this example, and the expectations are easy to compute. The three IS algorithms are based on $m = 100,000$ samples, and the numbers in parentheses are the standard errors. The true posterior mean is also provided (denoted by "True mean").

Parameter	VB	VB as proposal	Prior as proposal	AIS	True mean
μ	1.1445	1.1453 (0.0007)	1.1448 (0.0226)	1.1443 (0.0021)	1.1445
τ	0.8992	0.9170 (0.0006)	0.9192 (0.0183)	0.9181 (0.0015)	0.9169

Table 2.1: Simulation results for the univariate normal example.

Table 2.1 shows that IS with variational distribution as proposal gives much smaller standard errors than IS with prior as the proposal and AIS. The computation time of AIS is much longer than IS with VB or prior as the proposal, since AIS needs to update the proposal distribution adaptively based on samples from previous steps, while the variational distribution and the prior are relatively easier to obtain. Using variational method directly gives a biased estimate for τ (the estimate for μ happens to be the same as the true mean),

and the variability of the estimate is unknown.

Since the true posterior distribution is known in this example, we can calculate β_1 and β_2 defined in (2.3). The values of β_1 and β_2, which are presented in Table 2.2, are related to the ratio between the posterior distribution p and the proposal distribution q. They also appear in the upper and lower bounds of the χ^2 distance between p and q in (2.8) and (2.9). Since β_1^{-1} is smaller when VB is the proposal and $\kappa(t)$ is an increasing function for $0 < t < 1$, that implies the upper bounds $2\beta_1^{-1}\kappa(\beta_1^{-1})$ in (2.8) and $2\beta_1^{-1}\kappa(\beta_1^{-1})KL(q||p)$ in (2.9) are smaller when VB is the proposal (note that $KL(q||p)$ is minimized for VB proposal). Similarly, VB proposal has a larger β_2 which implies $2\beta_2\kappa(\beta_2)$ in the lower bound in (2.8) and (2.9) is larger for the VB proposal. All these suggest that using VB as the proposal may lead to a smaller χ^2 distance and better performance.

	VB as proposal	Prior as proposal
β_1^{-1}	1.751	2.513
β_2	0.673	0.282

Table 2.2: The values of β_1^{-1} and β_2 for the univariate normal example.

2.5.2 Gaussian mixture model

Suppose we have N i.i.d. observations $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ from a Gaussian mixture distribution, and each $\mathbf{x}_i$ is a D-dimensional vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iD})^T$. Suppose there are K mixture components and $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_K)$ denotes the mixture proportions. The labels that indicate the membership of the observations are denoted by the latent variables $\mathbf{z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\}$, where $\mathbf{z}_i \sim \text{Multinomial}(1, \boldsymbol{\pi})$. In other words, $\mathbf{z}_i = (z_{i1}, z_{i2}, \ldots, z_{iK})^T$ is a K-dimensional vector with one element equal to 1, which specifies the label of $\mathbf{x}_i$, and all other elements equal to 0. If the k-th element of $\mathbf{z}_i$ is 1, we write

$$\mathbf{x}_i | z_{ik} = 1 \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}),$$

where μ_i and $\mathbf{\Lambda}_i$ are the mean and precision matrix of each multivariate Gaussian component.

We use a symmetric Dirichlet distribution with hyperparameter α_0 as the prior distribution for $\boldsymbol{\pi}$:

$$p(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0) = C(\boldsymbol{\alpha}_0)\prod_{k=1}^{K}\pi_k^{\alpha_0}, \quad \text{where } \boldsymbol{\alpha}_0 = (\alpha_0, \alpha_0, \ldots, \alpha_0).$$

For the mean vector $\boldsymbol{\mu}_k$ and the precision matrix $\mathbf{\Lambda}_k$, we use a normal-Wishart prior distribution as the conjugate prior for these two parameters:

$$\mathbf{\Lambda}_k \sim \text{Wishart}(\boldsymbol{W}_0, \nu_0) \quad \Rightarrow \quad p(\mathbf{\Lambda}) = \prod_{k=1}^{K}\mathcal{W}(\mathbf{\Lambda}_k|\boldsymbol{W}_0, \nu_0),$$

$$\boldsymbol{\mu}_k \sim \mathcal{N}\left(\boldsymbol{\mu}_0, (\beta_0\mathbf{\Lambda}_k)^{-1}\right) \quad \Rightarrow \quad p(\boldsymbol{\mu}|\mathbf{\Lambda}) = \prod_{k=1}^{K}\mathcal{N}\left(\boldsymbol{\mu}_k|\boldsymbol{\mu}_0, (\beta_0\mathbf{\Lambda}_k)^{-1}\right),$$

where $\mathbf{\Lambda} = (\mathbf{\Lambda}_1, \mathbf{\Lambda}_2 \ldots, \mathbf{\Lambda}_K)$ and $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K)$. The likelihood function of the Gaussian mixture model is

$$p(\mathbf{z}|\boldsymbol{\pi}) = \prod_{i=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{ik}},$$

$$p(\mathbf{x}|\boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{\Lambda}) = \prod_{i=1}^{N}\left(\sum_{k=1}^{K}\pi_k\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \mathbf{\Lambda}_k^{-1})\right).$$

The posterior distribution is

$$p(\boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{\Lambda}|\mathbf{x}) \propto p(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{\Lambda}) = p(\mathbf{x}|\boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{\Lambda})p(\boldsymbol{\pi})p(\boldsymbol{\mu}|\mathbf{\Lambda})p(\mathbf{\Lambda}).$$

For variational approximation, following Bishop (2006) we first factorize $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{\Lambda})$ into the following variational distribution:

$$q(\boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{\Lambda}) = q(\boldsymbol{\pi})\prod_{k=1}^{K}q(\boldsymbol{\mu}_k, \mathbf{\Lambda}_k).$$

After calculating the logarithm of the optimal distribution, we get:

$$\ln q^*(\boldsymbol{\pi}) = (\alpha_0 - 1)\sum_{k=1}^{K} \ln \pi_k + \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik} \ln \pi_k$$

$$\Rightarrow \qquad q^*(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \qquad \text{where} \;\; \alpha_k = \alpha_0 + N_k \;\; \text{and} \;\; N_k = \sum_{i=1}^{N} r_{ik}.$$

Then we further decompose the variational distribution as $q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) = q^*(\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k)q^*(\boldsymbol{\Lambda}_k)$, and the variational joint posterior distribution of $(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$ is also normal-Wishart distribution with different parameters from the prior distributions:

$$q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) = \mathcal{N}\left(\boldsymbol{\mu}_k|\mathbf{m}_k, (\beta_k\boldsymbol{\Lambda}_k)^{-1}\right) \mathcal{W}(\boldsymbol{\Lambda}_k|\mathbf{W}_k, \nu_k).$$

If we follow the updating rules for each parameter, we can obtain the variation approximation for Gaussian mixture model as in Algorithm 5.

Algorithm 5 Variational algorithm for Gaussian mixture model

1. Initialize $\boldsymbol{\alpha}$, $\bar{\mathbf{x}}_k$, $\mathbf{W}_k$, $\mathbf{m}_k$, $\mathbf{S}_k$ and r_{ik}

2. Repeat the following steps until convergence

3. Calculate $N_k = \sum_{i=1}^{N} r_{ik}$ and update $\boldsymbol{\alpha}$ by $\alpha_k = \alpha_0 + N_k$

4. Update $\bar{\mathbf{x}}_k$ and $\mathbf{S}_k$ by

$$\bar{\mathbf{x}}_k = \frac{1}{N_k}\sum_{i=1}^{N} r_{ik}\mathbf{x}_i \quad \text{and} \quad \mathbf{S}_k = \frac{1}{N_k}\sum_{i=1}^{N} r_{ik}(\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^{\mathrm{T}}$$

5. Update $\mathbf{W}_k$ and ν_k by

$$\mathbf{W}_k^{-1} = \mathbf{W}_0^{-1} + N_k\mathbf{S}_k + \frac{\beta_0 N_k}{\beta_0 + N_k}(\bar{\mathbf{x}}_k - \boldsymbol{\mu}_0)(\bar{\mathbf{x}}_k - \boldsymbol{\mu}_0)^{\mathrm{T}} \quad \text{and} \quad \nu_k = \nu_0 + N_k$$

6. Update $\mathbf{m}_k$ and β_k by

$$\mathbf{m}_k = \frac{1}{\beta_k}(\beta_0\boldsymbol{\mu}_0 + N_k\bar{\mathbf{x}}_k) \text{ and } \beta_k = \beta_0 + N_k$$

7. Update r_{ik} by

$$\rho_{ik} = \exp\left(-\frac{D}{2\beta_k} - \frac{\nu_k}{2}(\mathbf{x}_i - \mathbf{m}_k)^T\mathbf{W}_k(\mathbf{x}_i - \mathbf{m}_k)\right) \quad \text{and} \quad r_{ik} = \frac{\rho_{ik}}{\sum_{j=1}^{K}\rho_{ik}}$$

8. Variational distribution is $q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) = \mathcal{N}(\boldsymbol{\mu}_k|\mathbf{m}_k, (\beta_k\boldsymbol{\Lambda}_k)^{-1})\,\mathcal{W}(\boldsymbol{\Lambda}_k|\mathbf{W}_k, \nu_k)$ and $q^*(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha})$.

In the following simulation, we fix the hyperparameters $\alpha_0 = 1$, $\beta_0 = 5$, $\boldsymbol{\mu}_0 = \mathbf{0}$, $\mathbf{W}_0 = \mathbf{I}_D$, and $\nu_0 = 5$. Tables 2.3-2.5 show the results for different combinations of the dimension of the data D and the number of mixture components K. The variational distributions are well-known standard distributions in this example, and the expectations of all parameters are

easy to compute when applying VB directly. The two IS algorithms are based on $m = 10,000$ samples. The last column denotes the true parameters when we generated the observed data. The 'True mean' is an estimate of the true posterior mean based on $1,000,000$ samples from importance sampling with VB approximation as the proposal.

Parameter	VB	VB as proposal	Prior as proposal	'True mean'	True parameter
ω_1	0.7816	0.7735 (0.004)	0.6645 (0.125)	0.7751	0.7
ω_2	0.2183	0.2265 (0.004)	0.3354 (0.125)	0.2349	0.3
μ_1	-2.6710	-2.6663 (0.007)	-1.1485 (0.892)	-2.6632	-3
μ_2	1.9445	1.8699 (0.032)	0.1613 (1.193)	1.8473	3
Λ_1	0.2924	0.2865 (0.006)	1.4192 (0.726)	0.2781	1
Λ_2	0.6862	0.6815 (0.005)	0.4463 (0.316)	0.6766	1

Table 2.3: Simulation results for Gaussian mixture model with $D = 1$, $K = 2$, and $\alpha_0 = 1$.

Parameter	VB	VB as proposal	Prior as proposal	'True mean'	True parameter
ω_1	0.4991	0.4816 (0.012)	0.4494 (0.088)	0.4835	0.5
ω_2	0.2658	0.2882 (0.010)	0.3910 (0.125)	0.2901	0.3
ω_3	0.2350	0.2300 (0.014)	0.1594 (0.754)	0.2264	0.2
μ_1	-3.6698	-3.6905 (0.102)	-1.0197 (0.217)	-3.6102	-5
μ_2	-0.1359	-0.1700 (0.021)	0.0080 (0.157)	-0.1581	0
μ_3	2.6100	2.3060 (0.102)	0.3003 (0.136)	2.4152	5
Λ_1	3.5765	3.9011 (0.153)	6.9669 (1.833)	3.7530	1
Λ_2	0.1936	0.1887 (0.006)	4.7721 (2.148)	0.1852	1
Λ_3	0.1600	0.1372 (0.005)	0.9650 (0.813)	0.1462	1

Table 2.4: Simulation results for Gaussian mixture model with $D = 1$, $K = 3$, and $\alpha_0 = 1$.

Parameter	VB	VB as proposal	Prior as proposal	'True mean'	True parameter
ω_1	0.7304	0.7320 (0.002)	0.6175 (0.125)	0.7331	0.7
ω_2	0.2695	0.2680 (0.002)	0.3824 (0.125)	0.3669	0.3
μ_{11}	-2.6889	-2.6887 (0.005)	-0.6543 (1.412)	-2.6875	-3
μ_{21}	-2.6707	-2.6676 (0.006)	-0.1346 (1.617)	-2.6629	-3
μ_{12}	2.0117	2.0057 (0.012)	0.5534 (0.925)	1.9725	3
μ_{22}	1.5919	1.5781 (0.011)	-0.0810 (0.825)	1.5864	3
Λ_{111}	0.5793	0.5584 (0.006)	0.8220 (0.183)	0.5623	1
Λ_{121}	-0.2919	-0.2816 (0.005)	-1.5826 (0.902)	-0.2759	0
Λ_{221}	0.7004	0.6873 (0.007)	1.9365 (1.025)	0.6871	1
Λ_{112}	1.9014	1.9175 (0.014)	5.7718 (2.245)	1.9201	2
Λ_{122}	-1.6550	-1.6794 (0.014)	-2.7911 (0.725)	-1.6803	-1
Λ_{222}	2.2336	2.2620 (0.016)	3.1288 (0.616)	2.2712	3

Table 2.5: Simulation results for Gaussian mixture model with $D = 2$, $K = 2$, and $\alpha_0 = 1$.

From Tables 2.3-2.5, we can see that IS with variational distribution as proposal gives smaller standard errors than IS with prior as the proposal. In addition, using VB directly will introduce bias to the estimates.

2.5.3 Linear regression model

Let $\{(y_i, \mathbf{x}_i)\}_{i=1}^N$ be the observed pairs of data, where $\mathbf{x}_i \in \mathbb{R}^p$. Consider the linear regression model

$$y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i,$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ and $\epsilon_i \sim N(0, \sigma^2)$. The likelihood function is

$$\mathbf{y}|\boldsymbol{\beta}, \sigma^2 \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}),$$

where $\mathbf{y} = (y_1, y_2, \ldots, y_N)^T$, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T$, and $\mathbf{I}$ is the identity matrix. Similar to You et al. (2014), we use inverse gamma and normal conjugate priors for $\boldsymbol{\beta}$ and σ^2 as follows:

$$\sigma^2 \sim \text{Inv-Gamma}(A, B), \quad \boldsymbol{\beta} \sim N(\mathbf{0}, \sigma^2_{\boldsymbol{\beta}}\mathbf{I}),$$

where $A, B, \sigma^2_{\boldsymbol{\beta}}$ are hyperparameters.

Let $\mathbf{z}$ be all parameters of interests, i.e., $\mathbf{z} = [\boldsymbol{\beta}^T, \sigma^2]^T$. We consider a factorized variational approximation $q^*(\mathbf{z}) = q^*_{\boldsymbol{\beta}}(\boldsymbol{\beta})q^*_{\sigma^2}(\sigma^2)$. Since we chose the conjugate priors for $\mathbf{z}$, the variational distributions can be written as:

$$q^*_{\boldsymbol{\beta}}(\boldsymbol{\beta}) \sim N(\boldsymbol{\mu}_{q(\boldsymbol{\beta})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\beta})}), \quad q^*_{\sigma^2}(\sigma^2) \sim \text{Inv-Gamma}\left(A + n/2, B_{q(\sigma^2)}\right).$$

By solving the optimization problem iteratively, we can obtain the updating rules of all the parameters, as well as the corresponding variational algorithm in Algorithm 6.

Algorithm 6 Variational algorithm for linear regression model

1. Initialize $\boldsymbol{\Sigma}_{q(\boldsymbol{\beta})} = I_p$, $\boldsymbol{\mu}_{q(\boldsymbol{\beta})} = \mathbf{1}^T$, $B_{q(\sigma^2)} = 1$
2. Repeat the following until convergence
3. Update $\boldsymbol{\Sigma}_{q(\boldsymbol{\beta})}$:

$$\boldsymbol{\Sigma}_{q(\boldsymbol{\beta})} = \left[\left(\frac{A + n/2}{B_{q(\sigma^2)}}\right)\mathbf{X}^T\mathbf{X} + \sigma^{-2}_{\boldsymbol{\beta}}\mathbf{I}\right]^{-1}$$

4. Update $\boldsymbol{\mu}_{q(\boldsymbol{\beta})}$:

$$\boldsymbol{\mu}_{q(\boldsymbol{\beta})} = \left(\frac{A + n/2}{\mathbf{B}_{q(\sigma^2)}}\right)\boldsymbol{\Sigma}_{q(\boldsymbol{\beta})}\mathbf{X}^T\mathbf{y}$$

5. Update $B_{q(\sigma^2)}$:

$$B_{q(\sigma^2)} = B + \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\mu}_{q(\boldsymbol{\beta})}||^2 + \frac{1}{2}\text{tr}\left(\mathbf{X}^T\mathbf{X}\boldsymbol{\Sigma}_{q(\boldsymbol{\beta})}\right)$$

In the simulation, we generated $N = 50$ data pairs from the following true model

$$y = 3 + 0 \cdot x_1 - 3x_2 + 5x_3 + \epsilon \quad , \quad \epsilon \sim N(0, \sigma^2),$$

where x_1 has no influence on the response variable y. We fix the hyperparameters $\sigma_{\boldsymbol{\beta}} = 2$, $A = 2$, and $B = 5$. The variational distribution obtained from Algorithm 6 is used to estimate the parameters directly and also as the proposal for IS. The two IS algorithms with different proposals are both based on $m = 10,000$ samples. The 'True mean' is an estimate of the true posterior mean based on $1,000,000$ samples from IS with VB approximation as the proposal.

Parameter	VB	VB as proposal	Prior as proposal	'True mean'	True parameter
β_0	2.9339	2.9487 (0.034)	2.7226 (0.174)	2.9027	3
β_1	-0.0963	-0.0480 (0.054)	-0.3141 (0.262)	-0.0732	0
β_2	-2.7448	-2.7102 (0.050)	-1.9892 (0.482)	-2.7025	-3
β_3	4.4420	4.3498 (0.069)	3.2586 (0.565)	4.3713	5
σ^2	5.8314	5.8533 (0.138)	7.9202 (1.401)	5.8521	4

Table 2.6: Simulation results for linear regression model

Table 2.6 shows that IS with variational distribution as proposal gives smaller standard errors than IS with prior as the proposal. Using variational method directly gives a biased estimate and variability of the estimate is unknown. For example, using VB directly gives an estimate of -0.0963 for β_1 without quantification of the uncertainty of the estimate, so it is hard to tell whether the true value of β_1 is 0. On the other hand, the 95% confidence interval of the estimates based on both IS algorithms contain 0, which indicates that β_1 is not significant in the linear model.

2.5.4 Hidden Markov model

The hidden Markov model (HMM) consists of a Markov chain with hidden states $\mathbf{z} = \{z_0, z_1, z_2, \ldots, z_T\}$ and an observed sequence of data $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$, where z_0 is the initial state, and T is the length of the sequence. The hidden states evolve according to

$$Z_t|(Z_{t-1} = z_{t-1}) \sim f(z_t|z_{t-1}),$$

and the dependence between the observed data and hidden state can be represented as

$$X_t|(Z_t = z_t) \sim g(x_t|z_t).$$

Given the observed data, the posterior distribution of the hidden states can be written as:

$$p(\mathbf{z}_{0:T}|\mathbf{x}_{1:T}) = \frac{p(\mathbf{z}_{0:T}, \mathbf{x}_{1:T})}{p(\mathbf{x}_{1:T})} \propto p(\mathbf{z}_{0:T})p(\mathbf{x}_{1:T}|\mathbf{z}_{0:T}),$$

where

$$p(\mathbf{z}_{0:T}) = f(z_0)\prod_{t=1}^{T} f(z_t|z_{t-1}) \quad \text{and} \quad p(\mathbf{x}_{1:T}|\mathbf{z}_{0:T}) = \prod_{t=1}^{T} g(x_t|z_t).$$

We consider the filtering problem, which is to infer $\mathbf{z}_{1:t}$ from the observations $\mathbf{x}_{1:t}$, $t = 1, \ldots, T$. When applying SIS to the filtering problem, the naive choice of the proposal distribution is to sample z_t from $f(z_t|z_{t-1})$. However, this proposal is not very efficient because it does not take into account the information contained in the observations.

The two variational approximations in Section 2.3.1, VB-SIS1 and VB-SIS2, can be used to construct better proposals for SIS. The corresponding algorithm is the same as Algorithms 2 and 3, and the weight updating step for HMM can be written explicitly as

$$w_t(\mathbf{z}_{1:t}^{(i)}) = w_{t-1}(\mathbf{z}_{1:t-1}^{(i)})\frac{p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})}{p(\mathbf{z}_{1:t-1}|\mathbf{x}_{1:t-1})q_{tt}(z_t^{(i)})} = w_{t-1}(\mathbf{z}_{1:t-1}^{(i)})\frac{g(x_t|z_t^{(i)})f(z_t^{(i)}|z_{t-1}^{(i)})}{q_{tt}(z_t^{(i)})},$$

or

$$w_t(\mathbf{z}_{1:t}^{(i)}) = w_{t-1}(\mathbf{z}_{1:t-1}^{(i)}) \frac{p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})}{p(\mathbf{z}_{1:t-1}|\mathbf{x}_{1:t-1})\tilde{q}_t(z_t^{(i)})} = w_{t-1}(\mathbf{z}_{1:t-1}^{(i)}) \frac{g(x_t|z_t^{(i)}) f(z_t^{(i)}|z_{t-1}^{(i)})}{\tilde{q}_t(z_t^{(i)})}.$$

We study two examples below, one is a discrete HMM and the other one is a continuous HMM.

Discrete hidden Markov model

In the discrete HMM example, assume $z_t \in \{1, 2, \ldots, K\}$ and $x_t \in \{1, 2, \ldots, W\}$. Then the model can be specified by two matrices: transition matrix $\mathbf{A}_{K\times K}$ and emission matrix $\mathbf{B}_{K\times W}$, where A_{ij} denotes the probability of transitioning from state i to state j and B_{kw} denotes the probability of emitting observation w from state k. We propose the variational approximation similar to Wang and Blunsom (2013).

In the simulation study, we set $z_0 = 1$, $K = 3$ and $W = 4$, i.e., $z_t \in \{1, 2, 3\}$ and $x_t \in \{1, 2, 3, 4\}$. The transition and emission matrices are chosen to be:

$$A = \begin{bmatrix} 0.1 & 0.4 & 0.5 \\ 0.4 & 0.2 & 0.4 \\ 0.6 & 0.2 & 0.2 \end{bmatrix}, \; B = \begin{bmatrix} 0.3 & 0.3 & 0.3 & 0.1 \\ 0.4 & 0.1 & 0.2 & 0.3 \\ 0.1 & 0.6 & 0.2 & 0.1 \end{bmatrix}.$$

We considered different combinations of the length of the sequence T, the number of samples m, and the tuning parameter Δ. The results are presented in Tables 2.7 and 2.8 and Figure 2.2.

Proposal	m	cv^2	Time (seconds)
$f(z_t\|z_{t-1})$	1000	321.0979	0.8
VB-SIS1 $\Delta = 7$	1000	78.0338	235
VB-SIS2 $\Delta = 7$	1000	205.3263	52
$f(z_t\|z_{t-1})$	5000	342.0129	4.2
VB-SIS1 $\Delta = 7$	5000	75.1225	251
VB-SIS2 $\Delta = 7$	5000	202.2352	63
$f(z_t\|z_{t-1})$	30000	336.1599	20.6
VB-SIS1 $\Delta = 7$	30000	77.9406	306
VB-SIS2 $\Delta = 7$	30000	208.3262	75

Table 2.7: Simulation results for discrete HMM with $\Delta = 7$, $T = 50$, and varying sample size m

Proposal	T	cv^2	Time (seconds)
$f(z_t\|z_{t-1})$	30	97.0153	3.1
VB-SIS1 $\Delta = 7$	30	18.0764	149
VB-SIS2 $\Delta = 7$	30	45.6237	34
$f(z_t\|z_{t-1})$	50	342.0129	4.2
VB-SIS1 $\Delta = 15$	50	75.1225	335
VB-SIS2 $\Delta = 15$	50	202.2352	63
$f(z_t\|z_{t-1})$	100	1252.2339	8.3
VB-SIS1 $\Delta = 32$	100	193.3824	703
VB-SIS2 $\Delta = 32$	100	527.2363	233

Table 2.8: Simulation results for discrete HMM with $m = 5000$ and varying length of sequence T

From Table 2.7, we can see that if we fix Δ and the length of sequence T, the cv^2 for each method will not change much when we increase the number of samples m. Table 2.8

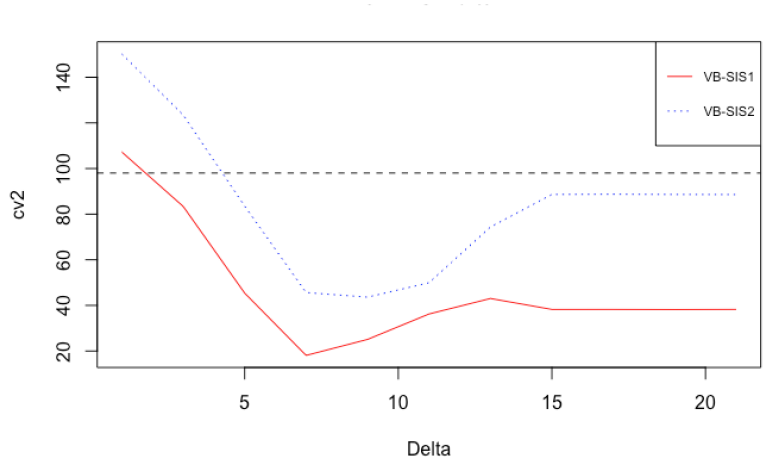


Figure 2.2: cv^2 for variational SIS for discrete HMM with $m = 5000$, $T = 30$, and varying tuning parameter Δ

shows that if we fix m, then T will influence both cv^2 and the computation time a lot. In general, using the state evolution $f(z_t|z_{t-1})$ takes less time, but the cv^2 is large. VB-SIS1 gives the smallest cv^2, but the computation time is the longest. The performance of VB-SIS2 is somewhere between the other two methods. Note that after the data are generated, we only need to compute the variational approximation once, so this time-consuming step will not be influenced by the sample size m. Figure 2.2 shows how the cv^2 of importance sampling changes with the value of Δ. The horizontal dashed line is the cv^2 when the state evolution $f(z_t|z_{t-1})$ is used as the proposal, and it can serve as a benchmark.

Stochastic volatility model

The stochastic volatility model consists of the following state equation and observation equation:

$$Z_t = \alpha Z_{t-1} + \sigma V_t, \quad X_t = \beta \exp(Z_t/2) W_t,$$

where $V_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,1)$, $W_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,1)$, and both the hidden state Z_t and the observation X_t are continuous real-valued random variables.

In the simulation study, the initial state $Z_0 \sim \mathcal{N}(0, \sigma^2/(1-\alpha^2))$, and we set $\alpha = 0.3$, $\sigma = 5$ and $\beta = 2$. In this case, the variational distributions $\{q_t(z_t)\}_{t=1}^T$ also follow the normal

distribution. We considered different combinations of the length of the sequence T, the number of samples m, and the tuning parameter Δ. The results are in Tables 2.9 and 2.10.

From Table 2.9, we can see that if we fix Δ and the length of sequence T, the cv^2 for each method will not change much when we increase the number of samples m. Table 2.10 shows that if we increase the length of the observed sequence T, then the cv^2 increases for all proposal distributions we tested. Tables 2.9 and 2.10 indicate that using the state evolution $f(z_t|z_{t-1})$ as the proposal distribution takes less time, but the cv^2 is relatively large. VB-SIS1 gives the smallest cv^2, but the computation time is the longest. The performance of VB-SIS2 is somewhere between the other two methods. If we fix the running time, VB-SIS2 has a larger effective sample size than VB-SIS1.

Proposal	Estimate (s.e.)	m	cv^2	Time (seconds)
$f(z_t\|z_{t-1})$	15.323 (1.42)	1000	151.0883	0.7
VB-SIS1 $\Delta = 7$	13.897 (0.97)	1000	48.0338	15.3
VB-SIS2 $\Delta = 7$	13.627 (1.23)	1000	68.5262	3.6
$f(z_t\|z_{t-1})$	14.984 (0.42)	5000	134.9283	3.2
VB-SIS1 $\Delta = 7$	14.714 (0.25)	5000	45.1735	17.7
VB-SIS2 $\Delta = 7$	14.642 (0.36)	5000	62.2415	5.7
$f(z_t\|z_{t-1})$	14.534 (0.04)	30000	142.1737	17.5
VB-SIS1 $\Delta = 7$	14.483 (0.03)	30000	51.2624	24.2
VB-SIS2 $\Delta = 7$	14.437 (0.03)	30000	98.1525	19.7

Table 2.9: Simulation results for stochastic volatility model with $\Delta = 7$, $T = 50$, and varying sample size m

Proposal	Estimate (s.e.)	T	cv^2	Time (seconds)
$f(z_t\|z_{t-1})$	15.323 (1.42)	30	151.0883	0.7
VB-SIS1 $\Delta = 7$	13.897 (0.97)	30	48.0338	15.3
VB-SIS2 $\Delta = 7$	13.627 (1.23)	30	65.5262	3.6
$f(z_t\|z_{t-1})$	24.723 (2.42)	50	412.5422	2.2
VB-SIS1 $\Delta = 15$	26.373 (1.75)	50	73.2527	22.4
VB-SIS2 $\Delta = 15$	26.426 (1.98)	50	83.6236	8.4
$f(z_t\|z_{t-1})$	-24.523 (3.42)	100	1524.3532	15.3
VB-SIS1 $\Delta = 32$	-27.124 (2.52)	100	265.3262	32.5
VB-SIS2 $\Delta = 32$	-27.264 (2.97)	100	436.2363	20.3

Table 2.10: Simulation results for stochastic volatility model with $m = 5000$ and varying length of the sequence T

2.5.5 Dirichlet process

The last example is a Dirichlet process (DP) mixture model widely used in Bayesian inference. Dirichlet Process can be written as $G \sim \mathrm{DP}(\alpha, G_0)$, where G_0 is the base distribution of this stochastic process, and α is a positive scalar parameter. In addition, G and G_0 should have the same support, but G is a discrete distribution with countably infinite number of point masses. Given the previous $n-1$ observations, we generate the next one as follows:

$$X_n|X_1,\ldots,X_{n-1} = \begin{cases} X_i & \text{with probability } \frac{1}{n-1+\alpha} \quad (i = 1,\ldots,n-1), \\ \text{a new draw from } G_0 & \text{with probability } \frac{\alpha}{n-1+\alpha}. \end{cases}$$

Let K be the unique values among $\{X_1, \ldots, X_{n-1}\}$, denoted by $\{X_k^*\}_{k=1}^K$, and we can rewrite the sampling procedure as

$$X_n|X_1, \ldots, X_{n-1} = \begin{cases} X_k^* & \text{with probability } \frac{\text{num}_{n-1}(X_k^*)}{n-1+\alpha} \quad (k = 1, \ldots, K), \\ \text{a new draw from } G_0 & \text{with probability } \frac{\alpha}{n-1+\alpha}, \end{cases}$$

where $\text{num}_{n-1}(X_k^*)$ is the number of of X_k^* in the set $\{X_1, \ldots, X_{n-1}\}$. Then, the joint density function can be written as

$$\begin{aligned} P(X_1, \ldots, X_N) &= P(X_1)P(X_2|X_1)\cdots P(X_N|X_1, \ldots, X_{N-1}) \\ &= \frac{\alpha^k \prod_{k=1}^K (\text{num}_N(X_k^*) - 1)!}{\alpha(1+\alpha)\cdots(N-1+\alpha)} \prod_{k=1}^K G_0(X_K^*), \end{aligned}$$

which does not depend on the order of variables.

Dirichlet process can also be treated as a stick breaking process. We first draw $V_1, V_2, \ldots \sim \text{Beta}(1, \alpha)$, then generate $X_1^*, X_2^*, \ldots \sim G_0$. A multinomial distribution can be derived as

$$\pi_i(\mathbf{v}) = v_i \prod_{j=1}^{i-1}(1 - v_j).$$

The Dirichlet process G is a discrete distribution with $P(G = X_i^*) = \pi_i(\mathbf{v})$, and it can be written as $G = \sum_{i=1}^\infty \pi_i(\mathbf{v})\delta_{X_i^*}$, where δ_x is the Dirac measure at point x. In Dirichlet process mixture model, data come from a mixture of an infinite number of distributions. If we have N observed data points $\{x_i\}_{i=1}^N$, they will be generated from at most N different components. The following is the generating procedure of DP mixture model.

- $V_1, V_2, \ldots \sim \text{Beta}(1, \alpha)$
- $\pi_i(\mathbf{v}) = v_i \prod_{j=1}^{i-1}(1 - v_j)$
- $y_i \sim \text{Multinomial}(\pi)$
- $\eta_k \sim G_0$

- $X_i|y_i, \boldsymbol{\eta} \sim p(x_i|\eta_{y_i})$

Given the latent variable z_i, we assume the observation x_i follows a distribution from an exponential family with the likelihood function $p(x_i|\eta_{y_i})$.

Following Blei and Jordan (2006) and Hughes and Sudderth (2013), let $\mathbf{Z} = \{\mathbf{V}, \boldsymbol{\eta}, \mathbf{Y}\}$ be all latent variables and $\theta = \{\alpha\}$ be the hyper parameter. Since the number of different components is infinite, we introduce a truncated level T as an upper bound of the number of clusters, that is, mixture proportions $\pi_t(\mathbf{v}) = 0$ for $t > T$. Then we can factorize the posterior distribution and obtain the following variational decomposition:

$$q(\mathbf{v}, \boldsymbol{\eta}, \mathbf{y}) = \prod_{t=1}^{T-1} q_{1,t}(v_t) \prod_{t=1}^{T} q_{2,t}(\eta_t) \prod_{n=1}^{N} q_{3,n}(y_n),$$

where $q_{1,t}(v_t)$ are beta distributions, $q_{2,t}(\eta_t)$ are exponential family distributions, and $q_{3,n}(y_n)$ are multinomial distributions. We can use the coordinate ascent algorithm to solve the optimization problem. A general rule to choose the truncated level T is to be close to the theoretical value of the expected number of clusters, given N observations:

$$E\left[\text{number of clusters}|x_1, \ldots, x_N\right] = \sum_{i=1}^{N} \frac{\alpha}{\alpha + i - 1} = \alpha(\psi(\alpha + N) - \psi(\alpha)),$$

where $\psi(\cdot)$ is the digamma function.

We generated $N = 50$ observed data from DP mixture model, and implemented IS with different proposal distributions based on $m = 1,000$ samples. We considered different combinations of the hyper parameters (α, T). Since the number of parameters is large, we only reported the cv^2 and the average of the ratios of the standard errors of the parameter estimates from different methods.

T	α	cv^2 (naive proposal)	cv^2 (VB proposal)	s.e. ratio (naive/VB)
2	1	159.43	32.62	1.52
3	1	142.52	21.63	3.62
5	1	163.13	19.63	10.39
7	1	158.40	29.64	7.52
5	3	235.12	62.35	3.74
7	3	265.32	53.52	5.96
9	3	257.41	37.36	12.94
11	3	246.51	51.74	9.62

Table 2.11: Simulation results for Dirichlet process mixture models

From the results in Table 2.11, we can see that IS with variational distribution as proposal gives smaller cv^2 than IS with prior as the proposal. The average of the ratios of the standard errors is greater than 1 in all settings, which means using VB as the proposal usually gives smaller standard errors than using the naive proposal. This average ratio becomes larger when the truncated level T is close to the theoretical expectation of the number of clusters (4.49 for $\alpha = 1$ and 9.11 for $\alpha = 3$).

2.6 Discussion

In this section, we combine variational approximation and IS to improve the performance of both methods. Using variational approximation as the proposal distribution of IS can avoid the biased estimate and the lack of uncertainty quantification of the VB estimate. It also provides a way to design a good proposal for IS. We provide theoretical justification of the proposed methods, and numerical results also show that using variational approximation as the proposal can enhance the performance of IS and SIS.

Using VB as proposal for IS tends to be computationally more expensive than some

naive choice of the proposal. This is mainly due to the computational cost for finding the VB solution. Sometimes it might be worthwhile to stop the VB algorithm a little early to obtain a rough approximation and allow more time for IS to correct the bias. The tradeoff between VB-SIS1 and VB-SIS2 also illustrates this point.

2.7 Proofs

2.7.1 Proof of Lemma 2.1

Proof. We have $\lim_{n\to\infty} \beta_{2,n} = 1$ immediately from the definition of convergence in (2.4).

Now we prove $\lim_{n\to\infty} \beta_{1,n} = 1$. For $\forall\ \epsilon > 0$ and $\delta > 0$, define $I_1^{(n)} = \{x : \frac{p_n(x)}{q(x)} < 1-\epsilon\}$, $I_2^{(n)} = \{x : 1-\epsilon \le \frac{p_n(x)}{q(x)} < 1+\delta\}$, and $I_3^{(n)} = \{x : \frac{p_n(x)}{q(x)} \ge 1+\delta\}$. Since the Lebesgue measure of $I_1^{(n)}$ is 0 for large enough n, we have

$$\int_{I_1^{(n)}} \frac{p_n}{q}\, q\, dx = 0. \qquad \text{(for large enough } n\text{)}.$$

So

$$1 = \int_{I_1^{(n)}} \frac{p_n}{q}\, q\, dx + \int_{I_2^{(n)}} \frac{p_n}{q}\, q\, dx + \int_{I_3^{(n)}} \frac{p_n}{q}\, q\, dx = \int_{I_2^{(n)}} \frac{p_n}{q}\, q\, dx + \int_{I_3^{(n)}} \frac{p_n}{q}\, q\, dx. \quad \text{(for large enough } n\text{)}.$$

Suppose $\liminf_{n\to\infty} \int_{I_2^{(n)}} q\, dx = \theta \ge 0$, then

$$1 = \int_{I_2^{(n)}} \frac{p_n}{q}\, q\, dx + \int_{I_3^{(n)}} \frac{p_n}{q}\, q\, dx \ge (1-\epsilon) \int_{I_2^{(n)}} q\, dx + (1+\delta) \int_{I_3^{(n)}} q\, dx. \tag{2.10}$$

Take limit inferior on both sides of (2.10), we have

$$1 \ge (1-\epsilon)\theta + (1+\delta)(1-\theta) = 1 + (1-\theta)\delta - \theta\epsilon. \tag{2.11}$$

Therefore $\theta\epsilon \ge (1-\theta)\delta$ for any $\epsilon > 0$ and $\delta > 0$. Since $0 \le \theta \le 1$, we have $\theta = 1$. Thus we

have $\liminf_{n\to\infty} \int_{I_3^{(n)}} q\,dx = 0$ for any $\delta > 0$. From the definition of $\beta_{1,n}$, we have $\lim_{n\to\infty} \beta_{1,n} = 1$.

Since

$$D_f(p_n||q) = \int f\left(\frac{p_n}{q}\right) q\,dx \leq \sup_{\beta_{2,n}\leq\beta\leq\beta_{1,n}^{-1}} f(\beta),$$

let $n \to \infty$, we have $\lim_{n\to\infty} D_f(p_n||q) \leq f(1) = 0$. □

2.7.2 Proof of Theorem 2.2

Proof. From Lemma 2.1, we have

$$\lim_{n\to\infty} \beta_{1,n} = \lim_{n\to\infty} \beta_{2,n} = 1.$$

By L'Hospital's rule, we have $\lim_{t\to 1} \kappa(t) = 1$, where $\kappa(t)$ is defined in (2.5). Therefore, take limit on the both sides of (2.6) and (2.7), we have

$$\lim_{n\to\infty} \frac{KL(p_n||q)}{KL(q||p_n)} = 1\ , \quad \lim_{n\to\infty} \frac{KL(p_n||q)}{\chi^2(p_n||q)} = \frac{1}{2}.$$

□

Chapter 3

Statistical Inference on Social Influence

3.1 Introduction

Social network analysis plays an importance role in many fields, including sociology, psychology, biology etc. Many new methods have been developed in recent years to analyze the network data, and statistical technique sometimes is used when implementing the analysis procedure. Mathematically, we will use a graph to denote the whole network, and each person will be presented by a vertex or node in the graph. Also, the friendship can be represented by the edges between each pair of the nodes. This abstract notation provides us an intuitive way to describe the network, and is also convenient to build the statistical models and estimate the quantities of interest.

However, if people only concern about the community structure in the network, and the peer effects and the covariates of each person will be ignored. For example, medical research is centered on individual health outcomes, such as people smoke or not. The researchers should also care about if the behavior of one person has any influences on another one, or how long can the effects last, and also study the spread of features across network ties.

Here, we are interested in finding if the individuals traits can spread from one person to another, which is usually known as social influence or social diffusion. There has been some research about the spread of people's behavior within a social network in social science (Valente, 1996; Kempe et al., 2005; Centola, 2010). In addition, Sun and Tang (2011) provided a summary of statistical measures and models in social influence analysis. Researchers examined the spread of various features including smoking (Christakis and Fowler, 2008;

Miething et al., 2016), alcohol (Rosenquist et al., 2010), tastes (Lewis et al., 2012), happiness (Fowler and Christakis, 2008) and obesity (Christakis and Fowler, 2007). In addition, La Fond and Neville (2010) proposed a randomization test for temporal data, and measured the gain in correlation to determine whether the gain is due to influence. Christakis and Fowler (2013) developed a permutation test to identify causal effects using Framingham Heart Study (FHS) data. O'Malley (2013) provided a method to account for the confounding effect in the analyses of peer effects. Sewell (2018) proposed a hierarchical model to connect individuals susceptibility with individuals characteristics in egocentric network data. Kempe et al. (2003) and Goyal et al. (2011) also proposed some models for social influence maximization problem, which aims to find a sets of users in a network and maximize the expected spread of influence.

This chapter focuses on exploring the degrees of influence in an observed network. We build a multivariate Bernoulli model to specify the correlation structure of the people's behaviors in the whole network. In order to detect the true degrees of influence, we propose a sequential hypothesis testing procedure and overcome the issue of nuisance parameters by introducing double bootstrap (Beran, 1988). In addition, we show that under certain conditions, the power of our proposed hypothesis testing goes to one when the network is large. We also do some simulation studies and real data analyses to illustrate the performance of our proposed method.

The chapter is organized as follows. We introduce our proposed multivariate Bernoulli model for social influence in Section 3.2, and the general hypothesis testing procedure in Section 3.3. Then, we provide theoretical justification in Section 3.4, and give some simulation results of the new method in Section 3.5. We also implement our model on two real network datasets in Section 3.6. Section 3.7 concludes the chapter with a discussion.

3.2 Multivariate Bernoulli Model

Social influence indicates a process that the behaviors or opinions of an individual are affected by others in a network. In this chapter, we are exploring the degrees of influence in a network, which describes how long the influence can pass through individuals. We consider the static network, and the structure of the network we are going to analyze can be represented as an $n \times n$ adjacency matrix A, where n is the size of the network.

Suppose $A = (a_{ij})_{n\times n}$, and a_{ij} represents the relationship between two individuals i and j, and A is not necessarily symmetric in a directed network. Here, $a_{ij} = 1$ means individual j has some influences on individual i, and in the graph representation, there is an arrow directing from i to j. The individual we are focusing on is called ego, and all the other nodes that connect with ego through a path are called alters. We have the following representation

$$\text{ego} \ \rightarrow \ \text{alter}_1 \ \rightarrow \ \text{alter}_2 \ \rightarrow \ \cdots,$$

where alter_1 is the first-degree alter of the ego, and alter_2 is the second-degree alter of the ego, and it should not be connected to the ego directly, that is, $A_{\text{ego,alter}_1} = 1$, $A_{\text{alter}_1,\text{alter}_2} = 1$ and $A_{\text{ego,alter}_2} = 0$.

For each node j, there is a binary random variable Y_j representing its current status (such as smoking). For each pair of nodes i and j, individual j has some influence on i is equivalent to the following inequality

$$P(Y_i = 1|Y_j = 1) > P(Y_i = 1|Y_j = 0).$$

Since both Y_i and Y_j are binary variables, we have

$$Cov(Y_i, Y_j) = E[Y_iY_i] - E[Y_i]E[Y_j] = P(Y_i = 1, Y_j = 1) - P(Y_i = 1)P(Y_j = 1).$$

Then,

$$
\begin{aligned}
& P(Y_i = 1 | Y_j = 1) > P(Y_i = 1 | Y_j = 0) \\
\Leftrightarrow \quad & \frac{P(Y_i = 1, Y_j = 1)}{P(Y_j = 1)} > \frac{P(Y_i = 1, Y_j = 0)}{P(Y_j = 0)} \\
\Leftrightarrow \quad & \frac{P(Y_i = 1, Y_j = 1)}{P(Y_j = 1)} > \frac{P(Y_i = 1, Y_j = 1) + P(Y_i = 1, Y_j = 0)}{P(Y_j = 1) + P(Y_j = 0)} \\
\Leftrightarrow \quad & \frac{P(Y_i = 1, Y_j = 1)}{P(Y_j = 1)} > P(Y_i = 1) \\
\Leftrightarrow \quad & P(Y_i = 1, Y_j = 1) > P(Y_j = 1) P(Y_i = 1) \\
\Leftrightarrow \quad & Cov(Y_i, Y_j) > 0 \\
\Leftrightarrow \quad & Corr(Y_i, Y_j) > 0.
\end{aligned}
$$

Thus, individual j has influence on i is equivalent to the correlation between Y_i and Y_j is positive.

For a given degrees of influence d, we propose the following multivariate Bernoulli model to illustrate the joint distribution of $(Y_1, \ldots, Y_n)$. For a random vector $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$, we say $\mathbf{Y}$ follows a multivariate Bernoulli distribution (Dai et al., 2013), if Y_i can only take values either 0 or 1, and the marginal distribution of Y_i is a Bernoulli distribution for $i = 1, \ldots, n$. In order to implement the randomization test, we need to know how to generate samples from multivariate Bernoulli distribution with particular mean vector and the correlation matrix.

3.2.1 Sample from multivariate Bernoulli distribution

Leisch et al. (1998) proposed the following method to sample from the multivariate Bernoulli distribution given the mean vector $\mathbf{p}$ and the correlation matrix $\mathbf{R}$. We can also use the R package `bindata` (Leisch et al., 2012) to sample from it directly. Suppose $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$ follows a multivariate Bernoulli distribution where $\mathbf{p} = (p_1, \ldots, p_n)^T$ and $\mathbf{R} = (r_{ij})$. Then the marginal distribution of $\mathbf{Y}$ is $Y_i \sim \text{Bernoulli}(p_i)$ $(i = 1, \ldots, n)$, and the

correlation between each pair of components is $Corr(Y_i, Y_j) = r_{ij}$. Thus, the following equation holds:

$$\begin{aligned} P(Y_i = 1, Y_j = 1) = E[Y_i Y_j] &= Cov(Y_i, Y_j) + E[Y_i]E[Y_j] \\ &= \sqrt{p_i(1-p_i)p_j(1-p_j)}\, r_{ij} + p_i p_j = \tau_{ij}. \end{aligned}$$

We can generate a sample $\mathbf{Z} = (Z_1, \ldots, Z_n)^T$ from the multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \tilde{\boldsymbol{\Sigma}})$ with mean $\boldsymbol{\mu}$ and covariance matrix $\tilde{\boldsymbol{\Sigma}} = (\tilde{\sigma}_{ij})$ where $\tilde{\sigma}_{ii} = 1$ $(i = 1, \ldots, n)$. Let $Y_i = 1_{\{Z_i \geq 0\}}$, where 1_A is the indicator function. Also, we set $\mu_i = \Phi^{-1}(p_i)$, where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution, then,

$$P(Y_i = 1) = P(Z_i \geq 0) = P(Z_i - \mu_i \geq -\mu_i) = 1 - \Phi(-\mu_i) = p_i.$$

The relationship between $\tilde{\sigma}_{ij}$ and τ_{ij} is shown as follows:

$$\tau_{ij} = P(Y_i = 1, Y_j = 1) = P(Z_i \geq 0, Z_j \geq 0) = \int_{-\mu_i}^{\infty} \int_{-\mu_j}^{\infty} \phi(x, y; \tilde{\sigma}_{ij}) dx dy,$$

where

$$\phi(x, y; \rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{x^2 - 2\rho xy + y^2}{2(1-\rho^2)}\right).$$

We can use the bisection method to obtain $\tilde{\sigma}_{ij}$ given τ_{ij}. For the rest of the chapter, We use the following notation to denote the multivariate Bernoulli distribution generated by the above steps:

$$\mathbf{Y} \sim \text{multiBern}(\mathbf{p}, \mathbf{R}),$$

where $\mathbf{p} = E[\mathbf{Y}]$, and $\mathbf{R} = (r_{ij})_{n \times n}$ is the correlation matrix of $\mathbf{Y}$ with $r_{ij} = Corr(Y_i, Y_j)$. For different degrees of influence in the network, we further assume the popularity of a behavior for all individuals in the network is p, and propose the following corresponding correlation

matrix structures for $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$.

3.2.2 Degrees of influence is 0

If the degrees of influence is zero, then the behavior of each individual will not be affected by other people, so $Y_1, \ldots, Y_n$ will be independent to each other. Then, we have $Y_1, \ldots, Y_n$ are independent and identically distributed random variables with the following joint distribution:

$$\mathbf{Y} \sim \text{multiBern}(\mathbf{p}, \mathbf{R}),$$

where $\mathbf{p} = (p, p, \ldots, p)^T$ and $\mathbf{R} = \mathbf{I}_n$, where $\mathbf{I}_n$ is the identical matrix.

3.2.3 Degrees of influence is 1

If the degrees of influence is 1, then people's behavior will be influenced by their friends, and the influence phenomenon happens only when two people are directly connected in the network. We have $Y_1, \ldots, Y_n \sim \text{Bernoulli}(p)$, but they are not independent. We introduce a positive parameter q_1 to quantify the correlation between a person and his or her neighbor, then the correlation matrix $\mathbf{R}$ will no longer be diagonal. The structure of the correlation matrix $\mathbf{R}$ is shown as follows.

- If $a_{ij} = 1$, $a_{ji} = 1$, then $r_{ij} = Corr(Y_i, Y_j) = 2q_1 > 0$.
- If $a_{ij} = 1$, $a_{ji} = 0$, then $r_{ij} = Corr(Y_i, Y_j) = q_1 > 0$.
- If $a_{ij} = 0$, $a_{ji} = 1$, then $r_{ij} = Corr(Y_i, Y_j) = q_1 > 0$.
- If $a_{ij} = 0$, $a_{ji} = 0$, then $r_{ij} = Corr(Y_i, Y_j) = 0$.

3.2.4 Degrees of influence is greater than 1

We can generalize the multivariate Bernoulli model to a network with arbitrary degrees of influence. Suppose the true degrees of influence is d^*. Let d_{ij} be the length of shortest

path from i to j, then we have $d_{ij} \geq 1$ for $i, j = 1, 2, \ldots, n$. Since we consider a directed network, $d_{ij} = d_{ji}$ is not always true. If two node i and j are not connected in the network, then $d_{ij} = \infty$.

We propose the following correlation structure for the random vector $\mathbf{Y}$.

- If $1 \leq d_{ij} \leq d^*$ and $1 \leq d_{ji} \leq d^*$, then $r_{ij} = Corr(Y_i, Y_j) = q_{d_{ij}} + q_{d_{ji}} > 0$.
- If $1 \leq d_{ij} \leq d^*$ and $d_{ji} > d^*$, then $r_{ij} = Corr(Y_i, Y_j) = q_{d_{ij}} > 0$.
- If $d_{ij} > d^*$ and $1 \leq d_{ji} \leq d^*$, then $r_{ij} = Corr(Y_i, Y_j) = q_{d_{ji}} > 0$.
- If $d_{ij} > d^*$ and $d_{ji} > d^*$, then $r_{ij} = Corr(Y_i, Y_j) = 0$.

Another assumption for the correlation parameters is $q_1 > 0, q_2 > 0, \ldots, q_{d^*} > 0$, which means the model with smaller degrees of influence is nested in the model with larger degrees of influence. In order to guarantee that the correlation matrix $\mathbf{R}_{n \times n}$ is semi-positive definite, $q_1, \ldots, q_{d^*}$ should be selected specifically. Also, the model we proposed in Section 3.2.3 is just a special case for $d^* = 1$.

3.3 Hypothesis testing

3.3.1 Different ways to determine the degree

In order to determine the degrees of influence in a given network, there are two hypothesis testing based methods. The first way is testing H_0 : degree $= 0$ vs. H_1 : degree $= d$ at each time. The second way is testing H_0 : degree $= d - 1$ vs. H_1 : degree $= d$ at each time. For both procedures, we will start from $d = 1$, and increase d by 1 if rejecting the null hypothesis. Each procedure will be stopped until we do not reject the null, and then we can claim that $d - 1$ is the degrees of influence.

However, $H_0 : d = 0$ is not always the null hypothesis of primary interest if we concern about the higher degrees of influence. A claim of the degrees of influence is 3 will be

more convincing if we consider using next simplest case $d = 2$ as the model under the null hypothesis.

Actually, using sequential hypothesis testing to determine the degrees of influence in network is similar to variable selection in linear regression. The second procedure adds one new predictor to the model at each time, and implements the goodness of fit test to compare it with the model without the new variable. So, it is just like the forward variable selection procedure. However, the first procedure just consider the intercept only model as the null model for each time.

Here, we consider using the multivariate Bernoulli distribution to propose a new sequential hypothesis testing procedure to detect the degrees of influence. More discussions about the comparison of these two procedures is in Section 3.5.1.

Christakis-Fowler method

If we consider the following hypothesis test: H_0 : degree $= 0$ vs. H_1 : degree $= d$. Christakis and Fowler (2013) proposed a permutation test. We can obtain the following contingency table by counting the frequencies for all (ego, alter$_d$) pairs.

	$Y_{\text{alter}_d} = 1$	$Y_{\text{alter}_d} = 0$
$Y_{\text{ego}} = 1$	a_1	b_1
$Y_{\text{ego}} = 0$	c_1	d_1

Table 3.1: Contingency table for hypothesis testing H_0 : degree $= 0$ vs. H_1 : degree $= d$

The proposed test statistic is

$$T = \frac{a_1}{a_1 + c_1} - \frac{b_1}{b_1 + d_1} = \hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_d} = 1) - \hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_d} = 0).$$

We will use Christakis-Fowler to denote this method in the rest of our chapter.

Proposed method

Christakis and Fowler (2013) just considered the hypothesis testing when the degree under H_0 is 0. Here, we propose the following hypothesis test, H_0 : degree $= d-1$ vs. H_1 : degree $= d$. For the new hypothesis testing, $Y_1, \ldots, Y_n$ are not independent under the null when $d > 1$, so we need to eliminate the effect for alter_{d-1} when designing the test statistic. We can obtain the following two contingency tables by counting the frequencies for all cases containing egos, alter_{d-1}'s and alter_d's.

	$Y_{\text{alter}_d} = 1$	$Y_{\text{alter}_d} = 0$
$Y_{\text{ego}} = 1$	a_2	b_2
$Y_{\text{ego}} = 0$	c_2	d_2

Table 3.2: Contingency table for hypothesis testing H_0 : degree $= d-1$ vs. H_1 : degree $= d$ when $Y_{\text{alter}_{d-1}} = 1$

	$Y_{\text{alter}_d} = 1$	$Y_{\text{alter}_d} = 0$
$Y_{\text{ego}} = 1$	e_2	f_2
$Y_{\text{ego}} = 0$	g_2	h_2

Table 3.3: Contingency table for hypothesis testing H_0 : degree $= d-1$ vs. H_1 : degree $= d$ when $Y_{\text{alter}_{d-1}} = 0$

The test statistic is:

$$\begin{aligned} T &= \frac{a_2}{a_2+c_2} - \frac{b_2}{b_2+d_2} + \frac{e_2}{e_2+g_2} - \frac{f_2}{f_2+h_2} \\ &= \hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_{d-1}} = 1, Y_{\text{alter}_d} = 1) - \hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_{d-1}} = 1, Y_{\text{alter}_d} = 0) \\ &+ \hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_{d-1}} = 0, Y_{\text{alter}_d} = 1) - \hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_{d-1}} = 0, Y_{\text{alter}_d} = 0). \end{aligned}$$

Our proposed test statistics is valid since only when the alternative hypothesis is true, T tends to be larger. However, it is very hard to know the distribution of T under H_0, so

we will use the randomization test (Dwass, 1957) and bootstrap hypothesis testing (MacKinnon, 2009) to obtain the p value. This is a non-parameter method which can estimate the distribution of the test statistic under the null.

For the Christakis-Fowler method, $Y_1, \ldots, Y_n$ are independent under the null hypothesis, we can just randomly shuffle all components for the observation $\mathbf{Y}_0$ to obtain new samples $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(B)}$. For our proposed method, if the null hypothesis is degree $= d - 1 \geq 1$, we can consider randomly generate new samples $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(B)}$ under H_0 from the multivariate Bernoulli model described in Section 3.2.

Toy example

We use the following toy example to illustrate how to calculate the test statistic. We generated a network with $n = 10$ nodes, and assigned the corresponding smoking status for each node. The network is shown as follows:

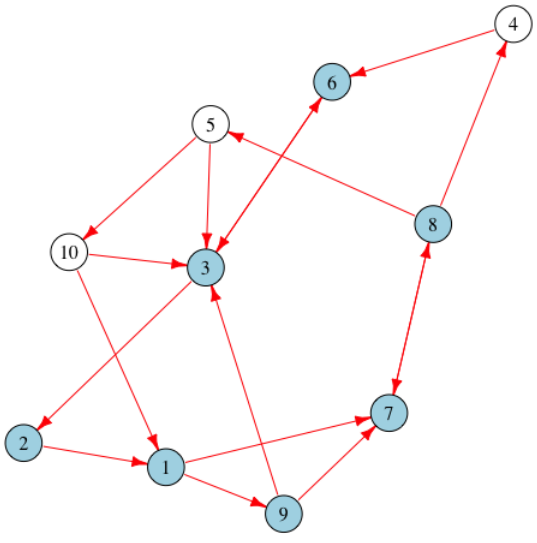


Figure 3.1: Network structure for the toy example

In the network, we use white nodes to denote the three people who smoke, and the rest of the seven blue nodes indicate non-smokers. In order to calculate the test statistic, we consider all pairs of $(\text{ego}, \text{alter}_1)$ and their smoking status in this network, and the results are shown as follows.

	$(\text{ego}, \text{alter}_1)$ pairs
$Y_{\text{ego}} = 1, Y_{\text{alter}_1} = 1$	(5, 10)
$Y_{\text{ego}} = 1, Y_{\text{alter}_1} = 0$	(4, 6), (5, 3), (10, 3), (10, 1)
$Y_{\text{ego}} = 0, Y_{\text{alter}_1} = 1$	(8, 4), (8, 5)
$Y_{\text{ego}} = 0, Y_{\text{alter}_1} = 0$	(1, 9), (1, 7), (2, 1), (3, 2), (3, 6), (6, 3), (7, 8), (8, 7), (9, 1), (9, 3)

Table 3.4: All $(\text{ego}, \text{alter}_1)$ pairs for the toy example

From Table 3.4, we can see that there is only one pair of $(Y_{\text{ego}}, Y_{\text{alter}_1}) = (1, 1)$ in the network. For other cases, there are two pairs of $(Y_{\text{ego}}, Y_{\text{alter}_1}) = (0, 1)$, four pairs of $(Y_{\text{ego}}, Y_{\text{alter}_1}) = (1, 0)$ and ten pairs of $(Y_{\text{ego}}, Y_{\text{alter}_1}) = (0, 0)$. Then, we can obtain the following contingency table.

	$Y_{\text{alter}_1} = 1$	$Y_{\text{alter}_1} = 0$
$Y_{\text{ego}} = 1$	1	4
$Y_{\text{ego}} = 0$	2	10

Table 3.5: Contingency table for hypothesis testing H_0 : degree $= 0$ vs. H_1 : degree $= 1$ for the toy example

The test statistics for the hypothesis testing H_0 : degree $= 0$ vs. H_1 : degree $= 1$ is

$$T = \frac{1}{1+2} - \frac{4}{4+10} = -0.0667.$$

3.3.2 Nuisance parameters

For the hypothesis testing H_0 : degree = 1 vs. H_1 : degree = 2, the parameter that we are interested in is q_2, and the nuisance parameter is q_1. It is easy to estimate nuisance parameter q_1 by using sample correlations between all $(\text{ego}, \text{alter}_1)$ pairs.

We constructed a network from the Erdős-Rényi model (n, p_e) with $n = 100$ nodes, and the edge probability is $p_e = 0.2$. For each node, the marginal distribution of Y_i is Bernoulli(p), where $p = 0.3$. We generated $m = 1000$ observed data $\mathbf{Y} = (Y_1, \cdots, Y_n)$ from the multivariate Bernoulli model when $d = 0$, and the mean of estimated q_1 is -0.0097. Then, we also generated $m = 1000$ observed data $\mathbf{Y}$ from the model when $d = 1$, and the true value of the correlation is $q_1 = 0.03$. If we still used the sample correlation to estimate q_1, the results show that the mean of estimated q_1 is 0.0284. The p-value for the two sample t-test between the two estimated vectors corr_0 and corr_1 is 6×10^{-8}, which means we can distinguish the estimated values of parameters from different models.

The following plot shows the kernel density curves of the estimated q_1 under models with different degrees of influence.

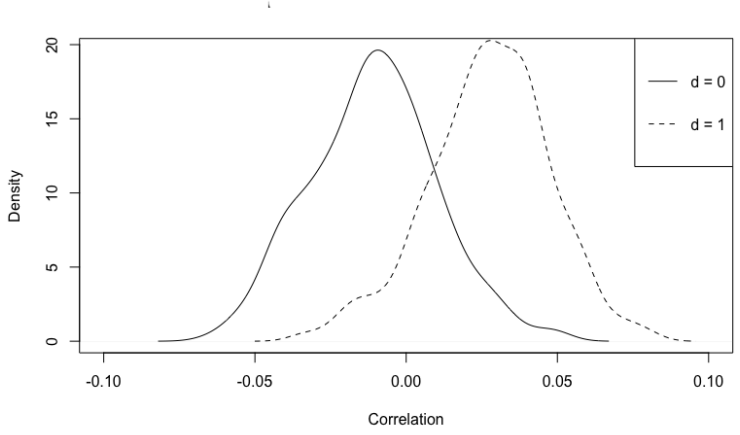


Figure 3.2: Kernel density plot for estimated q_1

There are a lot of methods which can deal with the hypothesis testing with nuisance parameters θ. The most commonly used one is the conditional method, which requires

the existence of a statistic T that is sufficient for the nuisance parameter under the null hypothesis. Dufour (2006) also proposed a method to maximize the p value with respect to the nuisance parameters θ, and used the following quantity $p_{\text{sup}} = \sup_\theta p(\theta)$ as the p-value. Berger and Boos (1994) used the confidence interval of the nuisance parameter under H_0 to build a confidence set, and showed the validity of the proposed p-value. In addition, there are some other methods including Bayesian p-value (Robins et al., 2000; Bayarri and Berger, 2000) and Generalized p-value (Tsui and Weerahandi, 1989).

However, our proposed method is based on randomization test, and all the above methods are not easy to be applied in our case. When the hypothesis we are testing involves estimated parameters, simple bootstrap (Efron, 1992) will introduce uncertainty for nuisance parameters. In order to correct this, we consider the following double bootstrap (Beran, 1988) procedure to improve the accuracy of the estimations of p-values. The following is the general procedure of double bootstrap:

- Calculate the test statistic T_0 and the estimation of the nuisance parameters $\hat{\theta}$ from observed data.
- Generate B_1 bootstrap samples from H_0 with $\hat{\theta}$, and use each of them to compute a bootstrap test statistics T_j^* $(j = 1, \ldots, B_1)$.
- Calculate the first-level bootstrap p-value

$$\hat{p}^*(T_0) = \frac{1}{B_1} \sum_{j=1}^{B_1} 1_{\{T_j^* > T_0\}}.$$

- For each of the B_1 first-level bootstrap samples, re-estimate the nuisance parameters and obtain $\tilde{\theta}_j$. Generate B_2 second-level bootstrap samples from H_0 with $\tilde{\theta}_j$, and use them to compute the second-level test statistics T_{jl}^{**} $(l = 1, \ldots, B_2)$.
- For each of the B_1 first-level bootstrap samples, compute the second-level bootstrap

p-values

$$\hat{p}_j^{**} = \frac{1}{B_2} \sum_{l=1}^{B_2} 1_{\{T_{jl}^{**} > T_j^*\}}.$$

- Compute the double-bootstrap p-value:

$$\hat{p}^{**}(T_0) = \frac{1}{B_1} \sum_{j=1}^{B_1} 1_{\{\hat{p}_j^{**} < \hat{p}^*(T_0)\}}.$$

The inner bootstrap is used to calculate the distribution of nominal bootstrap p-values. For this procedure, we need to draw second-level bootstrap samples from the bootstrap re-estimated $\tilde{\theta}$, not from estimation $\hat{\theta}$ based on the observed data. One constraint for double bootstrap method is that the estimated parameter $\hat{\theta}$ has to be a consistent estimator of the nuisance parameter θ. In our proposed method, we use sample correlation as the estimator for those correlation parameters q_i, which can guarantee the consistency.

3.4 Theory

In order to verify the correctness of our proposed method, we show some theoretical properties of the sequential hypothesis testing procedure. In this section we will first show a property of the Erdős-Rényi model (Erdős and Rényi, 1960), and use it to show how the power of hypothesis testing changes for different networks.

We consider the Erdős-Rényi model $ER(n, p_e)$, where n is the size of the network and p_e is the edge probability. For each pair of nodes $i \neq j$, we have $P(a_{ij} = 1) = p_e$ and $P(a_{ij} = 0) = 1 - p_e$. From our multivariate Bernoulli model, we are interested in the starting and the ending nodes for paths with some particular length k. Let λ be the expected degree for all nodes in the $ER(n, p_e)$, and we further assume how the edge probability decreases when the size of the network goes to infinity, then we have the following lemma about the number of length-k paths.

Lemma 3.1. *If a random graph is generated from $ER(n, p_e)$ with $p_e = O(\lambda/n)$, then the number of length-k paths in the graph goes to infinity in probability as $n \to \infty$ for any fixed positive integer k.*

The proof of Lemma 3.1 is in Section 3.8.2. We know that if $p_e = O(\lambda/n)$, the number of length-k paths goes to infinity in probability. From the lemma above, we have the following theorem which indicates the performance of the powers of the hypothesis testing at each step.

Theorem 3.2. *Suppose the network is generated by $ER(n, p_e)$ with $p_e = O(\lambda/n)$, and we fix all correlation parameters q_i $(i = 1, 2, \ldots)$ and the marginal probability p in the multivariate Bernoulli model. Suppose d^* is the true degrees of influence in a network with $q_1 > 0, \ldots, q_{d^*} > 0$. Let T be the proposed test statistic for the hypothesis testing $H_0 : degree = d - 1$ v.s. $H_1 : degree = d$, we have following results*

(i) $\lim_{n\to\infty} P(|T| \geq \epsilon | degree = d-1) = 0$ *for all* $\epsilon > 0$ *and all* $1 \leq d \leq d^*$, *which indicates that* $T \to 0$ *in probability under* H_0.

(ii) $P(T > c^* | degree = d) \to 1$ *as* $n \to \infty$ *for all* $1 \leq d \leq d^*$, *where* c^* *is the critical value of the hypothesis testing.*

The proof of Theorem 3.2 is in Section 3.8.3. From the above theorem, we know that the power of hypothesis testing goes to 1 when $n \to \infty$ at each step. So, we can always figure out the difference between the null and the alternative hypothesis as along as the network is large enough. It also indicates that our proposed method can detect the true degrees of influence with large probability.

3.5 Simulation results

3.5.1 Comparing the recovery rate

We generated network from the Erdős-Rényi model (n, p_e), where n is the size of the network and p_e is the edge probability. We set the marginal probability of Y_i (popularity) to be $p = 0.3$. For the rest of the chapter, we set the pre-specified level for hypothesis testing to be $\alpha = 0.05$. In order to compare our proposed method with the Christakis-Fowler method, we generated data with true degree $d^* = 2$ for $m = 30$ times, and obtained the following results:

$(n,\ q_1,\ q_2)$	Christakis-Fowler			Proposed method		
	$< d^*$	$= d^*$	$> d^*$	$< d^*$	$= d^*$	$> d^*$
(10, 0.3, 0.3)	2	15	13	3	21	6
(20, 0.2, 0.2)	3	17	10	2	24	4
(50, 0.15, 0.15)	1	14	15	1	22	7
(100, 0.1, 0.1)	2	14	14	2	22	6
(200, 0.05, 0.05)	4	13	13	3	22	5
(500, 0.015, 0.015)	2	18	10	2	24	4

Table 3.6: Results for different ways to detect the degrees of influence

From Table 3.6, we can see that our proposed procedure will be more likely to detect the true social influence comparing with the Christakis-Fowler method. We will choose it as the sequential hypothesis testing procedure when recovering the degree in the simulation study and real data analysis.

3.5.2 Erdős-Rényi model simulations

We generated network from Erdős-Rényi model (n, p_e), and set the marginal probability of Y_i (popularity) to be $p = 0.3$, and generated $m = 1000$ networks for each simulation. In

general, for hypothesis testing $H_0 : d = d_0$ v.s. $H_1 : d = d_1$, we generated bootstrap samples from H_0, and set $B_1 = 100$ and $B_2 = 200$ to be the parameters of double bootstrap to obtain p values. If the observed data is generated from $d = d_0$, we can estimate the level of the corresponding hypothesis testing by calculating the proportion of rejecting H_0 among the 1000 trials. On the other hand, if the observed data is generated from $d = d_1$, we can obtain the power in a similar way.

Fix the edge probability p_e

We first fix the edge probability p_e to be 0.1, and consider some relatively small networks. For each hypothesis test, we generate the observed data $\mathbf{Y}_0$ from either the null or the alternative to calculate the corresponding levels and powers of the hypothesis testing.

1. We first consider the simplest case. For the hypothesis testing $H_0 :$ degree $= 0$ vs. $H_1 :$ degree $= 1$, in addition to the randomization test, we can also use chi-squared test for 2-by-2 contingency table to determine whether the null hypothesis is true. Here, we obtain the following results.

$(n,\ q_1)$	Level (randomization test)	Level (chi-squared test)
(10, 0.1)	0.053	0.036
(10, 0.3)	0.064	0.038
(20, 0.1)	0.058	0.043
(20, 0.2)	0.051	0.031
(50, 0.1)	0.046	0.039
(50, 0.15)	0.054	0.045
(100, 0.1)	0.052	0.044

Table 3.7: Levels for hypothesis testing $H_0 :$ degree $= 0$ vs. $H_1 :$ degree $= 1$

$(n,\ q_1)$	Power (randomization test)	Power (chi-squared test)
(10, 0.1)	0.186	0.175
(10, 0.3)	0.275	0.223
(20, 0.1)	0.226	0.169
(20, 0.2)	0.302	0.234
(50, 0.1)	0.267	0.217
(50, 0.15)	0.287	0.233
(100, 0.1)	0.457	0.269

Table 3.8: Powers for hypothesis testing H_0 : degree $= 0$ vs. H_1 : degree $= 1$ (with $\alpha = 0.05$)

From Table 3.7, we can see that our randomization test can always acquire the correct pre-specified level $\alpha = 0.05$, but the chi-squared test are conservative sometimes and could not obtain the correct levels. From Table 3.8, our randomization test are always more powerful than the chi-squared test for given values of n and q_1. And when the network size n increases, the powers for our proposed method are also increasing. For two networks with same size n, it is easier to distinguish the null and the alternative for the one with larger value of q_1

2. For the hypothesis testing H_0 : degree $= 1$ vs. H_1 : degree $= 2$. In this case, the degrees of influence under the null is greater than zero, so we need to use double bootstrap method to obtain p values at each time. If the observed data $\mathbf{Y}_0$ is generated from H_0, no matter how we set the other parameters, the distribution for p-values is always close to a uniform [0,1]. The following figure is one of the histograms of p-values.

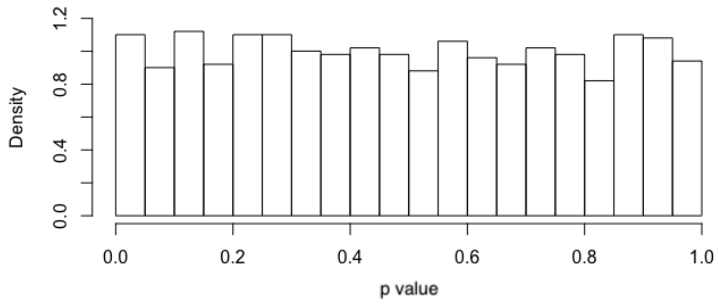


Figure 3.3: Histogram of p-value under H_0

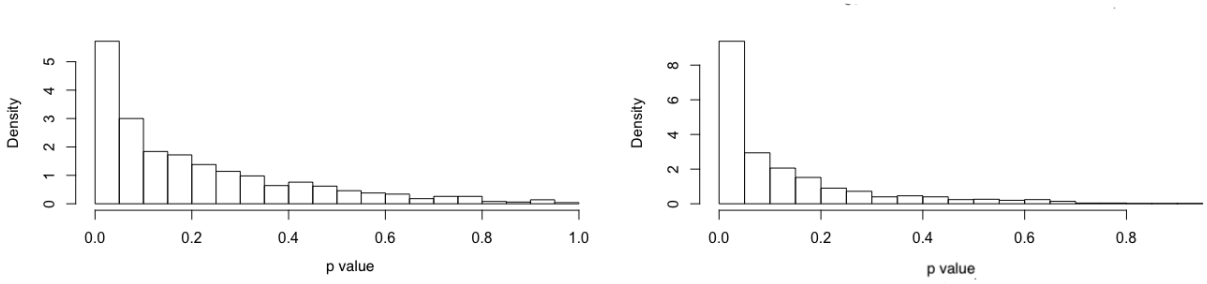
The following table shows the levels for hypothesis testing under different parameter settings.

$(n,\ q_1,\ q_2)$	Level (randomization test)
(10, 0.1, 0.1)	0.058
(10, 0.3, 0.3)	0.053
(20, 0.1, 0.1)	0.055
(20, 0.2, 0.2)	0.048
(50, 0.1, 0.1)	0.041
(50, 0.15, 0.15)	0.050
(100, 0.1, 0.1)	0.051

Table 3.9: Levels for hypothesis testing H_0 : degree $= 1$ vs. H_1 : degree $= 2$

From Table 3.9, we can see that our proposed method can still preserve the level of the hypothesis testing after introducing the double bootstrap, and the type I error is close to the pre-specified level $\alpha = 0.05$.

If the observed data $\mathbf{Y}_0$ is generated from H_1, then the p-value tends to be small and the distribution of it will not be uniform $[0, 1]$. Here, we show the two following histograms of p-values for different parameter settings:



(a) (b)

Figure 3.4: (a): Histogram of p-value under H_1 with $n = 50$, $q_1 = q_2 = 0.15$. (b): Histogram of p-value under H_1 with $n = 100$, $q_1 = q_2 = 0.1$

The following tables show the power for hypothesis test under different parameter settings:

(n, q_1, q_2)	Power (randomization test)	(n, q_1, q_2)	Power (randomization test)
(10, 0.1, 0.1)	0.236	(10, 0.1, 0.05)	0.215
(10, 0.3, 0.3)	0.322	(10, 0.3, 0.15)	0.315
(20, 0.1, 0.1)	0.266	(20, 0.1, 0.05)	0.273
(20, 0.2, 0.2)	0.315	(20, 0.2, 0.1)	0.296
(50, 0.1, 0.1)	0.389	(50, 0.1, 0.05)	0.352
(50, 0.15, 0.15)	0.456	(50, 0.15, 0.075)	0.416
(100, 0.1, 0.1)	0.587	(100, 0.1, 0.05)	0.569

Table 3.10: Powers for hypothesis testing H_0 : degree $= 1$ vs. H_1 : degree $= 2$ (with $\alpha = 0.05$)

The two histograms in Figure 3.4 are right skewed, which mean the p-values tend to be small under the alternative hypothesis. For most simulations, p-values we obtained are smaller than the pre-specified level α. From Table 3.10, we can see that if we fix the network size n and edge probability p_e, then increasing q_i $(i = 1, 2, \ldots)$ will make the hypothesis test more powerful. In addition, our proposed hypothesis testing is also more powerful when the network size n is larger.

3. For the test H_0 : degree $= 2$ vs. H_1 : degree $= 3$, similar to the previous case, we can generate the observed data $\mathbf{Y}_0$ from the model with degree 2 or 3. Then, we can obtain the following results for both the levels and the powers of the tests:

(n, q_1, q_2, q_3)	Level (randomization test)
(10, 0.1, 0.1, 0.1)	0.053
(10, 0.3, 0.3, 0.3)	0.057
(20, 0.1, 0.1, 0.1)	0.047
(20, 0.2, 0.2, 0.2)	0.051
(50, 0.1, 0.1, 0.1)	0.059
(50, 0.15, 0.15, 0.15)	0.052
(100, 0.1, 0.1, 0.1)	0.047

Table 3.11: Levels hypothesis testing H_0 : degree $= 2$ vs. H_1 : degree $= 3$

(n, q_1, q_2, q_3)	Power (randomization)	(n, q_1, q_2, q_3)	Power (randomization)
(10, 0.1, 0.1, 0.1)	0.296	(10, 0.1, 0.05, 0.025)	0.211
(10, 0.3, 0.3, 0.3)	0.363	(10, 0.3, 0.15, 0.075)	0.326
(20, 0.1, 0.1, 0.1)	0.389	(20, 0.1, 0.05, 0.025)	0.346
(20, 0.2, 0.2, 0.2)	0.402	(20, 0.2, 0.1, 0.05)	0.349
(50, 0.1, 0.1, 0.1)	0.491	(50, 0.1, 0.05, 0.025)	0.416
(50, 0.15, 0.15, 0.15)	0.587	(50, 0.15, 0.075, 0.0375)	0.528
(100, 0.1, 0.1, 0.1)	0.625	(100, 0.1, 0.05, 0.025)	0.546

Table 3.12: Powers for hypothesis testing H_0 : degree $= 2$ vs. H_1 : degree $= 3$ (with $\alpha = 0.05$)

From Tables 3.11 and 3.12, we can see that the level for our hypothesis test will be relatively stable with different network size, and we can always obtain the correct level

for different correlation structure parameters q_i $(i = 1, 2, 3)$. Our test will be more powerful for larger network when the edge probability p_e is fixed. The power of the hypothesis testing will be larger if we increase the correlation parameters q_i.

Change p_e with network size n

However, when the size of the network n is large, it is not proper to keep the edge probability p_e as a constant, so we assigned smaller values of p_e for large $n's$. Since we know the true values of the correlation parameters q_1, q_2, q_3 in this simulation study, the columns contain `true parameters` shows the results using the true values of parameters in hypothesis testing. However, in real data analysis, the true values of q_i are not available, and have to be estimated first. The columns contain `estimated parameters` indicates the results for levels or powers by estimating the nuisance parameters and using double bootstrap method. The following tables show the levels and powers of hypothesis tests H_0 : degree = 1 vs. H_1 : degree = 2 for different values of (n, p_e, q_1, q_2).

$(n,\ p_e,\ q_1,\ q_2)$	Level (estimated parameters)	Level (true parameters)
(20, 0.1, 0.2, 0.2)	0.053	0.046
(50, 0.1, 0.15, 0.15)	0.047	0.061
(100, 0.1, 0.15, 0.15)	0.052	0.048
(200, 0.05, 0.15, 0.15)	0.048	0.052
(500, 0.05, 0.15, 0.15)	0.064	0.058
(1000, 0.025, 0.1, 0.1)	0.051	0.057
(2000, 0.025, 0.1, 0.1)	0.056	0.045

Table 3.13: Levels for hypothesis testing H_0 : degree = 1 vs. H_1 : degree = 2 for larger networks

(n, p_e, q_1, q_2)	Power (estimated parameters)	Power (true parameters)
(20, 0.1, 0.2, 0.2)	0.325	0.315
(50, 0.1, 0.15, 0.15)	0.398	0.456
(100, 0.1, 0.15, 0.15)	0.473	0.532
(200, 0.05, 0.15, 0.15)	0.483	0.463
(500, 0.05, 0.15, 0.15)	0.490	0.505
(1000, 0.025, 0.1, 0.1)	0.542	0.564
(2000, 0.025, 0.1, 0.1)	0.536	0.593

Table 3.14: Powers for two hypothesis testing H_0 : degree $= 1$ vs. H_1 : degree $= 2$ for larger networks (with $\alpha = 0.05$)

From Table 3.13, our proposed method can preserve the levels for larger and sparser networks. The type I errors for both methods are close to the pre-specified level no matter using true parameters or estimated parameters. From Table 3.14, we can see that if we fix the parameters for correlation structure, the power of hypothesis testing will increase when the network size is larger. If we consider using the true parameter values, the power will be larger than using the estimated values and double bootstrap in most cases. In real data analysis, we can only consider the procedure with estimating the correlation parameters.

3.6 Real data analysis

3.6.1 Twitter data

Twitter (https://twitter.com) is an American online social media which provides a platform for users to post message ('tweet') and interact with other people. We analyzed a network with 244 nodes, 2478 edges, and 200 features for each individual, and this network is collected by Leskovec and Krevl (2014). For some of those features, their names start with hashtag '#', which is used to index keywords or topics on Twitter, and it allows people to

easily follow topics that they are interested in. For other features, their names start with at ('@'), which can directly interact with some other users including people or institutions. The following plot shows the network structure of this Twitter network.

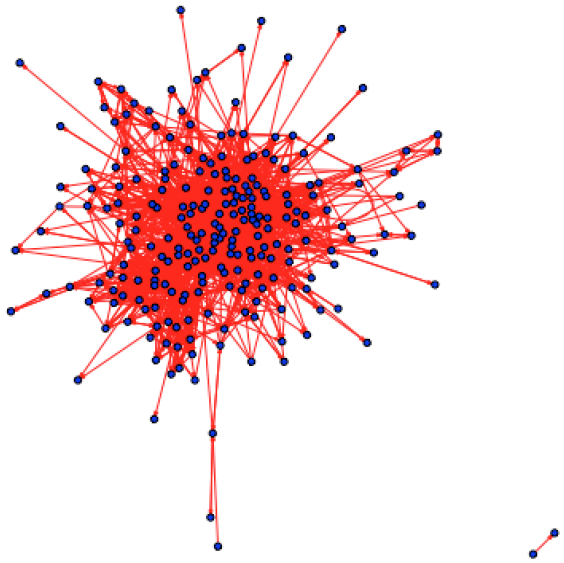


Figure 3.5: Twitter network structure

Based on sequential hypothesis testing procedure, we obtain the following results, including considering the Christakis-Fowler method and our proposed method. For the hypothesis testing H_0 : degree $= d-1$ vs H_1 : degree $= d$, we set the double bootstrap parameters to be $B_1 = 200$ and $B_2 = 500$. The frequency table of the degrees of influence is shown as follows.

Degrees of influence	0	1	2	3	4+
Number of features (Christakis-Fowler)	131	48	16	4	1
Number of features (proposed method)	125	36	31	5	3

Table 3.15: Degrees of influence of the Twitter data obtained from the Christakis-Fowler and the proposed method

From Table 3.15, we can see that there are more than 60% of features whose degrees of influence is 0, and only a few feature with degrees of influence greater than 3. In order to

indicate the difference of results between the two methods, we use the following contingency table to show the degrees of influences obtained from the Christakis-Fowler method and our proposed method:

	Christakis-Fowler					
Proposed method	0	1	2	3	4+	total
0	121	7	3	0	0	131
1	3	29	16	0	0	48
2	1	0	12	2	1	16
3	0	0	0	3	1	4
4+	0	0	0	0	1	1
total	125	36	31	5	3	200

Table 3.16: Comparing the degrees of influence between the Christakis-Fowler method and the proposed method for the Twitter data

From Table 3.16, we can see that these two methods give us different results for some features. After extracting some features with different degrees of influences, we obtain the following table with feature names and the corresponding degrees of influence.

feature	#appstore	#oil	@Cabel	@DeanDMX	#Android	@Dropbox	@BarackObama	@Berkeley
Proposed method	1	1	1	1	1	2	2	3
Christakis-Fowler	2	2	2	2	2	4+	3	4+

Table 3.17: Features with different degrees of influence for the Christakis-Fowler method and the proposed method

Table 3.17 shows that the Christakis-Fowler method gives higher degrees of influence sometimes. For features `#appstore`, `#oil`, `@Cabel`, `@DeanDMX` and `#Android`, the degrees of influence obtained from our proposed method is 1, but using the Christakis-Fowler method will lead the result to be 2. For the feature `@Dropbox`, the difference between the results obtained from these two methods are relatively large.

For the whole twitter data, Leskovec and Krevl (2014) provides more than 800 networks in total, but for each network, the name and the number of features are various. All features start with either '#' or '@', which indicate the users' interest in some way. In general, our proposed method gives smaller degrees of influences for some features comparing with the Christakis-Fowler method.

3.6.2 Pokec data

Pokec (https://pokec.azet.sk) is the most popular on-line social network in Slovakia, and its popularity has not changed even after the coming of Facebook. Pokec has been provided for more than 10 years and connects more than 1.6 million people. The dataset analyzed in this chapter is also from Leskovec and Krevl (2014), and it contains anonymized data of the whole network. Also, friendships in Pokec are oriented (directed).

Since the original dataset is too large (with more than 1 million nodes), we only consider a subset of the network, which only contains the people from a particular region Ceska. The size of the smaller network is 18,216.

There are 65 features for each user, including gender, age, hair color, hobbies, interests, education level etc. Here, We only consider the following four features of interests: relaition_to_smoke, relation_to_alcohol, like_comedy, going_to_concerts. All of these fours features are represented as binary variables, which indicate whether a person smokes, drinks alcohol, likes comedy and goes to concerts. Since some users did not complete their profiles, the original data contain missing values for some features. Here, we can choose to remove all nodes with missing profiles and analyze a smaller dataset which contains 9825 nodes. The following plot shows the network structure of this Pokec network.

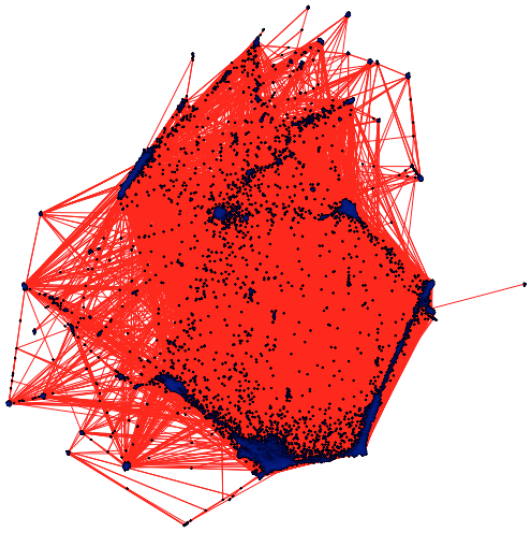


Figure 3.6: Pokec network structure

For our proposed method, we set $B_1 = 100$ and $B_2 = 200$ in the double bootstrap step when obtaining the p-value at each step. The degrees of influence for the four features with different methods are shown as follows.

Name of feature	smoke	alcohol	comedy	concert
Proposed method	1	2	2	1
Christakis-Fowler	2	3	2	1

Table 3.18: Degrees of influence of the Pokec data obtained from the Christakis-Fowler and the proposed method

From Table 3.18, we can see that for features smoke, alcohol, our proposed method gives smaller degrees of influence than the Christakis-Fowler method. Among these four features, the degrees of influence for relation_to_alcohol is the largest one no matter which method we use. In general, the two features going to concert and smoking have smaller degrees of influence than drinking alcohols and liking comedies. The results also provide some evidence that some bad habits could be influenced through a process in a network. For teenagers'

parents, it is necessary for them to tell their children not being friends with others who smoke or be addicted to alcohol.

3.7 Discussion

In this chapter, we build a multivariate Bernoulli model for static network with various degrees of influence. Also, we proposal a sequential hypothesis testing procedure to explore the degrees of influence with randomization test. Furthermore, we provide some theoretical justifications to show that our proposed hypothesis testing is more powerful for large networks. The approach for exploring the degrees of influence performs well for both simulation studies and real data analyses. We also find some features whose degrees of influence are greater than zero, such as smoking, drinking alcohol, and the results indicate that people's behavior or habits could be affected by others.

For the future work, we can consider detecting the degrees of influence for dynamic network, which means the status of each node and the network structure can be changed with time. Another potential topic we can explore is to find some better ways to deal with missing data in network.

3.8 Proofs

3.8.1 Notations

We define some symbols which will be used in the following proofs. For two sequences $\{x_n\}_{n=1}^{\infty}$ and $\{y_n\}_{n=1}^{\infty}$, we write

- $x_n = O(y_n)$, if $\exists\ M > 0$, such that $|x_n/y_n| \leq M$ for all large n
- $x_n \asymp y_n$, if $\exists\, M_2 > M_1 > 0$, such that $M_1 \leq |x_n/y_n| \leq M_2$ for all large n
- $x_n \preceq y_n$, if $x_n = O(y_n)$

3.8.2 Proof of Lemma 3.1

Proof. For any $k+1$ different nodes $(i_0, i_1, \ldots, i_k)$ in the graph, let $X(i_0, i_1, \ldots, i_k)$ be the indicator that all edges on the path $(i_0, i_1, \ldots, i_k)$ are presented, so we have

$$X(i_0, i_1, \ldots, i_k) = \begin{cases} 1 & \text{if } a_{i_u, i_{u+1}} = 1 \ (u = 0, 1, \ldots, k-1) \\ 0 & \text{otherwise} \end{cases}.$$

Then, $X(i_0, i_1, \ldots, i_k)$ follows a Bernoulli distribution with parameter p_e^k.

Let Y_n be the total number of simple k-paths, then

$$Y_n = \sum_{\text{all distinct choices of } (i_0, i_1, \ldots, i_k)} X(i_0, i_1, \ldots, i_k),$$

and the expectation of Y_n is

$$E(Y_n) = \frac{n!}{(n-k-1)!} p_e^k \asymp n^{k+1} p_e^k \asymp \lambda^k n.$$

The variance of Y_n is

$$Var(Y_n) = \sum_{(i_0, i_1, \ldots, i_k)} \sum_{(j_0, j_1, \ldots, j_k)} Cov(X(i_0, i_1, \ldots, i_k), X(j_0, j_1, \ldots, j_k)).$$

Suppose there are m common edges between the two edge sets $\{(i_u, i_{u+1}), u = 0, 1, \cdots, k-1\}$ and $\{(j_v, j_{v+1}), v = 0, 1, \cdots, k-1\}$, then

$$\begin{aligned} Cov(X(i_0, i_1, \ldots, i_k), X(j_0, j_1, \ldots, j_k)) &= E[(X(i_0, i_1, \ldots, i_k) - p_e^k)(X(j_0, j_1, \ldots, j_k) - p_e^k)] \\ &= E[X(i_0, i_1, \ldots, i_k) X(j_0, j_1, \ldots, j_k)] - p_e^{2k} \\ &= p_e^{2k-m} - p_e^{2k}. \end{aligned}$$

Let N_m be the number of path pairs $((i_0, i_1, \ldots, i_k), (j_0, j_1, \ldots, j_k))$ with m common edges,

and it is bounded by

$$N_m \leq \frac{n!}{(n-m-1)!} n^{k-m} n^{k-m} \asymp n^{2k-m+1}.$$

Then the variance of Y_n is bounded by

$$Var(Y_n) = \sum_{m=0}^{k} (p_e^{2k-m} - p_e^{2k}) N_m \preceq n \sum_{m=1}^{k} \lambda^{2k-m}.$$

By Paley-Zygmund inequality (Paley and Zygmund, 1932), for any $a \in (0,1)$, we have

$$P(Y_n > aE(Y_n)) \geq (1-a)^2 \frac{(EY_n)^2}{(EY_n)^2 + Var(Y_n)} \geq (1-a)^2 \frac{1}{1+O(1/n)} \asymp (1-a)^2.$$

For any $M > 0$, let $a_n = \frac{M}{E[Y_n]}$, then $P(Y_n > M) \geq (1 - a_n^2)$. Since $E[Y_n] \asymp \lambda^k n$, we have $a_n \to 0$. So, $P(Y_n > M) \to 1$ as $n \to \infty$, and then Y_n goes to infinity in probability. □

3.8.3 Proof of Theorem 3.2

Proof. For the hypothesis testing: H_0 : degree $= 0$ v.s. H_1 : degree $= 1$.

The test statistic is

$$T = \hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_1} = 1) - \hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_1} = 0)$$

Suppose there are N_s pairs of $(\text{ego}, \text{alter}_1)$ with $Y_{\text{alter}_1} = s$ $(s = 0, 1)$. Under H_0, since Y_{ego} and Y_{alter_1} are independent, we have

$$P(Y_{\text{ego}} = 1 | Y_{\text{alter}_1} = 1) = P(Y_{\text{ego}} = 1 | Y_{\text{alter}_1} = 0) = p,$$

for all pairs of $(\text{ego}, \text{alter}_1)$.

Given the following contingency table

	$Y_{\text{alter}_1} = 1$	$Y_{\text{alter}_1} = 0$
$Y_{\text{ego}} = 1$	a	b
$Y_{\text{ego}} = 0$	c	d

Table 3.19: Contingency table for hypothesis testing H_0 : degree $= 0$ vs. H_1 : degree $= 1$

we have

$$\hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_1} = 1) = \frac{1}{N_1}(X_{1,1} + \cdots + X_{1,N_1}),$$

$$\hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_1} = 0) = \frac{1}{N_0}(X_{0,1} + \cdots + X_{0,N_0}),$$

where $X_{1,i} \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(p)$ $(i = 1, \cdots, N_1)$ and $X_{0,j} \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(p)$ $(j = 1, \cdots, N_0)$. If $p_e = O(\lambda/n)$, then the total number of edges in graph goes to infinity as $n \to \infty$, and we also have $N_1 \to \infty$ and $N_0 \to \infty$. From the law of large number, we have

$$\hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_1} = 1) \xrightarrow{p} p,$$

$$\hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_1} = 0) \xrightarrow{p} p,$$

as $n \to \infty$. Thus $T \to 0$ in probability under H_0 as $n \to \infty$, and then the critical value of the hypothesis testing $c^* \to 0$ in probability.

Under H_1, for each pair of $(\text{ego}, \text{alter}_1)$, we assume $Corr(Y_{\text{ego}}, Y_{\text{alter}_1}) = q_1$.

$$\begin{aligned}
P(Y_{\text{ego}} = 1, Y_{\text{alter}_1} = 1) &= E[Y_{\text{ego}} Y_{\text{alter}_1}] \;=\; Cov(Y_{\text{ego}}, Y_{\text{alter}_1}) + E[Y_{\text{ego}}]E[Y_{\text{alter}_1}] \\
&= q_1\, p(1-p) + p^2, \\
P(Y_{\text{ego}} = 1, Y_{\text{alter}_1} = 0) &= P(Y_{\text{ego}} = 1) - P(Y_{\text{ego}} = 1, Y_{\text{alter}_1} = 1) \\
&= p - q_1\, p(1-p) - p^2 = p(1-p)(1-q_1), \\
P(Y_{\text{ego}} = 1 | Y_{\text{alter}_1} = 1) &= \frac{P(Y_{\text{ego}} = 1, Y_{\text{alter}_1} = 1)}{P(Y_{\text{alter}_1} = 1)} \;=\; p + q_1 - pq_1 \;\triangleq\; \tilde{p_1}, \\
P(Y_{\text{ego}} = 1 | Y_{\text{alter}_1} = 0) &= \frac{P(Y_{\text{ego}} = 1, Y_{\text{alter}_1} = 0)}{P(Y_{\text{alter}_1} = 0)} \;=\; p - pq_1 \;\triangleq\; \tilde{p_0}.
\end{aligned}$$

We have

$$\hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_1} = 1) = \frac{1}{N_1}(X_{1,1} + \cdots + X_{1,N_1}) \triangleq \bar{X}_1,$$

where $X_{1,i} \sim \text{Bernoulli}(\tilde{p_1})$ $(i = 1, \cdots, N_1)$, but they are not independent. Them, we have

$$\begin{aligned} Var(\bar{X}_1) &= \frac{\tilde{p_1}(1-\tilde{p_1})}{N_1} + \frac{2}{N_1^2} \sum_{1 \le i < j \le N_1} Cov(X_{1,i}, X_{1,j}) \\ &= \frac{\tilde{p_1}(1-\tilde{p_1})}{N_1} + \frac{2}{N_1^2} q_1 \tilde{p_1}(1-\tilde{p_1}) p_e \binom{N_1}{2}. \end{aligned}$$

Since $p_e = O(\lambda/n)$, we have $Var(\bar{X}_1) \to 0$. For any $\epsilon > 0$, by Chebyshev's inequality,

$$P(|\bar{X}_1 - \tilde{p_1}|) > \epsilon) \le \frac{Var(\bar{X}_1)}{\epsilon^2} \to 0.$$

then $\bar{X}_1 \to \tilde{p_1}$ in probability. Similarly, we have

$$\hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_1} = 0) = \frac{1}{N_0}(X_{0,1} + \cdots + X_{0,N_0}) = \bar{X}_0,$$

and $\bar{X}_0 \to \tilde{p_0}$ in probability. For the test statistics $T = \bar{X}_1 - \bar{X}_0 \xrightarrow{p} \tilde{p_1} - \tilde{p_0} = q_1 > 0$. Thus, the power $P(T > c^* | H_1) \to 1$ as $n \to \infty$.

For the hypothesis testing $H_0 : \text{degree} = d - 1$ v.s. $H_1 : \text{degree} = d$, the test statistic is

$$\begin{aligned} T &= \hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_{d-1}} = 1, Y_{\text{alter}_d} = 1) - \hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_{d-1}} = 1, Y_{\text{alter}_d} = 0) \\ &+ \hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_{d-1}} = 0, Y_{\text{alter}_d} = 1) - \hat{P}(Y_{\text{ego}} = 1 | Y_{\text{alter}_{d-1}} = 0, Y_{\text{alter}_d} = 0). \end{aligned}$$

Under H_0, the degrees of influence is $d - 1$, so Y_{ego} and Y_{alter_d} are independent. Given two contingency tables

	$Y_{\text{alter}_d}=1$	$Y_{\text{alter}_d}=0$
$Y_{\text{ego}}=1$	a_2	b_2
$Y_{\text{ego}}=0$	c_2	d_2

Table 3.20: Contingency table for hypothesis testing H_0 : degree $= d-1$ vs. H_1 : degree $= d$ when $Y_{\text{alter}_{d-1}}=1$

	$Y_{\text{alter}_d}=1$	$Y_{\text{alter}_d}=0$
$Y_{\text{ego}}=1$	e_2	f_2
$Y_{\text{ego}}=0$	g_2	h_2

Table 3.21: Contingency table for hypothesis testing H_0 : degree $= d-1$ vs. H_1 : degree $= d$ when $Y_{\text{alter}_{d-1}}=0$

We have

$$P(Y_{\text{ego}}=1|Y_{\text{alter}_{d-1}}=1,Y_{\text{alter}_d}=1)=P(Y_{\text{ego}}=1|Y_{\text{alter}_{d-1}}=1,Y_{\text{alter}_d}=0)=\tilde{p}_{1,d-1},$$

$$P(Y_{\text{ego}}=1|Y_{\text{alter}_{d-1}}=0,Y_{\text{alter}_d}=1)=P(Y_{\text{ego}}=1|Y_{\text{alter}_{d-1}}=0,Y_{\text{alter}_d}=0)=\tilde{p}_{0,d-1},$$

where $\tilde{p}_{1,d-1}=P(Y_{\text{ego}}=1|Y_{\text{alter}_{d-1}}=1)=p+q_{d-1}-pq_{d-1}$ and $\tilde{p}_{0,d-1}=P(Y_{\text{ego}}=1|Y_{\text{alter}_{d-1}}=0)=p-pq_{d-1}$.

Suppose there are N_{st} tuples of $(\text{ego},\text{alter}_{d-1},\text{alter}_d)$ with $Y_{\text{alter}_{d-1}}=s$ and $Y_{\text{alter}_d}=t$, where $s=0$ or 1 and $t=0$ or 1. From theorem 3.1, we have the number of pairs $(\text{ego},\text{alter}_d)$ goes to infinity as $n\to\infty$. Since $P(Y_{\text{alter}_d}=s,Y_{\text{alter}_d}=t)>0$ for all pairs of (s,t), we also have $N_{st}\to\infty$. Similar to the previous part of the proof, we have

$$Var(\bar{X}_{st})=\frac{\tilde{p}_{s,d-1}(1-\tilde{p}_{s,d-1})}{N_{st}}+\frac{2}{N_{st}^2}\,q_{d-1}\,\tilde{p}_{s,d-1}(1-\tilde{p}_{s,d-1})p_e\binom{N_{st}}{2}\to 0.$$

By Chebyshev's inequality, we have $\bar{X}_{st}\to\tilde{p}_{s,d-1}$ in probability. From $T=\bar{X}_{11}-\bar{X}_{10}+\bar{X}_{01}-\bar{X}_{00}$, $T\to 0$ in probability under H_0 as $n\to\infty$. Then the critical value of the hypothesis

testing $c^* \to 0$ in probability.

Under H_1 : degree $= d$, so Y_{ego} and Y_{alter_d} are not independent, we have

$$P(Y_{\text{ego}} = 1 | Y_{\text{alter}_{d-1}} = 1, Y_{\text{alter}_d} = 1) > P(Y_{\text{ego}} = 1 | Y_{\text{alter}_{d-1}} = 1, Y_{\text{alter}_d} = 0),$$

$$P(Y_{\text{ego}} = 1 | Y_{\text{alter}_{d-1}} = 0, Y_{\text{alter}_d} = 1) > P(Y_{\text{ego}} = 1 | Y_{\text{alter}_{d-1}} = 0, Y_{\text{alter}_d} = 0),$$

We still have $\bar{X}_{st} \to P(Y_{\text{ego}} = 1 | Y_{\text{alter}_{d-1}} = s, Y_{\text{alter}_d} = t)$ in probability because of the fact that $N_{st} \to \infty$. Since $T = \bar{X}_{11} - \bar{X}_{10} + \bar{X}_{01} - \bar{X}_{00}$, we have the test statistic $T \xrightarrow{p} E(T) > 0$. Thus, the power $P(T > c^* | \text{degree} = d) \to 1$ as $n \to \infty$. □

Chapter 4

Higher-order motif spectral clustering under a small-world dyads-triads random graph model

4.1 Introduction

We propose a random graph model for undirected networks with small-world properties, namely high clustering coefficient and low average path length. In the most basic form, the proposed model is a superimposition of a regular Erdős-Rényi dyadic (edge based) random graph $G_e(n, p_e)$ and a Erdős-Rényi triadic (triangle-based) random graph $G_t(n, p_t)$, where n denotes the number of nodes and p_e and p_t denote the probability of an edge in the dyadic graph and a triangle in the triadic graph respectively. A random graph from the model can be generated as follows. We start with n unconnected vertices. The $G_t(n, p_t)$ graph is generated by independently randomly placing a triangle in any of the $\binom{n}{3}$ 3-tuple of vertices. The $G_e(n, p_e)$ graph is generated by randomly placing edges with probability p_e in an identical copy of the vertices. The two graphs are then superimposed to obtain the final graph. We let the graph contain multiple edges between two nodes if and only if that edge is involved in a triangle. We let an additional edge between two nodes for each triangle the nodes together are involved in. (A possible variation could be to take only the vertices that are not a vertex of a triangle and form ER edges in between those vertices. This circumvents the problem of multiedges, but then we have to deal with non-independence)

4.2 Basic properties of our model

- Erdős-Rényi random graphs $G_e(n, p_e)$ and $G_t(n, p_t)$

- Expected number of edges: $\binom{n}{2}p_e + 3\binom{n}{3}p_t$
- Expected number of triangles: $\binom{n}{3}p_e^3 + \binom{n}{3}p_t$
- Degree decomposition: $d = d_e + d_t$, with $d_e \sim \text{Binomial}(n-1, p_e)$ and $d_t \sim 2\,\text{Binomial}\left(\binom{n-1}{2}, p_t\right)$
- Expected degree of a node: $(n-1)p_e + 2\binom{n-1}{2}p_t = (n-1)(p_e + (n-2)p_t)$
  We choose $d = O(\lambda)$, $p_e = O(\lambda/n)$ and $p_t = O(\lambda/n^2)$.
- Expected number of triangles a node is connected to: $\binom{n-1}{2}p_e^3 + \binom{n-1}{2}p_t$

4.3 Probability of multiedge

We fix the nodes i and j, there are two ways we can have a multiedge between these two nodes. Suppose there are t_{ij} triangles in G_t with vertices i and j, then $t_{ij} \sim \text{Binomial}(n-2, p_t)$, and we use e_{ij} to denote whether there is an edge between i and j in G_e. We can calculate the probability of multiedge in the following way:

$$\begin{aligned}
P(\text{multiedge}) &= P(e_{ij}=1)P(t_{ij}\geq 1) + P(e_{ij}=0)P(t_{ij}\geq 2) \\
&= p_e P(t_{ij}\geq 1) + (1-p_e)P(t_{ij}\geq 2) \\
&= 1 - P(t_{ij}=0) - (1-p_e)P(t_{ij}=1) \\
&= 1 - (1-p_t)^{n-2} - (1-p_e)(n-2)(1-p_t)^{n-3}p_t \\
&\asymp 1 - e^{(n-2)\log(1-\lambda/n^2)} - (1-\frac{\lambda}{n})(n-2)\frac{\lambda}{n^2}e^{(n-3)\log(1-\lambda/n^2)} \\
&\asymp 1 - e^{-\lambda/n} - \frac{(n-\lambda)\lambda}{n^2} = O(\lambda^2/n^2).
\end{aligned}$$

The expected number of edges who are multiedges in the graph is $O(\lambda^2)$.

We can compare it with the configuration model introduces the given degree for each node is $k_i = \lambda$ $(i = 1, \ldots, n)$. For any fixed pair of nodes i and j, the probability that they are connected is:

$$p_{ij} = \frac{k_i k_j}{2n-1} = \frac{\lambda^2}{2n-1} \simeq \frac{\lambda^2}{2n}.$$

Also, the probability that a second edge appears is $p_{ij}^{(2)} = (k_i - 1)(k_j - 1)/2n = (\lambda - 1)^2/2n$. The expected number of multiegdes in the configuration model will be

$$\sum_{i \neq j} p_{ij}\, p_{ij}^{(2)} = \binom{n}{2} \frac{\lambda^2}{2n} \frac{(\lambda - 1)^2}{2n} = O(\lambda^4).$$

4.4 Local clustering coefficient analysis

The local clustering coefficient (also known as transitivity) of a node u defined as follows:

$$cc(u) = \frac{\text{number of pairs of neighbors of u connected by an edge}}{\text{number of pairs of neighbors of } u}.$$

Suppose there are c_u triangles in G_t with vertex as u, then according to the previous definition, we have $c_u = \frac{d_t}{2}$. From the definition of local clustering coefficient, it is easy to show that:

$$cc(u) = \frac{\binom{d_e}{2} p_e + \frac{d_t}{2}}{\binom{d_e}{2} + \frac{d_t}{2}} \triangleq \frac{N}{D}.$$

Also, we assume the two random variables d_e and d_t are independent, where $d_e \sim \text{Binomial}(n-1, p_e)$, $\frac{d_t}{2} \sim \text{Binomial}\left(\binom{n-1}{2}, p_t\right)$. Using first order Taylor expansion, assured by the concentration of N and D around their means, we have

$$E[cc(u)] \simeq \frac{E[N]}{E[D]} + o_p\left(\frac{E[N]}{E[D]}\right),$$

where o_p means convergence in probability.

Next, we will show the asymptotic properties of both the enumerator $E[N]$ and the

denominator $E[D]$ as follows:

$$\begin{aligned} E[N] &= \frac{1}{2}p_e E[d_e(d_e-1)] + \frac{1}{2}E[d_t] = \frac{1}{2}p_e(E[d_e^2] - E[d_e]) + \frac{1}{2}E[d_t] \\ &= \frac{1}{2}p_e[(n-1)p_e(1-p_e) + (n-1)^2 p_e^2 - (n-1)p_e] + \binom{n-1}{2}p_t \\ &= \frac{1}{2}(n-1)(n-2)p_e^3 + \binom{n-1}{2}p_t = \frac{1}{2}(n-1)(n-2)(p_e^3 + p_t). \end{aligned}$$

Analogously, we can calculate

$$\begin{aligned} E[D] &= \frac{1}{2}E[d_e(d_e-1)] + \frac{1}{2}E[d_t] = \frac{1}{2}(E[d_e^2] - E[d_e]) + \frac{1}{2}E[d_t] \\ &= \frac{1}{2}[(n-1)p_e(1-p_e) + (n-1)^2 p_e^2 - (n-1)p_e] + \binom{n-1}{2}p_t \\ &= \frac{1}{2}(n-1)(n-2)p_e^2 + \binom{n-1}{2}p_t = \frac{1}{2}(n-1)(n-2)(p_e^2 + p_t). \end{aligned}$$

Hence, we have

$$E[cc(u)] \asymp \frac{p_e^3 + p_t}{p_e^2 + p_t}.$$

Since we choose $d = O(\lambda)$, $p_e = O(\lambda/n)$ and $p_t = O(\lambda/n^2)$, we have the expected local clustering coefficient from ER dyads and triads is:

$$E[cc(u)] = O(\frac{\lambda^3/n^3 + \lambda/n^2}{\lambda^2/n^2 + \lambda/n^2}) = O(\frac{\lambda^3 + n\lambda}{n\lambda^2 + n\lambda}) = O(1).$$

The expected local clustering coefficient from regular ER random graph with comparable degree density is $O(\lambda/n)$. It is clear that our model has higher clustering coefficient for comparable degree density.

4.5 Phase transition analysis

4.5.1 Connectivity threshold

For the original ER model, suppose the probability for the emergence of an edge between two nodes is p, then we have

$$P(\text{connectivity}) \to 0, \quad \text{if } \ p < \frac{\log n}{n}.$$

For our model, we have

$$P(\text{connectivity}) \to 0, \quad \text{if } \ p_e + \frac{n}{2} p_t < \frac{\log n}{n}.$$

Here, connectivity means there is no isolated node in the graph.

For a node i in the network, let $I(i)$, $I_e(i)$ and $I_t(i)$ be Bernoulli random variables, which are defined as follows

$$I(i) = \begin{cases} 1 & \text{if node } i \text{ is isolated in } G, \\ 0 & \text{otherwise.} \end{cases}$$

$$I_e(i) = \begin{cases} 1 & \text{if node } i \text{ is isolated in } G_e, \\ 0 & \text{otherwise.} \end{cases}$$

$$I_e(i) = \begin{cases} 1 & \text{if node } i \text{ is isolated in } G_t, \\ 0 & \text{otherwise.} \end{cases}$$

A node i is isolated in G if and only if i is isolated in both G_e and G_t. We can calculate the

probability that a node is isolated as follows:

$$\begin{aligned} q \triangleq P(I(i)=1) &= P(I_e(i)=1, I_t(i)=1) = P(I_e(i)=1)P(I_t(i)=1) \\ &= (1-p_e)^{n-1}(1-p_t)^{\binom{n-1}{2}} \\ &\asymp e^{-np_e-n^2p_t} \end{aligned}$$

For two different nodes i and j, the covariance of $I(i)$ and $I(j)$ can be calculated as follows:

$$\begin{aligned} Cov(I(i),I(j)) &= E[I(i)I(j)] - E[I(i)]E[I(j)] \\ &= P(I(i)=1, I(j)=1) - P(I(i)=1)P(I(j)=1) \\ &= (1-p_e)^{2(n-2)+1}(1-p_t)^{2\binom{n-2}{2}+(n-2)} - q^2 \\ &= \frac{q^2}{(1-p_e)(1-p_t)^{n-2}} - q^2 \end{aligned}$$

Let X_n be the total number of isolated nodes in G, then

$$X_n = \sum_{i=1}^{n} I(i).$$

The expectation of X_n is

$$E[X_n] = \sum_{i=1}^{n} E[I(i)] = \sum_{i=1}^{n} P(I(i)=1) = nq.$$

The variance of X_n is

$$\begin{aligned} Var(X_n) &= Var\left(\sum_{i=1}^{n} I(i)\right) \\ &= \sum_{i=1}^{n} Var(I(i)) + \sum_{i\neq j} Cov(I(i), I(j)) \\ &= nq(1-q) + n(n-1)\left(\frac{q^2}{(1-p_e)(1-p_t)^{n-2}} - q^2\right) \\ &= nq - n^2q^2 + n(n-1)q^2(1-p_e)^{-1}(1-p_t)^{-(n-2)} \end{aligned}$$

If $p_e + \frac{n}{2}p_t < \frac{\log n}{n}$, we have

$$E[X_n] = nq \asymp e^{n\left(\frac{\log n}{n} - (p_e + \frac{n}{2}p_t)\right)} \to \infty,$$

$$Var(X_n) \asymp nq - n^2q^2 + n(n-1)q^2 \asymp nq(1-q) \asymp nq.$$

By the second moment inequality, we have

$$P(X_n > 0) \geq \frac{(E[X_n])^2}{E[X_n^2]} = \frac{(E[X_n])^2}{Var(X_n) + (E[X_n])^2} = \frac{1}{1+o(1)} \asymp 1,$$

which implies

$$P(\text{connectivity}) = P(\text{no isolated node in } G) = P(X_n = 0) \to 0.$$

Thus, if $p_e + \frac{n}{2}p_t < \frac{\log n}{n}$, the graph is disconnected almost surely.

4.5.2 Giant component

We can also analyze the existence of giant component in our model. A giant component is a connected sub-graph in a given random graph which contains a constant fraction of all the vertices. In the original ER model, if $p > \frac{1}{n}$, the graph has a unique giant component;

and if $p < \frac{1}{n}$, all the components in the graph are small ones, having size $O(\log n)$ with a high probability.

For general undirected graph, we can determine whether the giant component exists or not by using the degree distribution. Since the specificity of our model, sometimes multiedges occurs between two nodes. But we need to count them only one time when considering the degree distribution, and we will denote the distribution as d_0, with the decomposition $d_0 = d_e + d_t - d_m$, where d_m indicates the multiedge. It is enough for us to judge the the existence of giant component, if we know the first and second moment of the random variable d_0. Let μ_i be the i^{th} moment of d_0, and $u_i = \sum_{k=0}^{\infty} k^i P(d_0 = k)$. According to Molloy and Reed (1995), if $\mu_2 > 2\mu_1$, there exists a giant component in the graph.

In this case, we have $d_e \sim \text{Binomial}(n-1, p_e)$ and $d_t \sim 2\,\text{Binomial}\left(\binom{n-1}{2}, p_t\right)$. Also, $0 \leq d_0 = d_e + d_t - d_m \leq n-1$ always holds. The first and second moments can be calculated as follows:

$$\mu_1 = E[d_0] = E[d_e] + E[d_t] - E[d_m] \leq E[d_e] + E[d_t],$$

$$\begin{aligned}
\mu_2 &= E[d_0^2] = E[(d_e + d_t - d_m)^2] \\
&= E[d_e^2] + E[d_t^2] + 2E[d_e d_t] + E[d_m(d_m - 2d_e - 2d_t)] \\
&\geq E[d_e^2] + E[d_t^2] + 2E[d_e d_t] - E[d_m(2(n-1) - d_m)] \\
&\geq E[d_e^2] + E[d_t^2] + 2E[d_e d_t] - 2(n-1)E[d_e + d_t].
\end{aligned}$$

Since d_e and d_t are independent, we have $E[d_e d_t] = E[d_e]E[d_t]$. In addition, the second moments for d_e and d_t are:

$$E[d_e^2] = Var(d_e) + (E[d_e])^2 = (n-1)p_e(1-p_e) + (n-1)^2 p_e^2,$$

$$E[d_t^2] = Var(d_t) + (E[d_t])^2 = (n-1)(n-2)p_t(1-p_t) + (n-1)^2(n-2)^2 p_t^2.$$

Thus, the condition for existing giant component in our model is $np_e + (n-1)(n-2)p_t > 1$. If we only consider G_t, people sometimes call it random hypergraph, which can be treated as a generalization of the original Erdős-Rényi graph. Schmidt-Pruzan and Shamir (1985) gives the claim that if $(n-1)(n-2)p_t > 1$, G_t has a large giant component.

References

Ali, S. M. and Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B*, pages 131–142.

Armagan, A. and Dunson, D. (2011). Sparse variational analysis of linear mixed models for large data sets. *Statistics & Probability Letters*, 81(8):1056–1062.

Bayarri, M. and Berger, J. O. (2000). P values for composite null models. *Journal of the American Statistical Association*, 95(452):1127–1142.

Beal, M. J. and Ghahramani, Z. (2003). The variational Bayesian em algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian Statistics*, 7:453–464.

Beran, R. (1988). Prepivoting test statistics: a bootstrap view of asymptotic refinements. *Journal of the American Statistical Association*, 83(403):687–697.

Berger, R. L. and Boos, D. D. (1994). P values maximized over a confidence set for the nuisance parameter. *Journal of the American Statistical Association*, 89(427):1012–1016.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Blei, D. M. and Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143.

Blei, D. M., Kucukelbir, A., and Mcauliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112:859–877.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022.

Bugallo, M. F., Elvira, V., Martino, L., Luengo, D., Miguez, J., and Djuric, P. M. (2017). Adaptive importance sampling: the past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79.

Cappé, O., Douc, R., Guillin, A., Marin, J.-M., and Robert, C. P. (2008). Adaptive importance sampling in general mixture classes. *Statistics and Computing*, 18(4):447–459.

Cappé, O., Guillin, A., Marin, J.-M., and Robert, C. P. (2004). Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929.

Centola, D. (2010). The spread of behavior in an online social network experiment. *science*, 329(5996):1194–1197.

Christakis, N. A. and Fowler, J. H. (2007). The spread of obesity in a large social network over 32 years. *New England Journal of Medicine*, 357(4):370–379.

Christakis, N. A. and Fowler, J. H. (2008). The collective dynamics of smoking in a large social network. *New England Journal of Medicine*, 358(21):2249–2258.

Christakis, N. A. and Fowler, J. H. (2013). Social contagion theory: examining dynamic social networks and human behavior. *Statistics in Medicine*, 32(4):556–577.

Dai, B., Ding, S., and Wahba, G. (2013). Multivariate bernoulli distribution. *Bernoulli*, 19(4):1465–1483.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, pages 1–38.

Dieng, A. B., Tran, D., Ranganath, R., Paisley, J., and Blei, D. (2017). Variational inference via χ upper bound minimization. In *Advances in Neural Information Processing Systems*, pages 2732–2741.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208.

Dowling, M., Nassar, J., Djurić, P. M., and Bugallo, M. F. (2018). Improved adaptive importance sampling based on variational inference. In *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*, pages 1632–1636. IEEE.

Dufour, J.-M. (2006). Monte carlo tests with nuisance parameters: A general approach to finite-sample inference and nonstandard asymptotics. *Journal of Econometrics*, 133(2):443–477.

Dwass, M. (1957). Modified randomization tests for nonparametric hypotheses. *The Annals of Mathematical Statistics*, pages 181–187.

Efron, B. (1992). Bootstrap methods: another look at the jackknife. In *Breakthroughs in Statistics*, pages 569–593. Springer.

Erdős, P. and Rényi, A. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60.

Fowler, J. H. and Christakis, N. A. (2008). Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study. *The BMJ*, 337:a2338.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):721–741.

Goyal, A., Bonchi, F., and Lakshmanan, L. V. (2011). A data-based approach to social influence maximization. *Proceedings of the VLDB Endowment*, 5(1):73–84.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Hofman, J. M. and Wiggins, C. H. (2008). Bayesian approach to network modularity. *Physical Review Letters*, 100(25):258701.

Hughes, M. C. and Sudderth, E. (2013). Memoized online variational inference for Dirichlet process mixture models. In *Advances in Neural Information Processing Systems*, pages 1133–1141.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.

Kempe, D., Kleinberg, J., and Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146. ACM.

Kempe, D., Kleinberg, J., and Tardos, É. (2005). Influential nodes in a diffusion model for social networks. In *International Colloquium on Automata, Languages, and Programming*, pages 1127–1138. Springer.

Kong, A. (1992). A note on importance sampling using standardized weights. *University of Chicago, Dept. of Statistics, Tech. Rep*, 348.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

La Fond, T. and Neville, J. (2010). Randomization tests for distinguishing social influence

and homophily effects. In *Proceedings of the 19th International Conference on World Wide Web*, pages 601–610. ACM.

Leisch, F., Weingessel, A., and Hornik, K. (1998). On the generation of correlated artificial binary data.

Leisch, F., Weingessel, A., and Hornik, K. (2012). *bindata: Generation of Artificial Binary Data*. R package version 0.9-19.

Leskovec, J. and Krevl, A. (2014). SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

Lewis, K., Gonzalez, M., and Kaufman, J. (2012). Social selection and peer influence in an online social network. *Proceedings of the National Academy of Sciences*, 109(1):68–72.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044.

MacKinnon, J. G. (2009). Bootstrap hypothesis testing. *Handbook of Computational Econometrics*, 183:213.

Martino, L., Elvira, V., and Louzada, F. (2017). Effective sample size for importance sampling based on discrepancy measures. *Signal Processing*, 131:386–401.

Miething, A., Rostila, M., Edling, C., and Rydgren, J. (2016). The influence of social network characteristics on peer clustering in smoking: A two-wave panel study of 19-and 23-year-old swedes. *PLoS One*, 11(10):e0164611.

Molloy, M. and Reed, B. (1995). A critical point for random graphs with a given degree sequence. *Random structures and algorithms*, 6(2-3):161–180.

Naesseth, C., Linderman, S., Ranganath, R., and Blei, D. (2018). Variational sequential Monte Carlo. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research*, pages 968–977.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, 11(2):125–139.

Neal, R. M. and Hinton, G. E. (1998). A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Springer.

O'Malley, A. J. (2013). The analysis of social network data: an exciting frontier for statisticians. *Statistics in Medicine*, 32(4):539–555.

Owen, A. B. (2013). *Monte Carlo Theory, Methods and Examples*. http://statweb.

stanford.edu/~owen/mc/.

Paley, R. and Zygmund, A. (1932). A note on analytic functions in the unit circle. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 28, pages 266–272. Cambridge University Press.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407.

Robins, J. M., van der Vaart, A., and Ventura, V. (2000). Asymptotic distribution of p values in composite null models. *Journal of the American Statistical Association*, 95(452):1143–1156.

Rosenquist, J. N., Murabito, J., Fowler, J. H., and Christakis, N. A. (2010). The spread of alcohol consumption behavior in a large social network. *Annals of Internal Medicine*, 152(7):426–433.

Sanguinetti, G., Lawrence, N. D., and Rattray, M. (2006). Probabilistic inference of transcription factor concentrations and gene-specific regulatory activities. *Bioinformatics*, 22(22):2775–2781.

Sason, I. and Verdú, S. (2016). f-divergence inequalities. *IEEE Transactions on Information Theory*, 62(11):5973–6006.

Schmidt-Pruzan, J. and Shamir, E. (1985). Component structure in the evolution of random hypergraphs. *Combinatorica*, 5(1):81–94.

Sewell, D. K. (2018). Heterogeneous susceptibilities in social influence models. *Social Networks*, 52:135–144.

Sun, J. and Tang, J. (2011). A survey of models and algorithms for social influence analysis. In *Social Network Data Analytics*, pages 177–214. Springer.

Tsui, K.-W. and Weerahandi, S. (1989). Generalized p-values in significance testing of hypotheses in the presence of nuisance parameters. *Journal of the American Statistical Association*, 84(406):602–607.

Valente, T. W. (1996). Social network thresholds in the diffusion of innovations. *Social Networks*, 18(1):69–89.

Wang, P. and Blunsom, P. (2013). Collapsed variational Bayesian inference for hidden Markov models. In *AISTATS*, pages 599–607.

Xing, E. P., Jordan, M. I., and Russell, S. (2002). A generalized mean field algorithm for variational inference in exponential families. In *Proceedings of the Nineteenth Conference*

on Uncertainty in Artificial Intelligence, pages 583–591. Morgan Kaufmann Publishers Inc.

You, C., Ormerod, J. T., and Mueller, S. (2014). On variational Bayes estimation and variational information criteria for linear regression models. *Australian & New Zealand Journal of Statistics*, 56(1):73–87.