Latent Class Models For At-Risk Populations

LATENT CLASS MODELS FOR AT-RISK POPULATIONS

A Dissertation Presented

by

SHUAIMIN KANG

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2020

Department of Mathematics and Statistics

LATENT CLASS MODELS FOR AT-RISK POPULATIONS

A Dissertation Presented

by

SHUAIMIN KANG

Approved as to style and content by:

Krista Gile, Chair

Anna Liu, Member

Daeyoung Kim, Member

Justin Gross, Member

Nathaniel Whitaker, Chair of the Faculty
Department of Mathematics and Statistics

DEDICATION

Although the world is full of suffering, it is also full of the overcoming of it.

- Helen Keller

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor, Krista Gile, who is a role model for me to be an independent and active thinker, a courteous and reliable person. I would also like to thank our collaborators, Megan Price, Pedro Mateu-Gelabert and Honoria Guarino, for sharing their data and knowledge about people in the data. Many thanks to my committee members, Anna Liu, Daeyoung Kim and Justin Gross, for supporting my defense at this special time. Great thanks to my friends and family who always trust me and love me!

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

CLUSTERING NETWORK TREE DATA FROM RESPONDENT-DRIVEN SAMPLING WITH APPLICATION TO OPIOID USERS IN NEW YORK CITY

1.1 Abstract

There is great interest in finding meaningful subgroups of attributed network data. There are many available methods for clustering complete network. Unfortunately, much network data is collected through sampling, and therefore incomplete. Respondent-driven sampling (RDS) is a widely used method for sampling hard-to-reach human populations based on tracing links in the underlying unobserved social network. The resulting data therefore have tree structure representing a sub-sample of the network, along with many nodal attributes. In this paper, we introduce an approach to adjust mixture models for general network clustering for samplings by RDS. We apply our model to data on opioid users in New York City, and detect communities reflecting group characteristics of interest for intervention activities, including drug use patterns, social connections and other community variables.

1.2 Introduction

Network clustering is used to detect groups within a graph where nodes in the same group have stronger social connections than nodes in different groups and where nodal attributes are more similar within groups. However, there are no existing methods for clustering social networks sampled with link-tracing mechanisms, such as Respondent-driven sampling (RDS). Traditional network clustering methods are

not appropriate for RDS networks because of the link tracing procedure in RDS. Clustering of networks with node or edge features is well studied [Yang et al., 2013], [Xu et al., 2012], [Qi et al., 2012]. In this paper, we build a mixture model for RDS network sample with node features, and add sampling weights to the likelihood to find clusters for the RDS network sample.

Respondent-driven sampling (RDS) [Heckathorn, 1997] is a link-tracing network sampling method popularly used in sampling data from hard-to-reach populations, such as drug users and sex workers. It starts by selecting several people in the target population as seeds, then those seeds expand the sample by distributing coupons to people they know, those newly added samples distribute coupons in a similar way, and this process continues until reaching the desired sample size. Each coupon has a unique number which makes clear who recruited whom. RDS is a sampling method without replacement and its resulting observed network has tree structure with each tree starting with a different seed. The maximum number of coupons one person can distribute or the maximum number of people each person can recruit is usually small, like 3, to make sure the tree is deep enough, which helps reduce dependency of samples in a tree on its seed.

Each sampled person in the RDS network completes a survey, creating a node-attributed RDS network. Some node-attributed RDS networks have obvious homophily [Gile and Handcock, 2010], which is the correlation between trait values of nodes connected by an edge. For example, in the opioid drug user RDS network, heavy drug users are more likely to be tied to, and therefore recruit heavy drug users. Network clustering methods have been developed extensively. Maximizing modularity [Newman, 2006], minimizing cut [Ding et al., 2001], eigenvector related spectral clustering [Ng et al., 2001] [Shi and Malik, 2002], and hierarchical clustering [Bandyopadhyay and Coyle, 2003] are widely used in computer science and biology to cluster complex graphs. Methods for clustering networks statistically through assigning dis-

tributions to network structures are also well developed. In the stochastic block model [Nowicki and Snijders, 2001][Karrer and Newman, 2011] [Airoldi et al., 2008], mixture and Bayesian mixture models [Daudin et al., 2008], edges follow Bernoulli or Bernoulli mixture distributions with the same connection probabilities if they're in the same block or community. Model based network clustering methods have also been used to cluster graphs with node or edge features. Handcock et al. (2007) models node pair connection probability as a logistic regression on covariates and the distance of the node pair in a latent social space. In Communities from Edge Structure and Node Attributes (CESNA) [Yang et al., 2013], links of the network and node attributes are modeled separately but connected by the node community membership probabilities. Xu et al. (2012) proposed a Bayesian probability model assuming network structure and node attributes are independent given node group status. In this paper, we build on Xu et al. (2012)'s assumption that node features and network structures are independent given node clustering status and build a mixture model from it. Since RDS generates incomplete network data with nodes and edges unequally sampled from a full network, the above network clustering methods are not valid. Therefore, we propose a weighted log-likelihood approach, adding nodal and edge inverse sampling probability weights (IPW) to the log-likelihood for inference.

In this paper, we are not only interested in clustering the RDS sample data, but also interested in the interpretation of those clusters and individuals within those clusters. To better interpret populations in each cluster, we should find and use less biased parameters given the sampled data. Weighting is a common way to reduce bias in sampled data. Weighted likelihood has been used in mixture models for reducing bias when outliers exist in the data [Markatou, 2000]. The inverse selection probability-weighted likelihood method has also been studied for fitting sampled data [Li et al., 2008] [Saegusa and Wellner, 2013]. Weighted likelihood has been used for automatic model selection in density mixture clustering [Cheung, 2005]. Weighted it-

erative clustering algorithms have also been well studied for better clustering [Topchy et al., 2004][Zhang, 2001][Hamerly and Elkan, 2002]. Based on those literatures and considering the un-equal sampling probabilities in RDS, the instances or nodes and edges in the RDS sample should not be treated equally. Therefore, we propose to add inverse sampling probabilities to the likelihood of the mixture model from the node attributed RDS sample data to approximate the likelihood in the pseduo-population, thus getting less biased parameter estimation and reasonable clustering.

In this paper, we review sampling probabilities in RDS in Section 1.3. We propose a mixture model without weights as Benchmark model and extend the Benchmark model by adding IPW in Section 1.4. Furthermore, we propose the weighted likelihood mixture model with tuning parameter to balance contribution of node features and network structure. In Section 1.5, we talk about evaluation of clustering algorithms and tuning parameter selection. In Section 1.6, we compare the approaches proposed in Section 1.4 through simulation studies. In Section 1.7, we apply our approach to opioid users' RDS data from New York City. In Section 1.8 , we summarize the weighted log-likelihood mixture model for clustering incomplete node attributed RDS network data.

1.3 Background

1.3.1 RDS network Structure and Notation

As a link-tracing without replacement sampling method, RDS results in tree structured graphs as in the RDS network sample in Figure 1.1. Each person in the network is called a node. If two nodes are connected, we say there is an edge or a tie connecting them. In general, an adjacency matrix is used to describe connections between nodes in the network. Assume there are N nodes in the full network and $n(n \leq N)$ nodes in the RDS sample. Denote $Y = [y_{ij}]_{N\times N}$ and $\tilde{Y} = [\tilde{y}_{ij}]_{n\times n}$ as adjacency matrices describing the full and RDS network structures, respectively.

In this paper, we focus on un-directed networks only, such that

$$y_{ij} = y_{ji} = \begin{cases} 1, & \text{if nodes } i, \text{and } j \text{ are connected in full network} \\ 0, & \text{otherwise,} \end{cases}$$

$$\tilde{y}_{ij} = \tilde{y}_{ji} = \begin{cases} 1, & \text{if nodes } i, \text{and } j \text{ are sampled and connected in the RDS sample} \\ 0, & \text{if node } i, \text{node } j \text{ are sampled, but not connected in the RDS sample.} \end{cases}$$

The number of edges incident to a node is called the degree of that node. In Figure 1.1, each node has a degree at most 4. This is because RDS restricts each respondent's recruitment has to be no more than 3. This results in two types of degree for nodes in the RDS network sample, one is their degree in the RDS sample, and the other one is their degree in the hidden full network. For example, in the drug user RDS network, if person A is recruited as a sample, even though its degree in the RDS network is 3, its degree in the population might be greater than 3 because person A might know more than 3 drug users and he just recruited two or three of them into the sample. We denote the degree for node i in the hidden full network as d_i. In this paper, when we use degree we mean degree in the population if not otherwise specified.

RDS data usually have node features describing each sample. We focus on clustering node-attributed RDS sample in this paper. Assume we have one continuous and one discrete feature describing the nodes. Without loss of generality, we label the sampled nodes with indices $1, \cdots, n$. Then,

- X_1 and $\tilde{X}_1$ are the continuous variables for the full and RDS networks, respectively.
- X_2 and $\tilde{X}_2$ are the discrete variables for the full and RDS networks, respectively.

- $Z = [z_{ik}]_{N\times K}$ and $\tilde{Z} = [\tilde{z}_{ik}]_{n\times K}$ are matrices describing latent cluster status for the attributed full and RDS networks. K is the number of latent clusters in the full network.

$$z_{ik} = \begin{cases} 1, & \text{if node i is in the } k^{th} \text{ cluster} \\ 0, & \text{otherwise,} \end{cases}$$

$$\tilde{z}_{ik} = \begin{cases} 1, & \text{if node i is in the } k^{th} \text{ cluster and is sampled} \\ 0, & \text{otherwise,} \end{cases}$$

Note that our goal is to get latent group memberships for nodes in the RDS network sample, which reflect their group memberships in the full network, which is $z_{ik} = \tilde{z}_{ik}$ for node i in the RDS network. Furthermore,

- $S = [S_i]_{n\times 1}$ is the node sampling probability vector, where

$$S_i = P(\text{node i is sampled}).$$

- $SS = [SS_{ij}]_{n\times n}$ is the node pair sampling probability matrix,

$$SS_{ij} = P(\text{node i and node j are sampled}).$$

- $R = [R_{ij}]_{n\times n}$ is the edge sampling probability matrix,

$$R_{ij} = P(\tilde{Y}_{ij} = 1 | Y_{ij} = 1).$$

1.3.2 Node and Node Pair Sampling Probabilities in RDS

The sampling probability for each node is highly related with its degree in the population. Taking an extreme case as an example, when we sample drug users' networks using RDS, if drug user A knows zero other drug users, and drug user B is a drug dealer who knows many other drug users, then person B has much higher degree than drug user A and has much higher probability to be sampled than person A, because person B knows many more other drug users and is more likely to be recruited into the sample. Since we have node features describing each node in the RDS network, unequal node sampling probabilities also means that those node features are sampled unequally. Therefore, in order to get a log-likelihood representing the full network from node features of the sample, taking node sampling probabilities into consideration is necessary.

RDS is a without replacement sampling procedure, so node sampling probability is not simply proportional to its degree. Gile (2011) proposed successive sampling (SS) to get improved node sampling probabilities. By iterating the successive sampling procedure to approximate RDS, Gile (2011) mapped nodes with degree k to their sampling probabilities S_k with $f : d \to S_k$. Following Gile (2011)'s node sampling probability, we can extend to get node pair sampling probabilities SS_{kh} for node pairs with one node having degree k and the other having degree h, through $g : (k, h) \to SS_{kh}$. In the second step of estimating node sampling probabilities in Gile's (2011) paper, we can add estimating node pair sampling probabilities by

$$g_{SS}((k,h); n, N^i) \approx \frac{U_k \cdot U_h + 1}{M \cdot N_k^i \cdot N_h^i + 1},$$

where $U_k, k = 1, \cdots, K$ is total number of observed units of size k in the M simulations.

1.3.3 Edge Sampling Probabilities in RDS

In a RDS network sample, if two nodes are connected, they must also be connected in the population network. If they are not connected in the RDS network sample, they may still be connected in the population network because of the without replacement sampling property of RDS. Node connections or edges play an important role in network clustering, so reflecting a true connection underlying the RDS network is critical. Therefore, edge sampling is worth considering if we want to get population clustering of nodes from the RDS network.

Due to link-tracing and without replacement sampling, edge sampling probabilities are not uniform in RDS. Ott and Gile (2006) extended the successive sampling approximation to estimate edge sampling probabilities in RDS [Ott and Gile, 2016]. Sampling probabilities are summarized below,

$$\begin{aligned}\text{Node pair sampling probability } SS_{ij} &= P(\text{i,j are sampled}) \\ &= P(\text{i,j are sampled}|Y_{ij}=1) \\ &= P(\text{i,j are sampled}|Y_{ij}=0),\end{aligned}$$

$$\begin{aligned}\text{Edge sampling probability } R_{ij} &= P(\text{i,j are sampled and connected in RDS}|Y_{ij}=1) \\ &= P(\tilde{Y}_{ij}=1|Y_{ij}=1),\end{aligned}$$

P(i,j are sampled and not connected in RDS$|Y_{ij} = 1$)

$= P(\tilde{Y}_{ij} = 0|Y_{ij} = 1)$

$= P$(i,j are sampled$|Y_{ij} = 1) - P$(i,j are sampled and connected in RDS$|Y_{ij} = 1)$

$= P$(i,j are sampled$) - P(\tilde{Y}_{ij} = 1|Y_{ij} = 1)$

$= SS_{ij} - R_{ij}$,

P(i,j are sampled and connected$|Y_{ij} = 0)$

$= P(\tilde{Y}_{ij} = 1|Y_{ij} = 0)$

$= 0$,

P(i,j are sampled and not connected$|Y_{ij} = 0)$

$= P(\tilde{Y}_{ij} = 0|Y_{ij} = 0)$

$= P$(i,j are sampled$|Y_{ij} = 0) - P$(i,j are sampled and connected$|Y_{ij} = 0)$

$= SS_{ij} - 0$

$= SS_{ij}$,

Overall, we can summarize edge sampling probabilities in the contingency table:

Table 1.1. Edge sampling probability

Full Network \ RDS Network	$\tilde{Y}_{ij} = 0$	$\tilde{Y}_{ij} = 1$	(i,j) not sampled
$Y_{ij} = 1$	$(SS_{ij} - R_{ij})P(Y_{ij} = 1)$	$R_{ij}P(Y_{ij} = 1)$	$(1 - SS_{ij})P(Y_{ij} = 1)$
$Y_{ij} = 0$	$SS_{ij}P(Y_{ij} = 0)$	0	$(1 - SS_{ij})P(Y_{ij} = 0)$

1.4 Mixture Model and Weighted log-likehood Mixture Model For Clustering Node Attributed RDS Network Data

Mixture modeling is a widely used clustering method. Gaussian mixtures are used for clustering continuous variables. Stochastic block models are used for clustering social networks. In this paper, we build a mixture model on both node features and network structures by assuming conditional independence between them given the cluster membership.

1.4.1 Mixture model

Assuming conditional independence between the social network and node features given their community labels, we can build a mixture model for the full network:

$$\begin{aligned}
(X_{i1}|z_i = k) &\sim N(\mu_k, \sigma_k), \\
(X_{i2}|z_i = k) &\sim \text{Cat}(\theta_{1k}, \cdots, \theta_{Mk}), \\
(Y_{ij}|z_i = k, z_j = h) &\sim \text{Bernoulli}(\phi_{kh}), \\
z_i &\sim \text{Cat}(\lambda_1, \cdots, \lambda_K),
\end{aligned}$$

where

- $k, h = 1, \cdots, K$, K is the number of latent clusters in the population.
- μ_k, σ_k are the mean and standard deviation of the continuous variable in the k^{th} cluster.
- $\theta_{mk} = P(X_{i2} = m|z_i = k)$ is probability that discrete variable $X_{i2} = m$ given node i in the k^{th} cluster, for any $i = 1, \cdots, N$, M is the number of categories for discrete covariate X_2,

$$\sum_{m=1}^{M} \theta_{mk} = 1.$$

- $\phi_{kh} = P(Y_{ij} = 1|z_i = k, z_j = h)$ is the probability that node i and j are connected given node i in the k^{th} cluster and node j in the h^{th} cluster.

- $\lambda_k = P(z_i = k)$ is the probability that node i is in the k^{th} cluster, for any $i = 1, \cdots, N$,

$$\sum_{k=1}^{K} \lambda_k = 1.$$

If we ignore sampling, a naive approach is to apply the mixture model for the full network directly to the RDS network sample. We set it as our Benchmark Model:

$$\begin{aligned}
(\tilde{X}_{i1}|z_i = k) &\sim N(\mu_k, \sigma_k), \\
(\tilde{X}_{i2}|z_i = k) &\sim \text{Cat}(\theta_{1k}, \cdots, \theta_{Mk}), \\
(\tilde{Y}_{ij}|z_i = k, z_j = h) &\sim \text{Bernoulli}(\phi_{kh}), \\
\tilde{z}_i &\sim \text{Cat}(\lambda_1, \cdots, \lambda_K).
\end{aligned}$$

In this paper, we apply variational EM algorithm to do approximate maximum likelihood inference. This algorithm is applicable even for large networks with thousands of nodes [Daudin et al., 2008].

Given the above mixture model, the variational EM algorithm contains two steps, the variational E-step and the variational M-step. In the E-step of the traditional EM algorithm, we calculate the expectation of the full log-likelihood:

$$\begin{aligned}
Q(\Theta|\Theta^{(t+1)}) &= E_{\tilde{Z}|\tilde{X}_1,\tilde{X}_2,\tilde{Y},\Theta^{(t)}} \log L(\Theta; \tilde{X}_1, \tilde{X}_2, \tilde{Y}, \tilde{Z}) \\
&= \sum_{i=1}^{n} \sum_{k=1}^{K} \pi_{ik} [\log P(\tilde{X}_{i1}|z_{ik}) + \log P(\tilde{X}_{i2}|z_i = k) + \log P(z_i = k)] \\
&+ \frac{1}{2} \sum_{i,j=1, i\neq j}^{n} \sum_{k,h=1}^{K} \pi_{ik,jh} \log P(\tilde{Y}_{ij}|z_i = k, z_j = h),
\end{aligned}$$

where $\pi_{ik} = P(z_i = k|\tilde{X}_1, \tilde{X}_2, \tilde{Y}), \pi_{ik,jh} = P(z_i = k, z_j = h|\tilde{X}_1, \tilde{X}_2, \tilde{Y})$.

It is not easy to calculate π_{ik} and $\pi_{ik,jh}$ because the cluster of node i is not only

associated with nodes connecting with it but is also dependent with other nodes not connecting with it. Considering this, the variational EM [Daudin et al., 2008] is proposed by approximating $P(Z|\tilde{X}_1, \tilde{X}_2, \tilde{Y}, \Theta^{(t)})$ with $R(Z) = \Pi_{i=1}^{n} \tau_{iz_i}$, where $\tau_{ik} \approx P(z_i = k|\tilde{X}_1, \tilde{X}_2, \tilde{Y}, \Theta)$, $\tau_{ik,jh} = \tau_{ik}\tau_{jh} \approx P(z_i = k, z_j = h|\tilde{X}_1, \tilde{X}_2, \tilde{Y}, \Theta)$, and $\sum_{k=1}^{K} \tau_{ik} = 1$ for any $i = 1, \cdots, n$.

- The variational E-step: Modify the E-step of the traditional EM algorithm by approximating π_{ik} with τ_{ik}:

$$\begin{aligned}
\mathcal{Q}(\Theta|\Theta^{(t)}) &= E_{R(Z)} \log L(\Theta; \tilde{X}_1, \tilde{X}_2, \tilde{Y}) - E_{R(Z)} D_{KL}(R(Z)||P(Z|\tilde{X}_1, \tilde{X}_2, \tilde{Y})) \\
&= E_{R(Z)} \log L(\Theta; \tilde{X}_1, \tilde{X}_2, \tilde{Y}, \tilde{Z}) - E_{R(Z)} \log R(Z) \\
&= \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik} [\log P(\tilde{X}_{i1}|z_{ik}) + \log P(\tilde{X}_{i2}|z_i = k) + \log P(z_i = k)] \\
&+ \frac{1}{2} \sum_{i,j=1, i \neq j}^{n} \sum_{k,h=1}^{K} \tau_{ik}\tau_{jh} \log P(\tilde{Y}_{ij}|z_i = k, z_j = h) - \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik} \log \tau_{ik},
\end{aligned}$$

  where $D_{KL}(R(Z)||P(Z|\tilde{X}_1, \tilde{X}_2, \tilde{Y})) = \sum_Z R(Z) \log \frac{R(Z)}{P(Z|\tilde{X}_1, \tilde{X}_2, \tilde{Y})}$ is Kullback–Leibler (KL) divergence from $R(Z)$ to $P(Z|\tilde{X}_1, \tilde{X}_2, \tilde{Y})$, $D_{KL} \geq 0$. The closer it is to 0, the better $R(Z)$ approximates $P(Z|\tilde{X}_1, \tilde{X}_2, \tilde{Y})$.

- The variational M-step: Similar to the M-step in the EM algorithm, in this step, we also update parameters by maximizing the expectation in the variational E-step.

$$\Theta^{(t+1)} = \max_{\theta} \mathcal{Q}(\Theta|\Theta^{(t)}),$$

  Taking the derivative of $\mathcal{Q}(\Theta|\Theta^{(t)})$ for each parameter, in the $(t+1)^{th}$ iteration we update parameters with:

$$
\begin{aligned}
\hat{\tau}_{ik}^{(t+1)} &\propto \hat{\lambda}_k^{(t)} P(\tilde{X}_{i1}|\hat{\mu}_k^{(t)}, \hat{\sigma}_k^{(t)}) P(\tilde{X}_{i2}|\hat{\theta}_{mk}^{(t)}, m = 1, ..., M) \\
&\quad \Pi_{j\neq i}\Pi_{h=1}^{K}[P(\tilde{Y}_{ij}|\hat{\phi}_{kh}^{(t)})], \\
\hat{\lambda}_k^{(t+1)} &= \frac{\sum_{i=1}^{n} \hat{\tau}_{ik}^{(t+1)}}{n}, \\
\hat{\mu}_k^{(t+1)} &= \frac{\sum_i \hat{\tau}_{ik}^{(t+1)} x_{i1}}{\sum_i \hat{\tau}_{ik}^{(t+1)}}, \ \hat{\sigma^2}_k^{(t+1)} = \frac{\sum_i \hat{\tau}_{ik}^{(t+1)} (x_{i1} - \hat{\mu}_k^{(t+1)})^2}{\sum_i \hat{\tau}_{ik}^{(t+1)}}, \\
\hat{\theta}_{mk}^{(t+1)} &= \frac{\sum_i \tau_{ik}^{(t+1)} \mathrm{I}(X_{i2} == m)}{\sum_i \tau_{ik}^{(t+1)}}, \\
\hat{\phi}_{kh}^{(t+1)} &= \frac{\sum_{i\neq j} \tau_{ik}^{(t+1)} \tau_{jh}^{(t+1)} \tilde{Y}_{ij}}{\sum_{i\neq j} \tau_{ik}^{(t+1)} \tau_{jh}^{(t+1)}}.
\end{aligned}
$$

1.4.2 Weighted Log-likelihood Mixture model

As we discussed in Section 1.3, RDS results in non-uniform node and edge sampling probabilities and it's necessary to consider both of them for valid clustering results and parameters estimation. In the paper, we modify the log-likelihood in the mixture model in Section 1.4.1 by adding node and edge weights as the inverse of their sampling probabilities to approximate the log-likelihood in the underlying graph of the RDS network. Based on this weighted log-likelihood we can update parameters and find cluster membership for nodes in the underlying graph. We call this model the weighted log-likelihood mixture model.

Given the full network mixture model, for nodes $i, j = 1, \cdots, N$:

$$
\begin{aligned}
(X_{i1}|z_i = k) &\sim N(\mu_k, \sigma_k), \\
(X_{i2}|z_i = k) &\sim \mathrm{Cat}(\theta_{1k}, \cdots, \theta_{Mk}), \\
(Y_{ij} = 1|z_i = k, z_j = h) &\sim \mathrm{Bernoulli}(\phi_{kh}), \\
Z_i &\sim \mathrm{Cat}(\lambda_1, \cdots, \lambda_K),
\end{aligned}
$$

the variational E-step starts with:

$$\mathcal{Q}_{full}(\Theta|\Theta^{(t)}) = E_{R(Z)}\text{log}L(\Theta; X_1, X_2, Y) - E_{R(Z)}D_{KL}(R(Z)||P(Z|X_1, X_2, Y))$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K}\tau_{ik}[\text{log}P(X_{i_1}|z_{ik}) + \text{log}P(X_{i_2}|z_i = k) + \text{log}P(z_i = k)] \quad \cdots\cdots \mathbf{A}$$

$$+ \frac{1}{2}\sum_{i,j=1,i\neq j}^{N}\sum_{k,h=1}^{K}\tau_{ik}\tau_{jh}[Y_{ij}\text{log}P(Y_{ij} = 1|z_i = k, z_j = h)] \quad \cdots\cdots \mathbf{B}$$

$$+ \frac{1}{2}\sum_{i,j=1,i\neq j}^{N}\sum_{k,h=1}^{K}\tau_{ik}\tau_{jh}[(1 - Y_{ij})\text{log}P(Y_{ij} = 0|z_i = k, z_j = h)] \quad \cdots\cdots \mathbf{C}$$

$$- \sum_{i=1}^{N}\sum_{k=1}^{K}\tau_{ik}\text{log}\tau_{ik}. \quad \cdots\cdots \mathbf{D}$$

In $\mathcal{Q}_{full}(\Theta|\Theta^{(t)})$, the full network log-likelihood contains four parts, part A is the log-likelihood of node features, part B is the log-likelihood of two connected nodes, part C is the log-likelihood of two nodes not connected, and part D is the penalty term from the KL divergence.

Based on node sampling probabilities $S = \{S_i, i = 1, \cdots, n\}$, part A can be approximated by weighted log-likelihood from node features in the RDS network:

$$\text{part A} \approx \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}\frac{1}{S_i}[\text{log}P(X_{i_1}|z_{ik}) + \text{log}P(X_{i_2}|z_i = k) + \text{log}P(z_i = k)],$$

Part D can be approximated using node sampling probabilities as well:

$$\text{part D} \approx \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}\frac{1}{S_i}\tau_{ik}\text{log}\tau_{ik},$$

Since all edges in the RDS network are sampled from edges in the full network with sampling probabilities $R = R_{ij,i,j=1,\cdots,n}$ and $R_{ij} = P(\tilde{Y}_{ij} = 1|Y_{ij} = 1)$, part B can be approximated by weighted log-likelihood of edges in the RDS network:

$$\text{part B} \approx \sum_{i,j=1,i\neq j}^{n}\sum_{k,h=1}^{K}\tau_{ik}\tau_{jh}\frac{1}{R_{ij}}[\tilde{Y}_{ij}\text{log}P(Y_{ij} = 1|z_i = k, z_j = h)].$$

Two nodes not connected in the RDS network may still be connected in the full network. To approximate part C, we first need to estimate the probability that unconnected nodes in the sample are also not connected in the full network, denoted by $P(Y_{ij}=0|\tilde{Y}_{ij}=0)$:

$$
\begin{aligned}
&P(Y_{ij}=0|\tilde{Y}_{ij}=0)\\
&=\frac{P(Y_{ij}=0,\tilde{Y}_{ij}=0)}{P(\tilde{Y}_{ij}=0)}\\
&=\frac{P(Y_{ij}=0,\tilde{Y}_{ij}=0)}{P(Y_{ij}=0,\tilde{Y}_{ij}=0)+P(Y_{ij}=1,\tilde{Y}_{ij}=0)}\\
&=\frac{P(\tilde{Y}_{ij}=0|Y_{ij}=0)P(Y_{ij}=0)}{P(\tilde{Y}_{ij}=0|Y_{ij}=0)P(Y_{ij}=0)+P(\tilde{Y}_{ij}=0|Y_{ij}=1)P(Y_{ij}=1)}\\
&=\frac{SS_{ij}P(Y_{ij}=0)}{SS_{ij}P(Y_{ij}=0)+(SS_{ij}-R_{ij})P(Y_{ij}=1)}\\
&=\frac{SS_{ij}P(Y_{ij}=0)}{SS_{ij}-R_{ij}P(Y_{ij}=1)}.
\end{aligned}
$$

Assume sampling probabilities are independent given cluster labels. We have $P(Y_{ij}=0|\tilde{Y}_{ij}=0,z_i=k,z_j=h)=\frac{SS_{ij}P(Y_{ij}=0|z_i=k,z_j=h)}{SS_{ij}-R_{ij}P(Y_{ij}=1|z_i=k,z_j=h)}$. Meanwhile, from Table 1.1 we also have sampling probabilities of two unconnected nodes, $P(\tilde{Y}_{ij}=0|Y_{ij}=0)=SS_{ij}$. Then we can approximate part C by:

$$
\begin{aligned}
&\text{part C}\\
&\approx\sum_{i,j=1,i\neq j}^{n}\sum_{k,h=1}^{K}\tau_{ik}\tau_{jh}\frac{1}{SS_{ij}}[P(Y_{ij}=0|\tilde{Y}_{ij}=0,z_i=k,z_j=h)(1-\tilde{Y}_{ij})\log P(Y_{ij}=0|z_i=k,z_j=h)]\\
&=\sum_{i,j=1,i\neq j}^{n}\sum_{k,h=1}^{K}\tau_{ik}\tau_{jh}\frac{1}{SS_{ij}}[\frac{SS_{ij}P(Y_{ij}=0|z_i=k,z_j=h)}{SS_{ij}-R_{ij}P(Y_{ij}=1|z_i=k,z_j=h)}(1-\tilde{Y}_{ij})\log P(Y_{ij}=0|z_i=k,z_j=h)]\\
&=\sum_{i,j=1,i\neq j}^{n}\sum_{k,h=1}^{K}\tau_{ik}\tau_{jh}[(1-\tilde{Y}_{ij})\frac{P(Y_{ij}=0|z_i=k,z_j=h)\log P(Y_{ij}=0|z_i=k,z_j=h)}{SS_{ij}-R_{ij}P(Y_{ij}=1|z_i=k,z_j=h)}].
\end{aligned}
$$

With all these weights, we get the full log-likelihood approximation for the variational E-step:

$$\mathcal{Q}_{full}(\Theta|\Theta^{(t)}) = \text{part A + part B + part C - part D}$$

$$\approx \mathcal{Q}_w(\Theta|\Theta^{(t)})$$

$$= \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}\frac{1}{S_i}[\log P(X_{i_1}|z_{ik}) + \log P(X_{i_2}|z_i = k) + \log P(z_i = k)] \quad \cdots\cdots \textbf{w-A}$$

$$+ \frac{1}{2}\sum_{i,j=1,i\neq j}^{n}\sum_{k,h=1}^{K}\tau_{ik}\tau_{jh}\frac{1}{R_{ij}}[\tilde{Y}_{ij}\log P(Y_{ij} = 1|z_i = k, z_j = h)] \quad \cdots\cdots \textbf{w-B}$$

$$+ \frac{1}{2}\sum_{i,j=1,i\neq j}^{n}\sum_{k,h=1}^{K}\tau_{ik}\tau_{jh}[(1-\tilde{Y}_{ij})\frac{P(Y_{ij} = 0|z_i = k, z_j = h)\log P(Y_{ij} = 0|z_i = k, z_j = h)}{SS_{ij} - R_{ij}P(Y_{ij} = 1|z_i = k, z_j = h)}] \quad \cdots\cdots \textbf{w-C}$$

$$- \sum_{i=1}^{n}\sum_{k=1}^{K}\frac{1}{S_i}\tau_{ik}\log\tau_{ik} \quad \cdots\cdots \textbf{w-D}$$

$$= \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}\frac{1}{S_i}[\log(\frac{1}{2\sigma_k\sqrt{2\pi}}) - \frac{(x_{i1}-\mu_k)^2}{2\sigma_k^2} + \log\sum_{m=1}^{M} I\{x_{i2} == m\}\theta_{mk} + \log\lambda_k]$$

$$+ \frac{1}{2}\sum_{i,j=1,\cdots,n;i\neq j}\sum_{k,h=1}^{K}\tau_{ik}\tau_{jh}[\tilde{Y}_{ij}\frac{\log\phi_{kh}}{R_{ij}} + (1-\tilde{Y}_{ij})(1-\phi_{kh})\frac{\log(1-\phi_{kh})}{SS_{ij} - R_{ij}\phi_{kh}}]$$

$$- \sum_{i=1}^{n}\sum_{k=1}^{K}\frac{1}{S_i}\tau_{ik}\log\tau_{ik}.$$

In the variational M-step, we update parameters by maximizing the weighted log-likelihood in the variational E-step:

$$\Theta_w^{(t+1)} = \max_{\theta}\mathcal{Q}_w(\Theta|\Theta^{(t)}),$$

$$\hat{\tau}_{ik}^{(t+1)} \propto [\hat{\lambda}_k^{(t)} P(X_{i1}|\hat{\mu}_k^{(t)}, \hat{\sigma}_k^{(t)}) P(X_{i2}|\hat{\theta}_{mk}^{(t)}, m = 1, ..., M)] \Pi_{j\neq i} \Pi_{h=1}^{K} [P(\tilde{Y}_{ij}|\hat{\phi}_{kh}^{(t)})]^{\tau_{jh}^{(t)} S_i}$$

$$= [\hat{\lambda}_k^{(t)} P(X_{i1}|\hat{\mu}_k^{(t)}, \hat{\sigma}_k^{(t)}) P(X_{i2}|\hat{\theta}_{mk}^{(t)}, m = 1, ..., M)]$$

$$\Pi_{j\neq i} \Pi_{h=1}^{K} [(\hat{\phi}_{kh}^{(t)})^{\tilde{Y}_{ij}/R_{ij}} (1 - \hat{\phi}_{kh}^{(t)})^{(1-\tilde{Y}_{ij})(1-\hat{\phi}_{kh}^{(t)})/(SS_{ij} - R_{ij}\hat{\phi}_{kh}^{(t)})}]^{\tau_{jh}^{(t)} S_i},$$

$$\hat{\lambda}_k^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{\tau}_{ik}^{(t+1)}/S_i}{n},$$

$$\hat{\mu}_k^{(t+1)} = \frac{\sum_i \hat{\tau}_{ik}^{(t+1)}/S_i x_{i1}}{\sum_i \hat{\tau}_{ik}^{(t+1)}/S_i}, \ \hat{\sigma^2}_k^{(t+1)} = \frac{\sum_i \hat{\tau}_{ik}^{(t+1)}/S_i (x_{i1} - \hat{\mu}_k^{(t+1)})^2}{\sum_i \hat{\tau}_{ik}^{(t+1)}/S_i},$$

$$\hat{\theta}_{mk}^{(t+1)} = \frac{\sum_i \tau_{ik}^{(t+1)}/S_i \mathrm{I}(X_{i2} == m)}{\sum_i \tau_{ik}^{(t+1)}/S_i},$$

$$\frac{\partial \mathcal{Q}_w}{\partial \phi_{k,h}^{(t+1)}} = \sum_{i,j=1\cdots,n; i\neq j} \tau_{ik}^{(t+1)} \tau_{jh}^{(t+1)} [\frac{\tilde{Y}_{ij}}{R_{ij}\phi_{k,h}^{(t+1)}} + (1 - \tilde{Y}_{ij}) \frac{(R_{ij} - S_{ij})\log(1 - \phi_{k,h}^{(t+1)}) - (S_{ij} - R_{ij}\phi_{k,h}^{(t+1)})}{(S_{ij} - R_{ij}\phi_{k,h}^{(t+1)})^2}].$$

Set $\frac{\partial \mathcal{Q}_w}{\partial} \phi_{k,h}^{(t+1)} = 0,$ and we can solve for $\phi_{k,h}^{(t+1)}$ using Newton-Raphson iteration.

1.4.3 Weighted log-likelihood mixture model with tuning parameter

In the weighted log-likelihood mixture model, the full log-likelihood approximation is

$$\mathcal{Q}_w(\Theta|\Theta^{(t)}) = \text{part } \textbf{w-A} + \text{part } \textbf{w-B} + \text{part } \textbf{w-C} - \text{part } \textbf{w-D},$$

where part **w-A** is the weighted log-likelihood from covariates, and (part **w-B** + part **w-C**) is the weighted log-likelihood from the network structure. In this section, we add a tuning parameter to balance contribution of the network structure and covariates, where

$$\mathcal{Q}_{w;\alpha}(\Theta|\Theta^{(t)}) = \text{part } \textbf{w-A} + \alpha * (\text{part } \textbf{w-B} + \text{part } \textbf{w-C}) - \text{part } \textbf{w-D}.$$

When $\alpha = 0$, the clustering is based on covariates only, when $\alpha = 1$, $\mathcal{Q}_{w;\alpha}(\Theta|\Theta^{(t)}) = \mathcal{Q}_w(\Theta|\Theta^{(t)})$, larger α, contribution of the network structure is larger. This is similar

to spectral clustering with covariates ([Binkiewicz et al., 2017][Shiga et al., 2007]). Adding the tuning parameter α only effects the cluster memberships of nodes.

$$\begin{aligned}\hat{\tau}_{ik;\alpha}^{(t+1)} &\propto [\hat{\lambda}_k^{(t)} P(X_{i1}|\hat{\mu}_k^{(t)}, \hat{\sigma}_k^{(t)}) P(X_{i2}|\hat{\theta}_{mk}^{(t)}, m = 1, ..., M)] \Pi_{j\neq i} \Pi_{h=1}^{K} [P(\tilde{Y}_{ij}|\hat{\phi}_{kh}^{(t)})]^{\alpha \tau_{jh}^{(t)} S_i} \\ &= [\hat{\lambda}_k^{(t)} P(X_{i1}|\hat{\mu}_k^{(t)}, \hat{\sigma}_k^{(t)}) P(X_{i2}|\hat{\theta}_{mk}^{(t)}, m = 1, ..., M)] \\ &\quad \Pi_{j\neq i} \Pi_{h=1}^{K} [(\hat{\phi}_{kh}^{(t)})^{\tilde{Y}_{ij}/R_{ij}} (1 - \hat{\phi}_{kh}^{(t)})^{(1-\tilde{Y}_{ij})(1-\hat{\phi}_{kh}^{(t)})/(SS_{ij} - R_{ij}\hat{\phi}_{kh}^{(t)})}]^{\alpha \tau_{jh}^{(t)} S_i}.\end{aligned}$$

Updates for all the other parameters are the same as those of the mixture model with weighted log-likelihood in Section 1.4.2.

1.5 Clustering evaluation and tuning parameter selection

When both node features and network have communities, we need to decide the tuning parameter value α to get desired clusters. To check if the clustering is what we want for the network with node attributes, we need to evaluate the clustering quality in terms of network structure and in terms of node attributes. Then the tuning parameter α can be chosen based on clustering evaluation metrics.

Evaluating the quality of clustering algorithms is typically in two ways, internal evaluation and external evaluation. The internal evaluation uses a score to summarize clustering quality and the external evaluation compares a known classification in the data with the clustering got from the model. Popular internal evaluation metrics for network clustering include modularity, conductance, coverage [Newman, 2006][Kobourov et al., 2014][Schaeffer, 2007] and common internal evaluations for attributes are Silhouette index, Dunn's indices, Davies-Bouldin index, etc [Rousseeuw, 1987] [Dunn†, 1974][Davies and Bouldin, 1979]. Popular external clustering evalution metrics include purity, entropy, normalized mutual information, F measure, Rand index [Larsen and Aone, 1999][Strehl and Ghosh, 2003][Rendón et al., 2011]. In this paper, we focus on modularity for the network clustering evaluation and normalized

mutual information for evaluating clustering of node features. For both of them, larger value indicates better clustering, can be used to compare different clustering algorithms and choose number of clusters for the clustering algorithm. In this paper, we use these two clustering evaluation metrics to determine tuning parameter α as well.

Modularity evaluates the strength of division of a network into clusters. Assume network G is clustered into K clusters with vertex sets $C = \{C_1, \cdots, C_K\}$, then the modularity Q(C) is

$$\mathrm{Q}(C) = \sum_{k=1}^{K} e_{kk} - a_k^2,$$

where $E_{kl} = \sum_{i \neq j}(\tilde{Y}_{ij}|z_i = K, z_j = l)$, $e_{kk} = \frac{E_{kk}}{\sum_{k,l} E_{kl}}$ is fraction of edges with both vertices in cluster k. $a_k = \frac{\sum_l E_{kl}}{\sum_{k,l} E_{kl}}$ is the fraction of ends of edges incident to cluster k, a_k^2 is the expected fraction of edges with both vertices in cluster k if edges were randomly distributed. The range of modularity is [-1, 1]. Higher modularity means more edges are within clusters than between clusters.

Mutual Information measures mutual dependence between two random variables, X and C:

$$I(X, C) = \sum_x \sum_c p(x, c) \log \frac{p(x, c)}{p(x)p(c)}.$$

The Normalized Mutual Information (NMI) is:

$$\mathrm{NMI}(X, C) = \frac{I(X, C)}{\sqrt{H(X)H(C)}},$$

where $\mathrm{NMI}(X, C) \in [0, 1]$, $\mathrm{NMI}(X, C) = 0$ indicates X and C are independent, and larger NMI means better clustering. $H(X) = -\sum_x p(x) \log p(x)$ is entropy of X. It is also true that $I(X, C) = H(X) + H(C) - H(X, C) = H(X) - H(X|C) = H(C) - H(C|X)$.

In our dataset, we have continuous and discrete node features. To calculate the NMI for all features, we have three steps. Step 1, we cut the continuous variables into discrete variables. Step 2, we calculate NMI for each node feature. Step 3, we take average of NMIs got in step 2 as our final NMI for node features.

Since RDS gives an incomplete social network, we don't know e_{kk} and a_k for the full network. Fortunately, we can estimate them through sampling weights,

$$\hat{e}_{kk} = \frac{\hat{E}_{kk}}{\sum_{k,l} \hat{E}_{kl}},$$

$$\hat{a}_k = \frac{\hat{E}_{kk} + \sum_{l \neq k} \hat{E}_{kl}}{\sum_{k,l} \hat{E}_{kl}},$$

where $\hat{E}_{kl} = \sum_{i \neq j} \frac{\tilde{Y}_{ij}}{R_{ij}} I(z_i = k, z_j = l)$, then $\hat{Q}(C) = \sum_k \hat{e}_{kk} - (\hat{a}_k)^2$.

We can also estimate $\mathrm{NMI}(X, C)$ for the full network $\hat{\mathrm{NMI}}(X, C) = \frac{\hat{I}(X,C)}{\sqrt{\hat{H}(X)\hat{H}(C)}}$ with

$$\hat{H}(X) = -\sum_x \hat{p}(x) \log \hat{p}(x), \ \hat{p}(x) = \frac{\sum_i I(X_i = x)/S_i}{\sum_i 1/S_i},$$

similarly, we can estimate $\hat{H}(C)$ and $\hat{H}(X|C)$.

By looking at how the clustering evaluation metrics, normalized mutual information $\hat{NMI}$ and modularity $\hat{Q}$ change with different values of α, we can decide the best tuning parameter α.

1.6 Simulation Study

In this section, we compare clustering performance using the mixture model with and without weighted log-likelihood and with different values of tuning parameters in four different cases. For each case, we simulate 100 full networks with one continuous variable and one categorical variable, then we sample a RDS network from each full network. Finally, we apply the candidate mixture models on those RDS networks. A

Table 1.2. Parameters for different simulation cases. ϕ is parameter for the network connection, μ is mean of the continuous variable, θ is parameter for the categorical variable, λ is parameter for the cluster membership.

	ϕ	μ	θ	λ
Case I: Both separate well	$\phi = \begin{bmatrix} 0.1 & 0.02 \\ 0.02 & 0.2 \end{bmatrix}$	[-2,2]	$\theta = \begin{bmatrix} 0.8 & 0.4 \\ 0.2 & 0.6 \end{bmatrix}$	1/3
Case II: Features separate well, Network does not	$\phi = \begin{bmatrix} 0.05 & 0.05 \\ 0.05 & 0.05 \end{bmatrix}$	[-2,2]	$\theta = \begin{bmatrix} 0.8 & 0.4 \\ 0.2 & 0.6 \end{bmatrix}$	1/3
Case III: Network separates well, Features do not	$\phi = \begin{bmatrix} 0.1 & 0.02 \\ 0.02 & 0.2 \end{bmatrix}$	[0,0]	$\theta = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$	1/3
Case IV: Both do not separate well	$\phi = \begin{bmatrix} 0.05 & 0.05 \\ 0.05 & 0.05 \end{bmatrix}$	[0,0]	$\theta = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$	1/3

summary of the different cases is in Table 1.2.

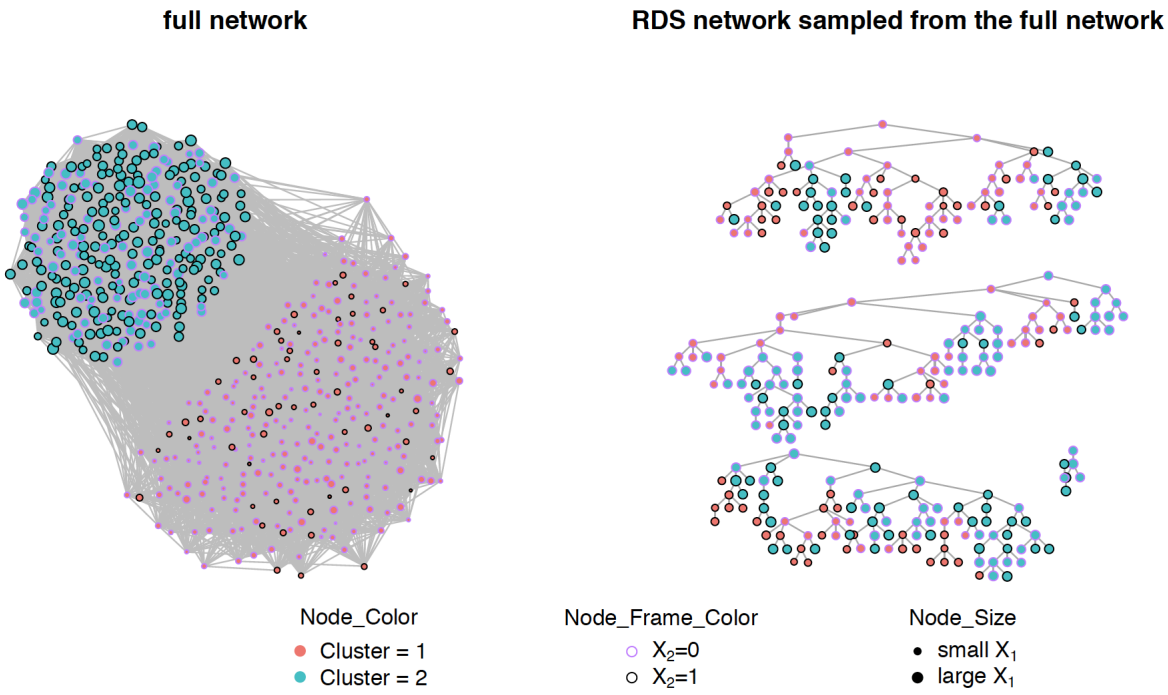
The full networks are generated by:

$$G \sim \text{SBM}(N = 600, \phi = \phi, \text{block.size} = c(200, 400)),$$
$$(X_{i1}|z_i = k) \sim N(\mu_k, 1),\ k = 1, 2; i = 1, \cdots, 600,$$
$$(X_{i2}|z_i = k) \sim \text{Cat}(\theta_{1k}, \theta_{2k}),\ k = 1, 2; i = 1, \cdots, 600,$$

where $\text{SBM}(N = 600, \phi = \phi, \text{block.size} = c(200, 400))$ is a stochastic block model with size $N = 600$, two blocks or communities of size 200 and 400. The social connection parameter within and between blocks is denoted by ϕ.

The RDS network sample is obtained by RDS sampling from the complete network G with 5 seeds for $n = 300$, 3 seeds for $n = 100$ and 3 coupons for each node. The distribution of number of recruitments for each sample is $[0, 1, 2, 3]$ with probabilities of $[0.1, 0.2, 0.3, 0.4]$ respectively. One example of the full network and its sampled RDS network is plotted in Figure 1.1. In both networks, nodes are colored by their cluster labels, frame colored by their categorical values and sized by their continuous variable values. In this full network, both features and network structure separate well. We can see from the full and RDS network that people in the same cluster have similar node features and are more likely to connect. In the RDS network sample,

Figure 1.1. Full network and one RDS network sampled from it



nodes in different trees may be in the same cluster even though they come from different seeds in the RDS network sample and are not connected visually. To detect this latent clustering truth, node features play an important role. From the RDS network sample, we can also see that sampled degree for all nodes is at most 4 which is the maximum number of coupons each person can distribute plus 1.

In the simulation study, we take a full network of size $N = 600$ and consider two types of its RDS sample with node samples of $n = 300$ and $n = 100$. Figures 1.2 to 1.5 are plots of modularity and NMI with different values of tuning parameter α, based on which we can determine the best tuning parameter for each RDS sample. Figures 1.6 to 1.9 are boxplots for parameter estimation, number of mis-clusterings, modularity and normalized mutual information by using five different models for the four different cases when $n = 300$ and Figures 1.10 to 1.13 are boxplots for RDS sample with $n = 100$. Five different models we use are mixture model without

weighting and $\alpha = 0$ (noW-alpha=0), mixture model without weighting and $\alpha = 1$ (noW-alpha=1), mixture model with weighting and $\alpha = 0$ (W-alpha=0), mixture model with weighting and $\alpha = \alpha$ (W-alpha=0.1 for case I and II, W-alpha=0.4 for case III and IV) and mixture model with weighting and $\alpha = 1$ (W-alpha=1).

1.6.1 Tuning parameter selection

The tuning parameter α controls contribution of network structure to the node cluster membership as we discussed in Section 1.4.3. The following talks about how tuning parameters influence modularity, NMI and clustering of the attributed RDS sample in the four different cases.

For case I (Both network and node features separate well), we can see from Figure 1.2 that NMI gets its largest value when $\alpha = 0$ and it decreases with increasing α. The modularity increases with increasing α. Meanwhile, modularity is not too small when $\alpha = 0$ which indicates that communities found by covariates only can reflect some community structures in the network. This captures the property that network and node features have the same communities in case I. If we care more about separation of node features, we can choose $\alpha = 0$. If we want a clustering with better network clustering, we can choose $\alpha = 0.1$ because the modularity increases obviously when α changes from 0 to 0.1 and NMI decreases more than the increase of modularity when α changes from 0.1 to 0.2. Therefore, we use W-alpha=0.1 as our best model for the weighted mixture model with tuning parameter in case I. For the mixture model without weighting, modularity increases almost linearly, NMI decreases slowly then faster, so we choose $\alpha = 1$ where NMI starts to drop faster as the tuning parameter value for the unweighted model (noW-alpha=1).

For case II (Features separate well, network does not) in Figure 1.3, when $\alpha = 0$ and $\alpha = 0.1$, the NMI is much larger and the modularity is very close to 0. This tells us that clustering of node features are not consistent with clustering of network

Figure 1.2. Plot of Modularity and NMI vs Tuning parameter α in mixture model w/ and w/o weights for case I (both separate well)

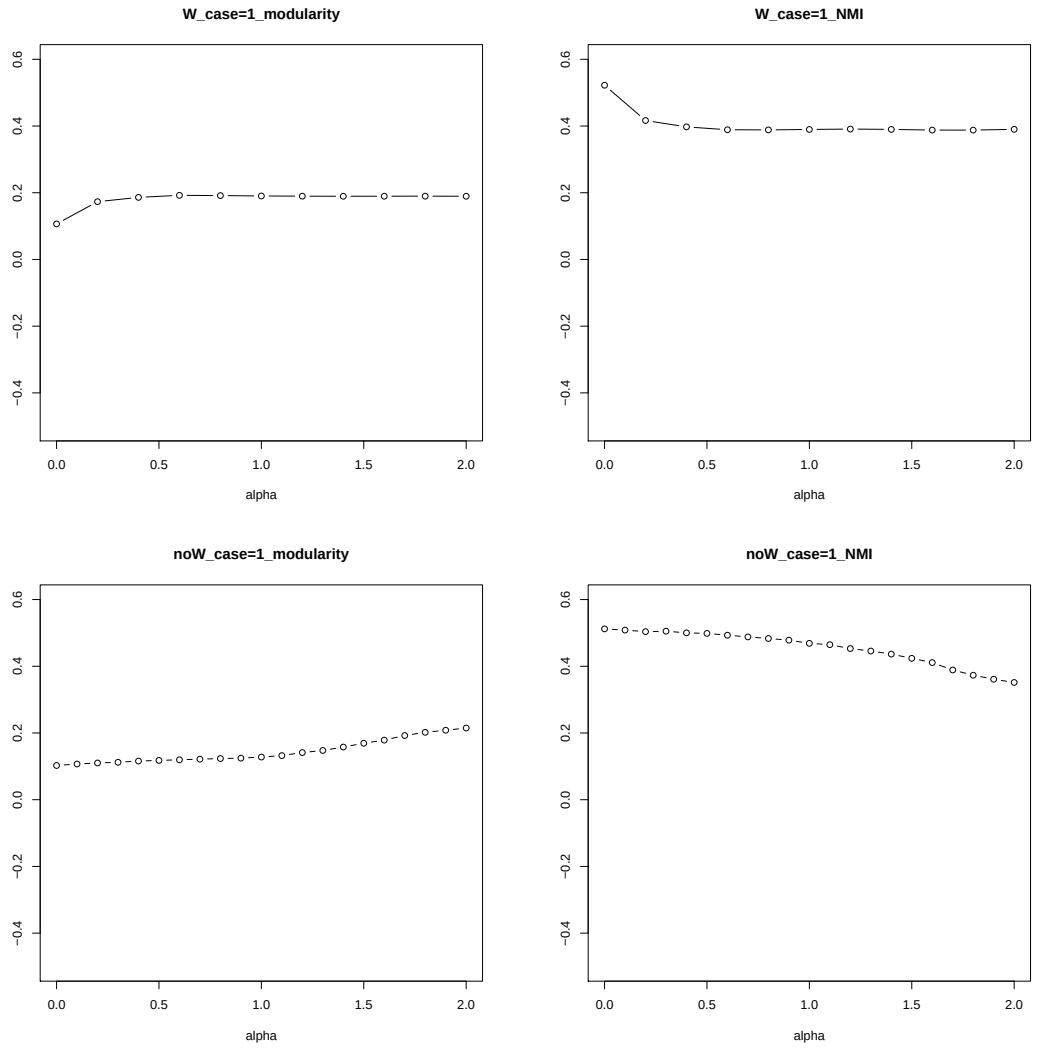


Figure 1.3. Plot of Modularity and NMI vs Tuning parameter α in mixture model w/ and w/o weights for case II (features separate well)

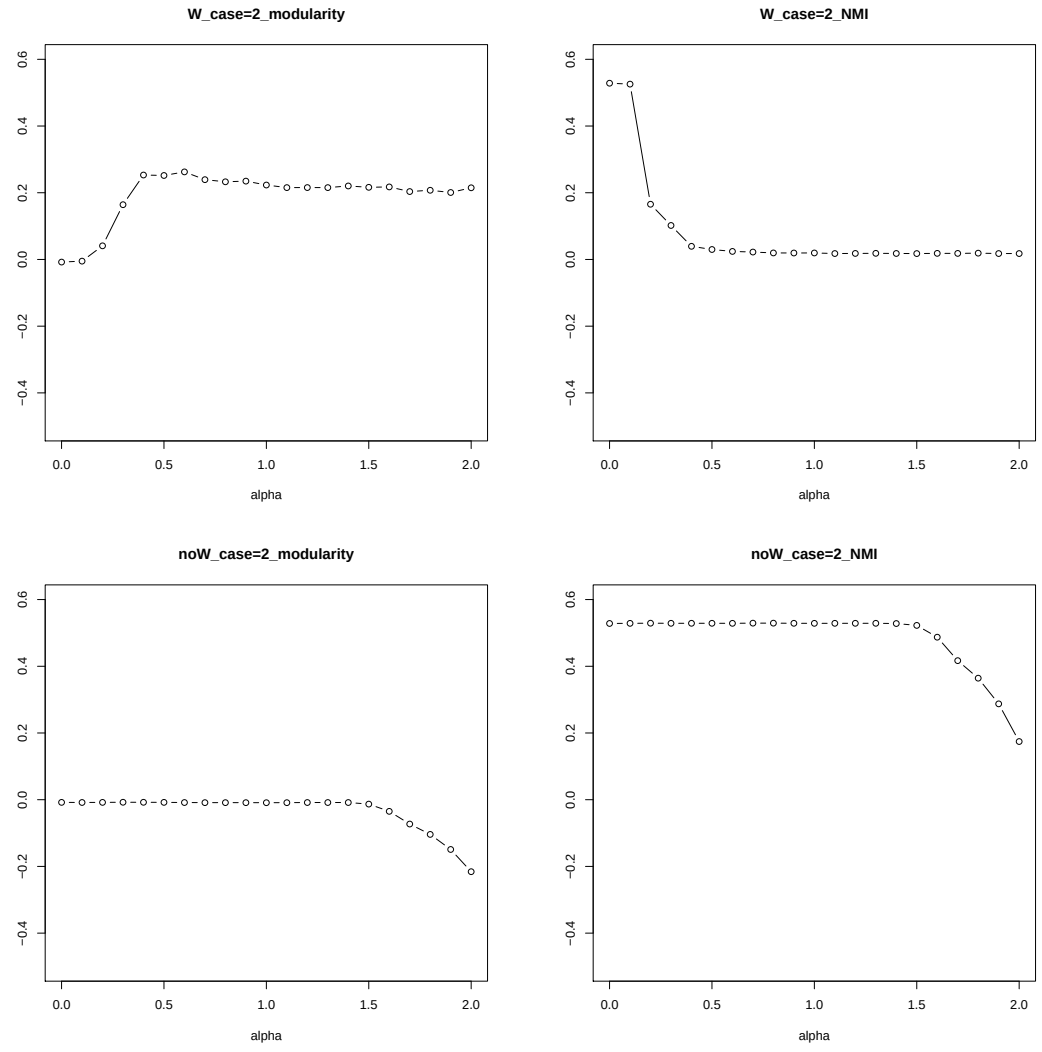
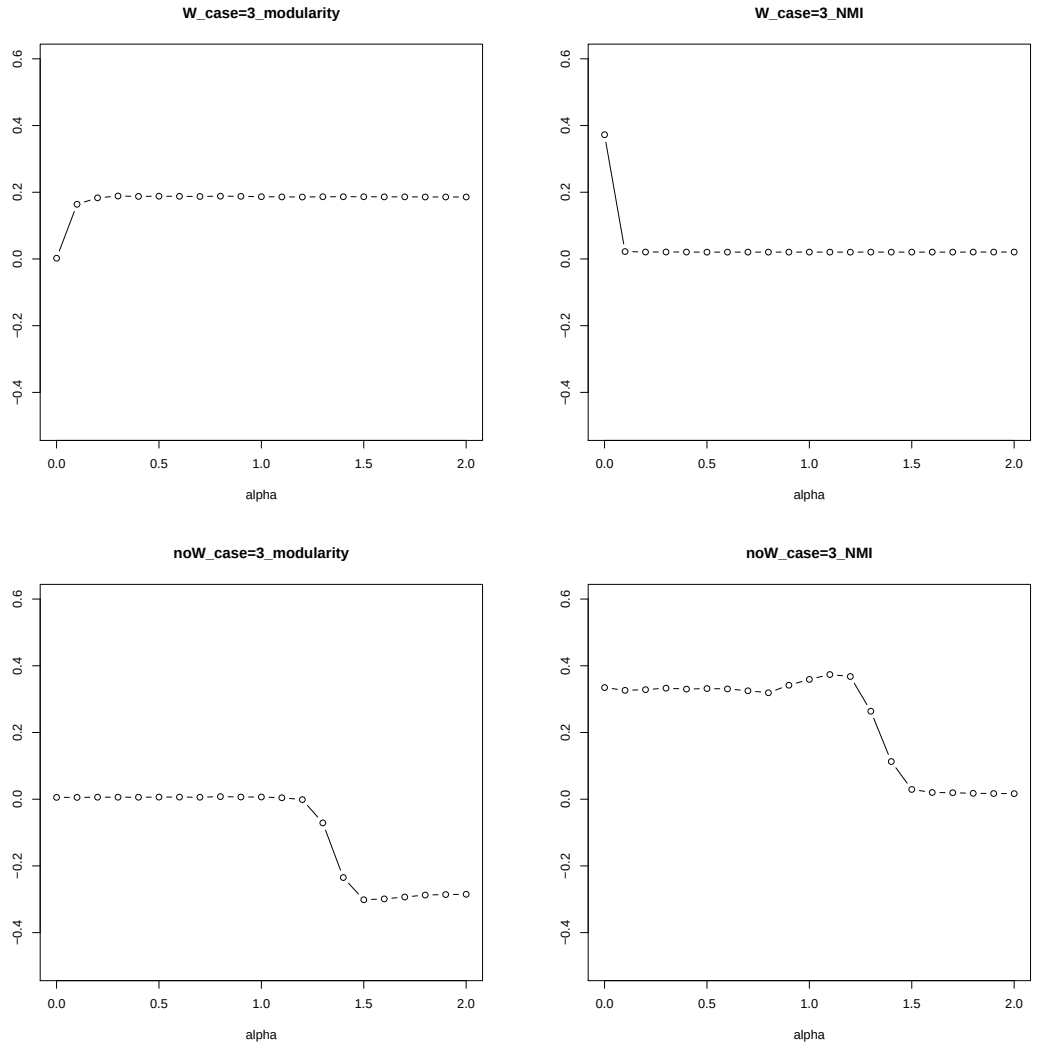


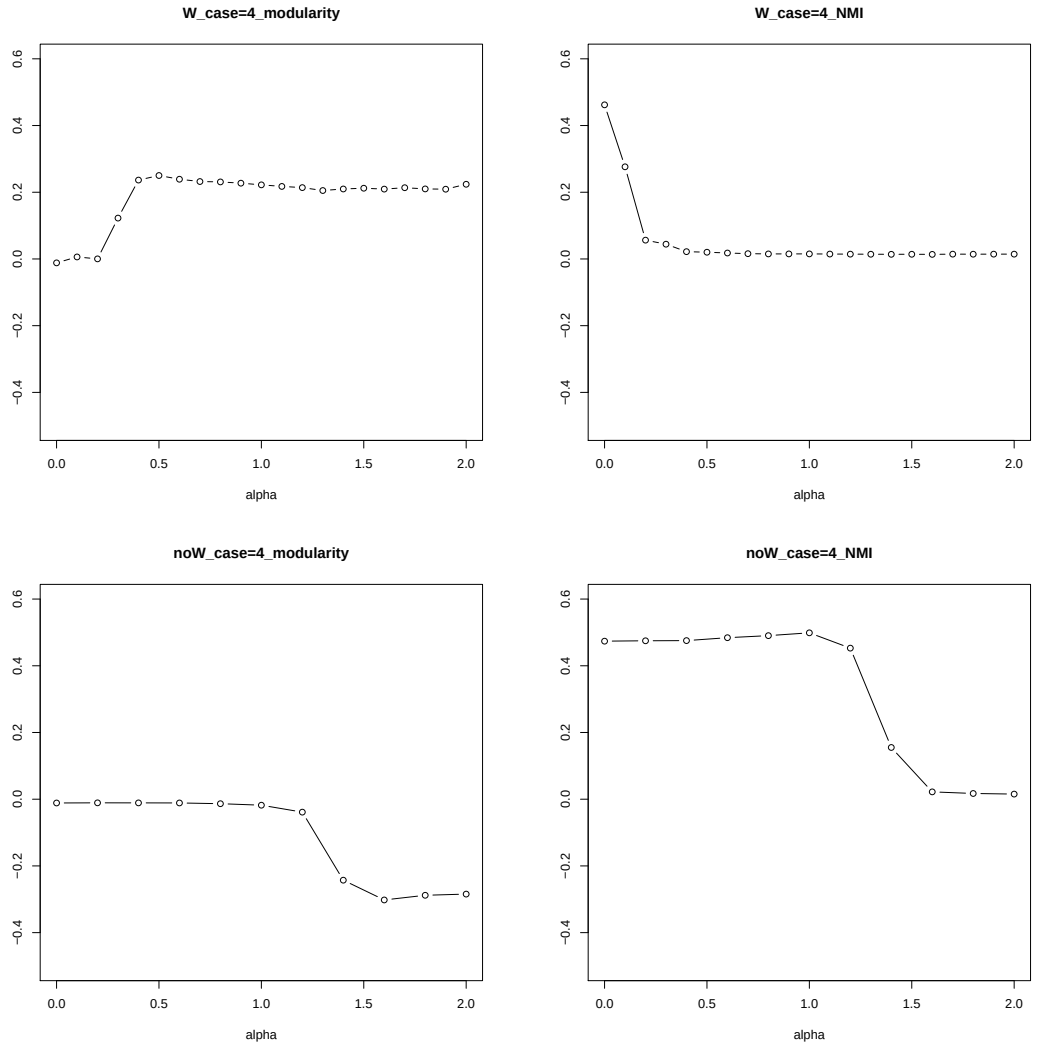
Figure 1.4. Plot of Modularity and NMI vs Tuning parameter α in mixture model w/ and w/o weights for case III (network separate well)



structures or the network structure does not separate well. Since if α changes from 0.1 to 0.2, NMI decreases obviously and modularity is still small, we conclude that the network does not have obvious communities and choose $\alpha = 0.1$ as our preferred tuning parameter value for this case. For the unweighted model, both modularity and NMI decrease with increasing α, especially when it's greater than 1. This further tells us that the network does not have communities.

For case III (Network separates well, node features don't) in Figure 1.4, when $\alpha = 0$, it's very similar with the plot for the case II Figure 1.3. However, when α increases

Figure 1.5. Plot of Modularity and NMI vs Tuning parameter α in mixture model w/ and w/o weights for case IV (both do not separate well)



from 0 to 0.1, the modularity increases and the NMI decreases quickly. This tells us that clustering exists within the network structure. We choose $\alpha = 0.4$ for case III because it is within the range where both the modularity increases and NMI decreases very slowly. Similar to case I and case II, the unweighted model in case III also choose α to be 0 and 1.

For case IV (both do not separate well), plots in Figure 1.5 are similar to plots in Figure 1.4 in case III. We again choose $\alpha = 0.4$. However, when α increases, modularity does not increase as quickly as in case III and it's not a large value. This

tells us that this network data does not separate well. The modularity and NMI vs α plots are very similar for RDS sample with size $n = 300$ and $n = 100$. Therefore, we use the same choosen α for the same case in those two types of RDS sample.

1.6.2 Clustering evaluation and parameter estimation

For case I (both separate well), from Figure 1.6 we can see that all five models are not bad in clustering and parameter estimation, this makes sense because both network and features separate well in case I. However, the mixture model with weights are better in parameter estimation, especially in the estimation of latent class proportions λ and network connections ϕ. The model with our choosen $\alpha = 0.1$ (W-alpha=0) has smaller number of mis-clusterings than the model with $\alpha = 1$ (W-alpha=1) because it has larger NMI and similar modularity, as we can see from the bottom two plots. The model noW-alpha=1 has the smallest number of mis-clusterings, because it has similar NMI and larger modularity as we can see from Figure 1.2. Therefore, modularity and NMI reflect clustering quality. We can use them to get some idea about clustering even though we don't have true labels in real data.

For case II (only features separate well), Figure 1.7 shows that when node features separate well, but the network does not, all models except the weighted model with $\alpha = 1$, get pretty good clustering results. Also, the model with weighting gives better network structure parameter estimation ϕ. This case tells us that when only node features are important and have obvious communities, the tuning parameter is essential to avoid overfitting of the noisy network structure.

Figure 1.8 are the result for the third case, only the network structure separates well. We can still see models with weighting give better parameter estimates. In this case, we can also see that the models with weights and larger tuning parameter (W-alpha=0.4 and W-alpha=1) have better clustering results. This is also consistent with the modularity and NMI plots in Figure 1.4. This case tells us that using a

Figure 1.6. Parameter estimations, Number of mis-clusterings, Modularity and NMI by using different models for case I (both separate well) when n=300

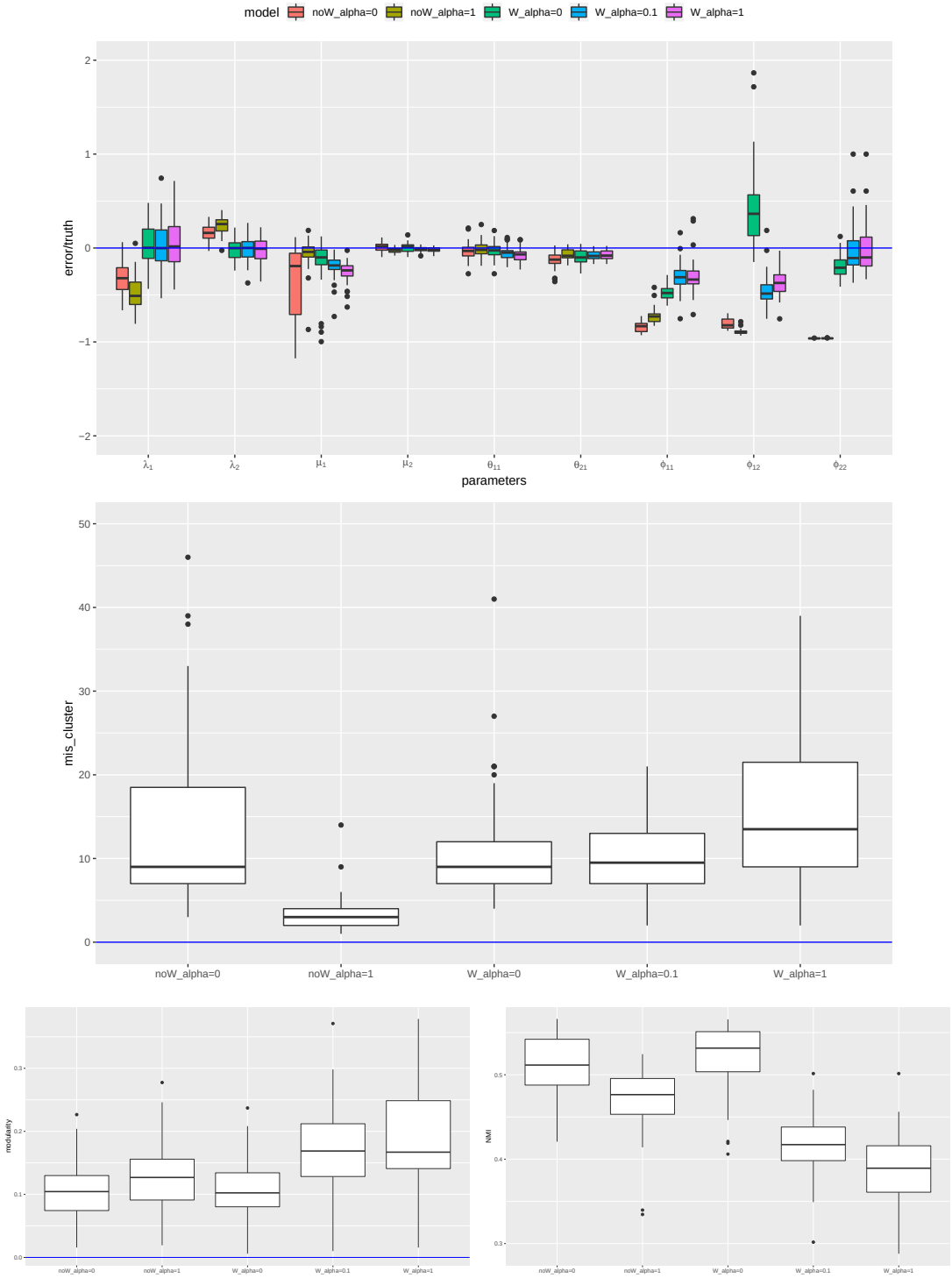


Figure 1.7. Parameter estimations, Number of mis-clusterings, Modularity and NMI by using different models for case II (features separate well) when n=300

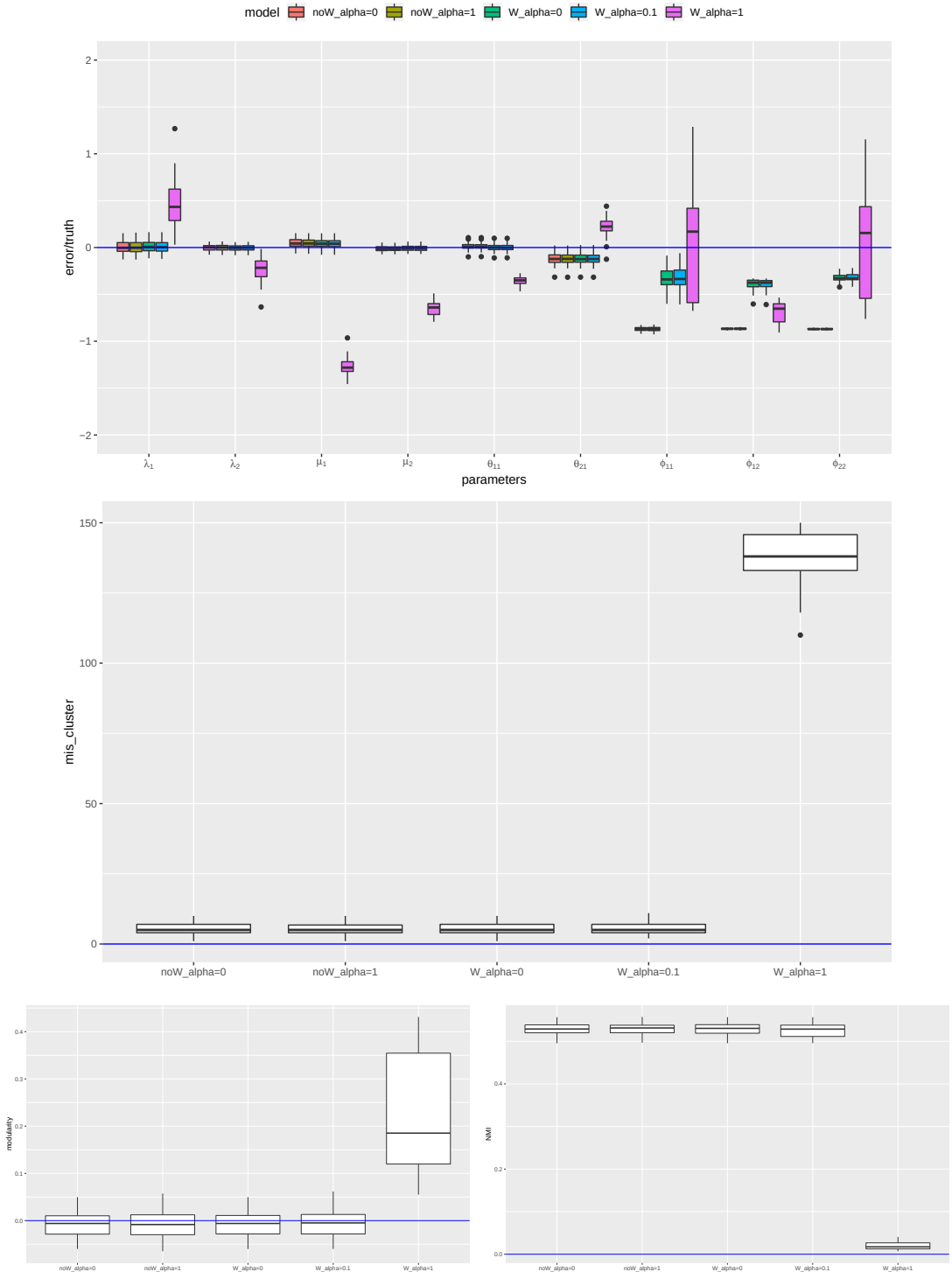
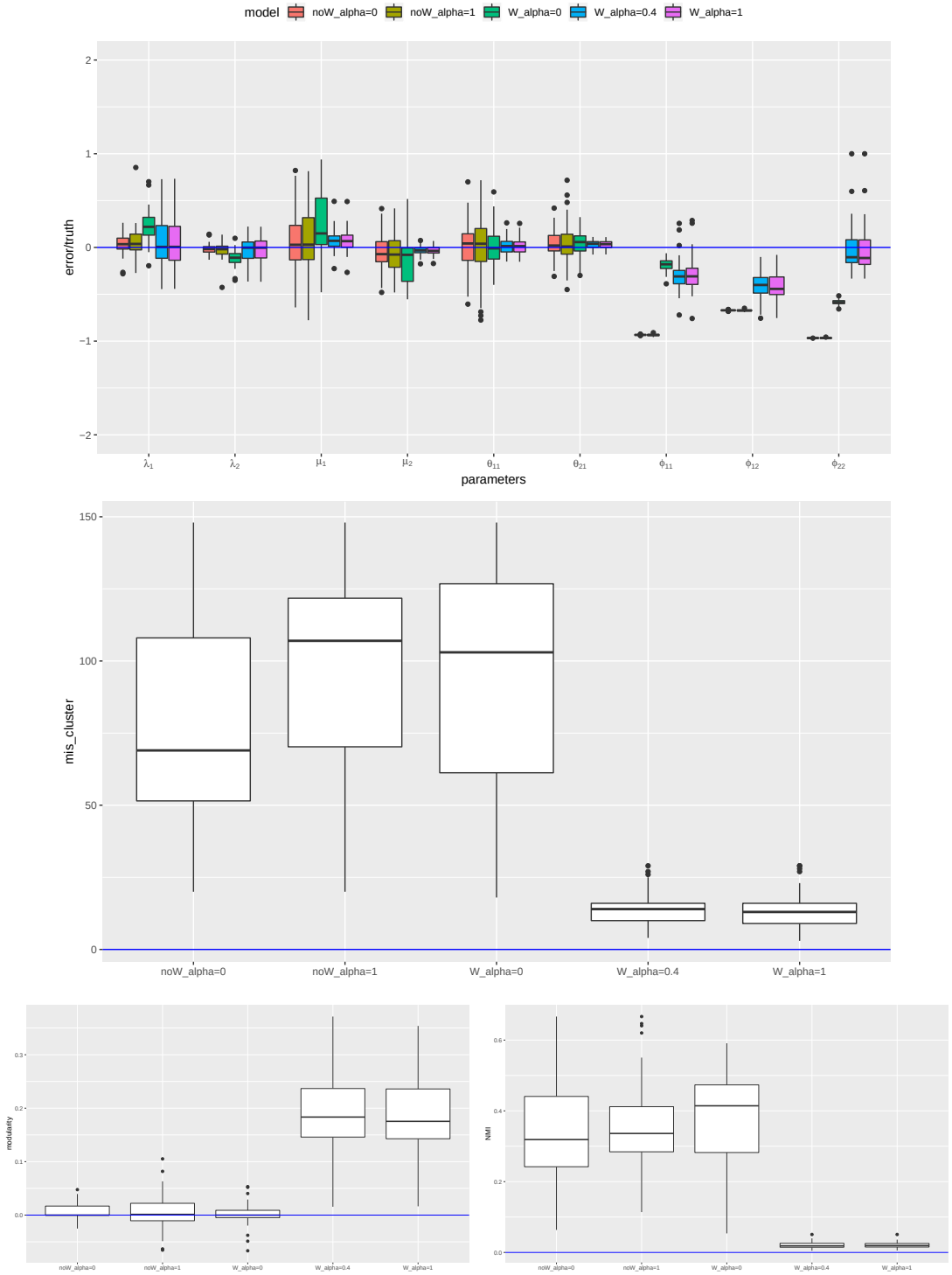


Figure 1.8. Parameter estimations, Number of mis-clusterings, Modularity and NMI by using different models for case III (network separate well) when n=300



larger tuning parameter is important when the network has clear communities. Both case II and case III support the importance of the tuning parameter in clustering sampled network data with node features by using the weighted mixture model.

For the last case, when both network and features don't separate well, Figure 1.9 shows that all methods give large numbers of mis-clusterings. But we can see that models with weighting still give better parameter estimates.

To study the effect of sample size, we do the same work for the RDS sample data with $n = 100$ and the results are summarized in Figures 1.10 to 1.13. We can see that those plots give similar conclusion as we got for RDS sample with $n = 300$. It's also worth to notice that the uncertainty of parameter estimations are larger when $n = 100$ if we compare parameter estimation box-plots in Figures 1.6 to 1.9 with Figures 1.10 to 1.13. This suggests that our proposed mixture model with weighting and tuning parameter for sample network data with node features is pretty robust to sample size of the sampled data in clustering even though smaller sample size results larger standard error for parameter estimates.

From all the simulation result we find that the mixture model with weights gives better parameter estimates. Adding tuning parameter α is essential in finding more interpretable communities. Modularity and normalized mutual information help to determine reasonable tuning parameter values and give us information about the quality of the clustering result.

1.7 Application

In this section, we apply the mixture models with and without weights to cluster RDS data collected on young adult opioid users in New York City (NYC).

Young adult opioid users RDS data in NYC

The data we use are RDS data sampled from opioid users aged 18-29 who had non-medical use of prescription opioids and/or heroin in the past 30 days, currently living

Figure 1.9. Parameter estimations, Number of mis-clusterings, Modularity and NMI by using different models for case IV (both do not separate well) when n=300

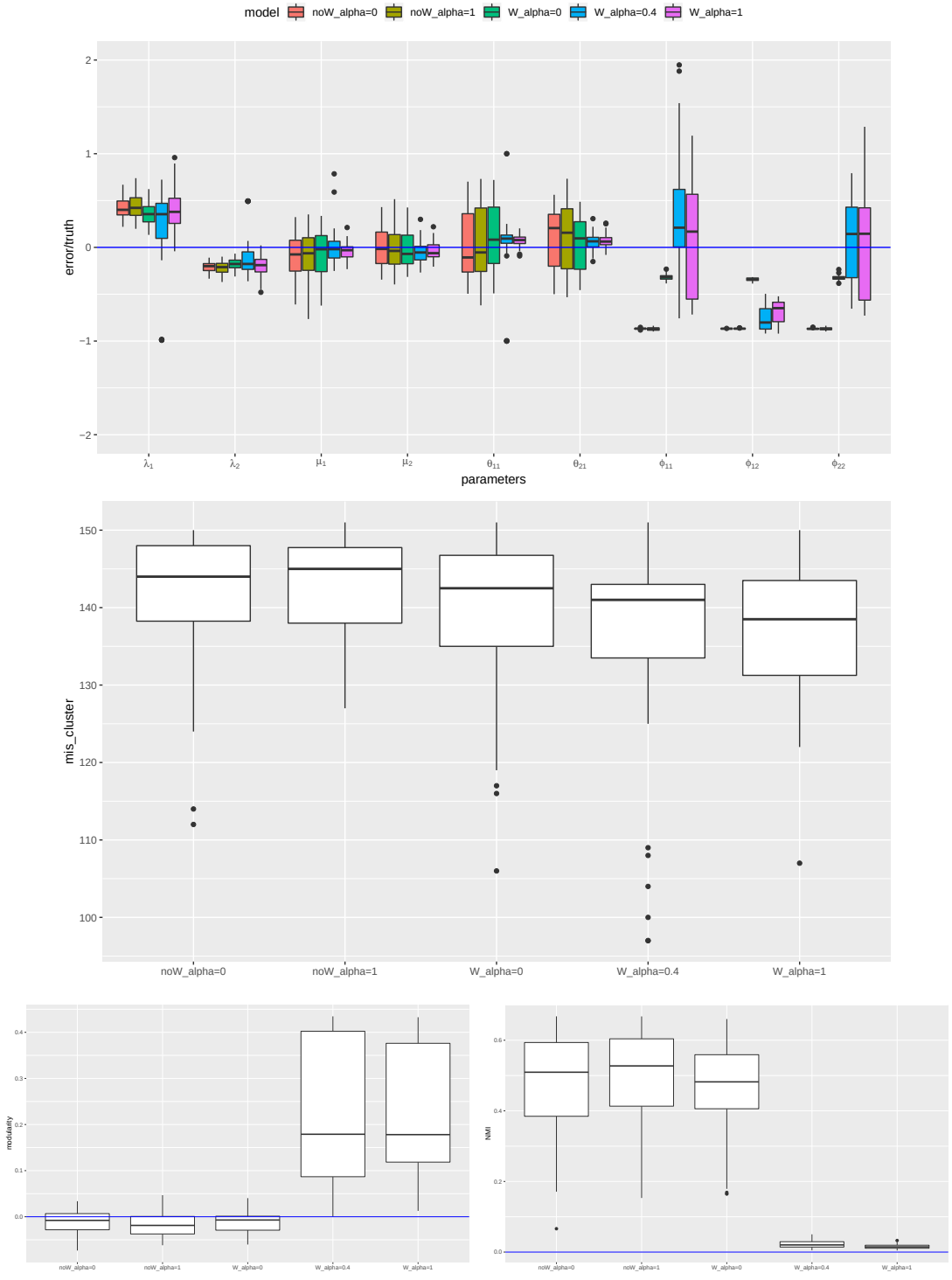


Figure 1.10. Parameter estimations, Number of mis-clusterings, Modularity and NMI by using different models for case I (both separate well) when n=100

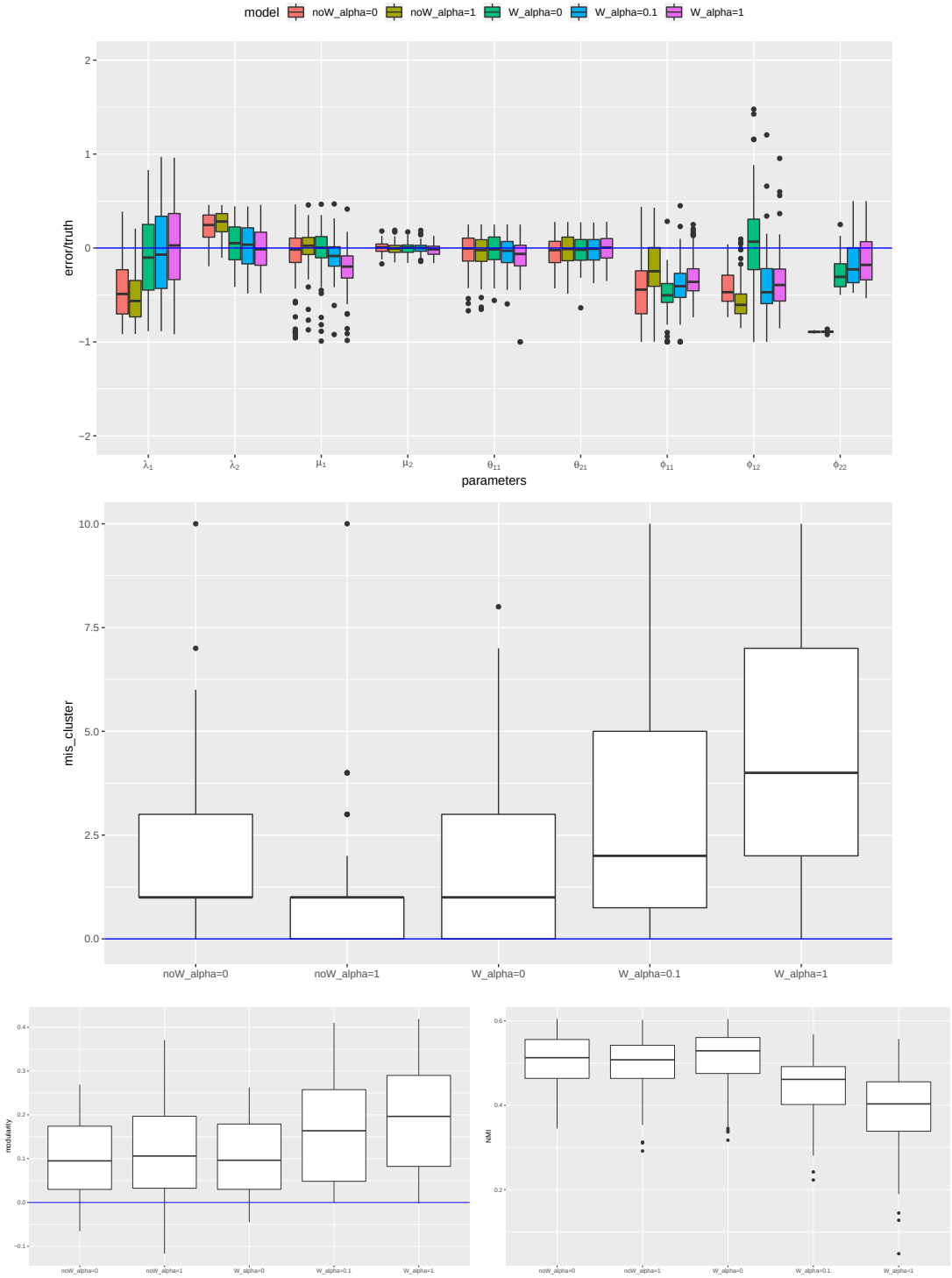


Figure 1.11. Parameter estimations, Number of mis-clusterings, Modularity and NMI by using different models for case II (features separate well) when n=100

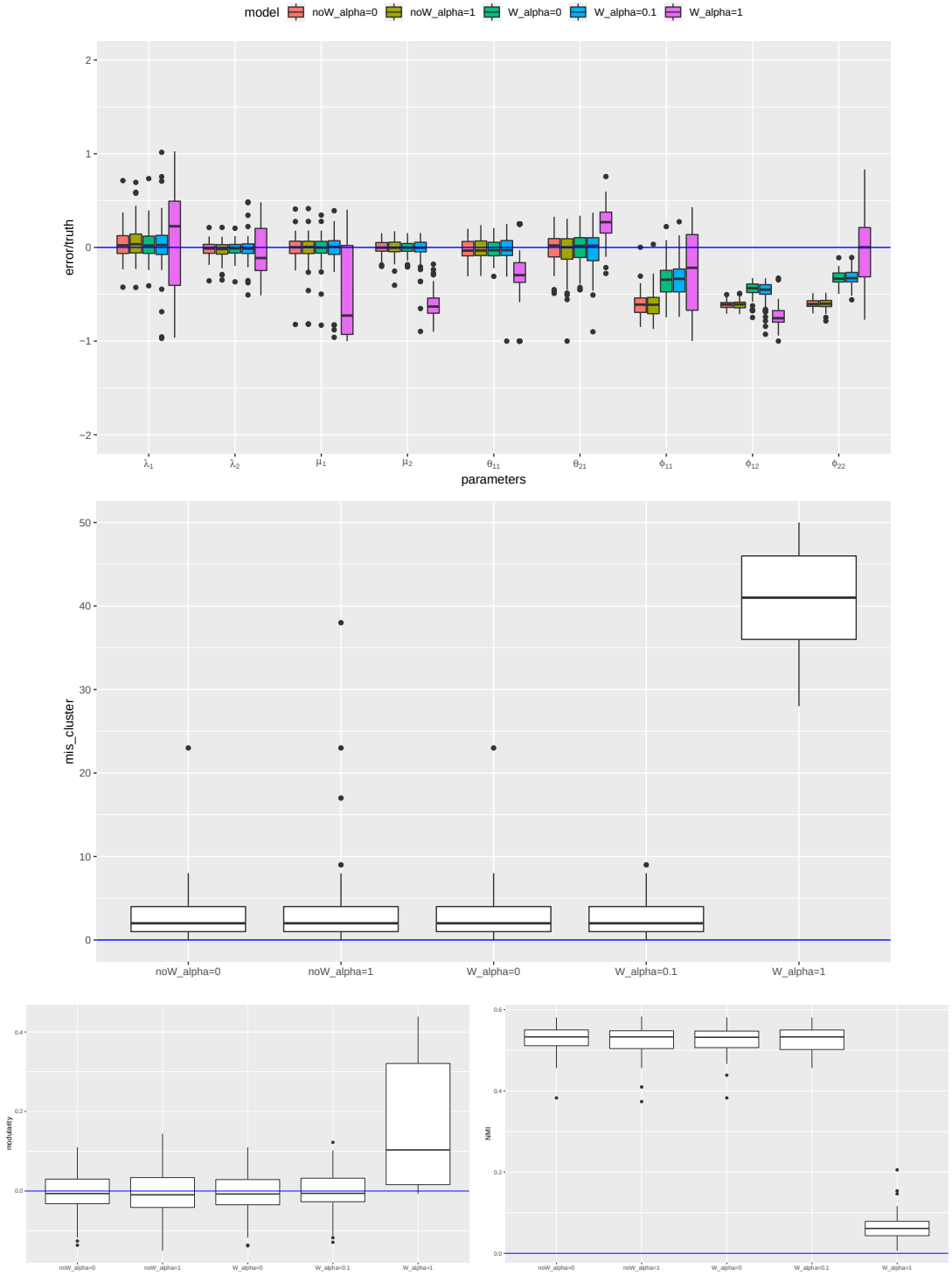


Figure 1.12. Parameter estimations, Number of mis-clusterings, Modularity and NMI by using different models for case III (network separate well) when n=100

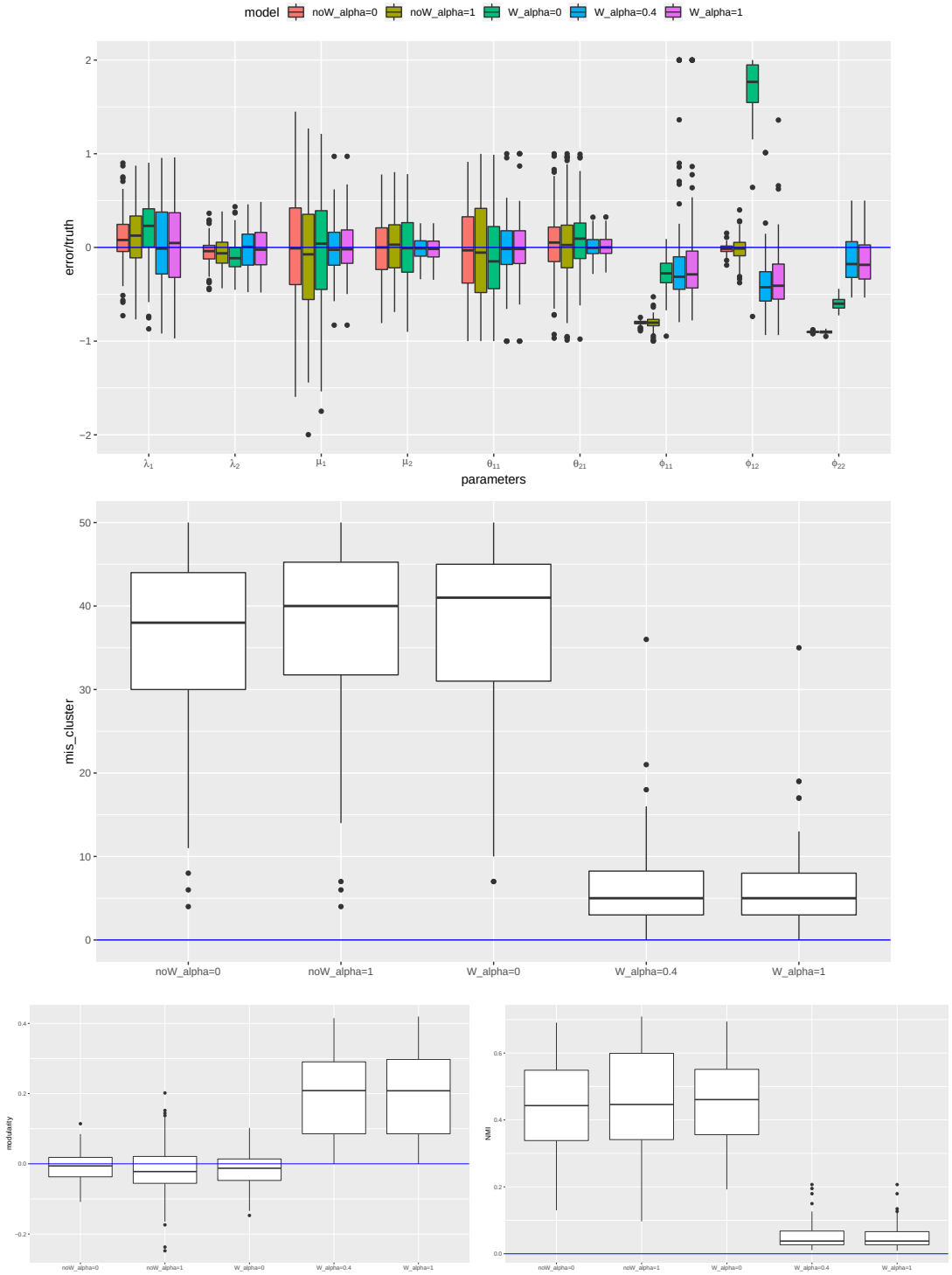
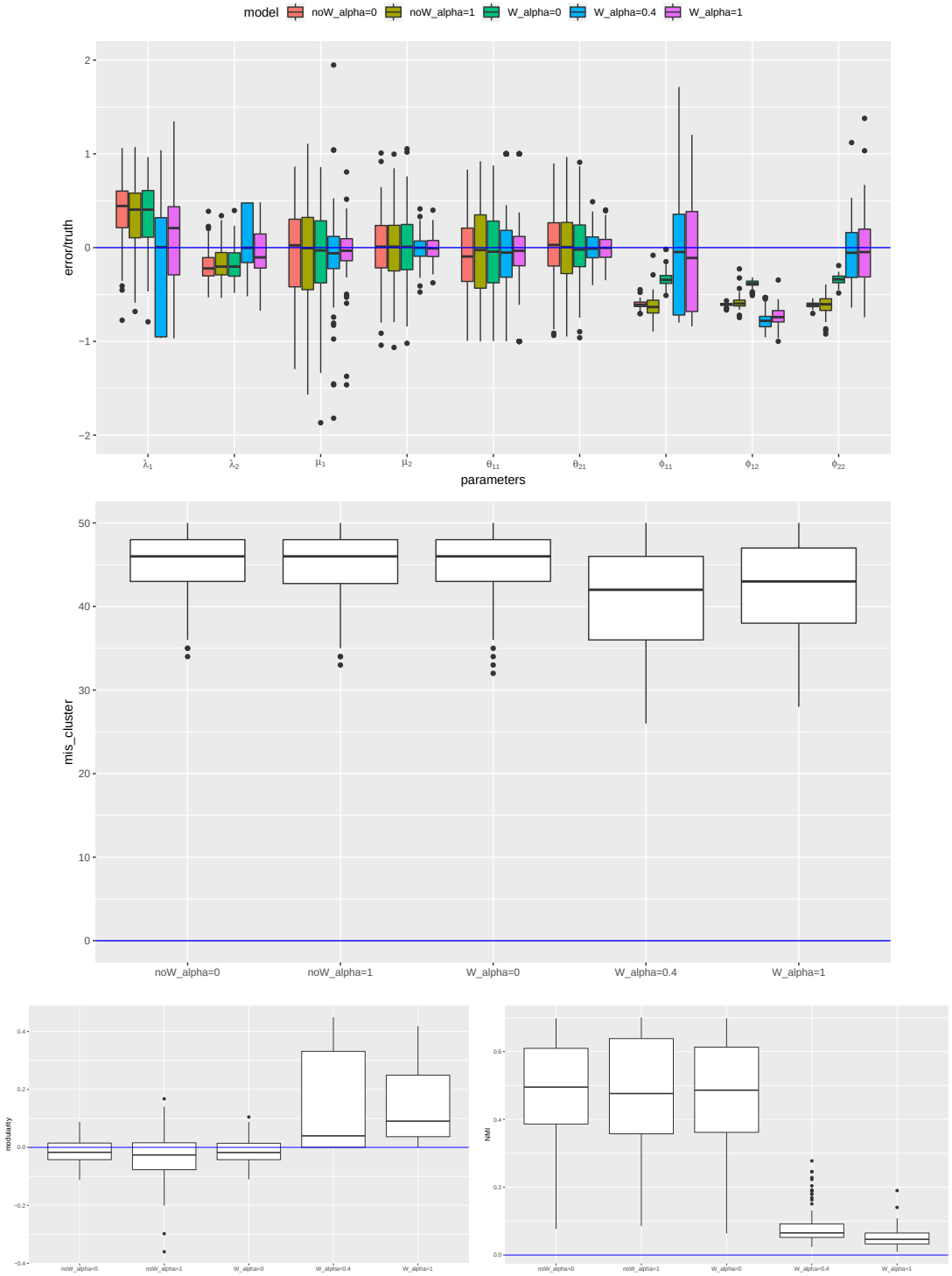


Figure 1.13. Parameter estimations, Number of mis-clusterings, Modularity and NMI by using different models for case IV (both do not separate well) when n=100



in NYC, speak English and are able to provide informed consent. Each participant was interviewed for personal demographic information and drug use behavioral questions. Since participants in this network are recruited through referral, it is believed that community structure exists in this observed recruitment network. To detect those communities, we apply the weighted log-likelihood mixture model with choosen tuning parameter to the NYC young adult opioid users data. Node features used for this clustering are age, borough, opioid injection years, other drugs injection years, homeless, how many are older than 29 among people you know that use POs and live in NYC (NetChar4) and how many inject drugs among people you know that use opioids and live in NYC (NetChar22). The clustering results are summarized in Tables 1.3 and 1.4 and Figure 1.15.

To balance opioid users' attributes and their network connections, we first find a tuning parameter. From Figure 1.14 we can see that the modularity is not small for this sampled network dataset which indicates social communities exist in the opioid users' RDS dataset. When $\alpha = 0$, modularity and NMI are around 0.2. We conclude that communities based on node features explains some community structures of the network which is reasonable for our opioid users RDS dataset because opioid users with similar use behavior are more likely to be connected.

In the mixture model with weighting, the modularity increases and NMI decreases. We choose $\alpha = 1$ as our tuning parameter values because the corresponding NMI values are still not very small and the modularity values are relatively large. In this way, our clustering result is based on both node features and network structure. For the model without weight, $\alpha = 1$ is also reasonable because NMI does not change much with different α values but modularity increase more appreciably from 0.8 to 1.0.

Hepatitis C Virus (HCV) is not included in the clustering model, but from the clustering result graph Figure 1.15, we can see that the weighted mixture model is more

Figure 1.14. Modularity and NMI vs α in the weighted and un-weighted mixture model for the Opioid users RDS data

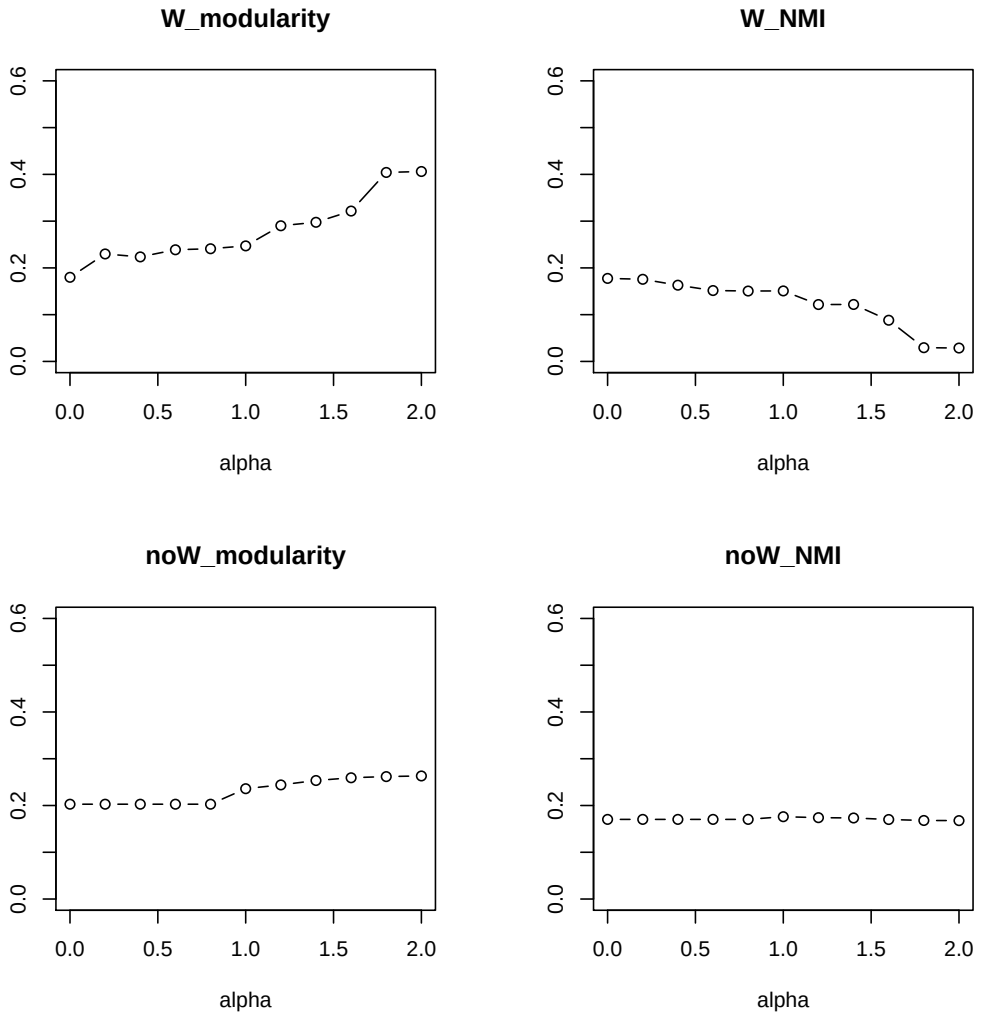


Figure 1.15. Clustering result using mixture model with and without weights on young adult opioid users RDS data in NYC

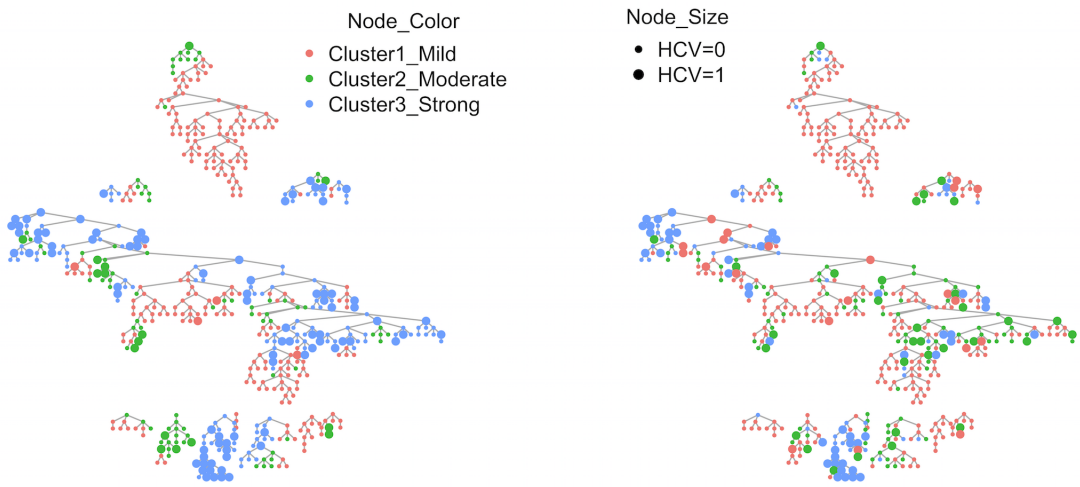


Table 1.3. Feature Comparisons based on clustering from weighted log-likelihood mixture model with = 1 on the young adults opioid users RDS data in NYC.

Cluster	Prop	Prop-HCV	Age	Inj-years	Inj-Others-years	Prop-(NetChar4≥5)	Prop-(NetChar22≥5)	Prop-Homeless
Strong	0.36	0.43	25	5	6.6	0.4	0.74	0.5
Moderate	0.21	0.17	24.5	3.9	2.1	0.27	0.66	0.17
Mild	0.43	0.016	23	0	0.6	0.2	0.21	0.09

- Prop: proportion of sample in each cluster.
- Prop-HCV: proportion of HCV position.
- Age, Inj-years, Inj-Others-year: average age, opioid injection years and others drugs injection years in each cluster.
- NetChar4: how many are older than 29 among people you know that use opioids and live in NYC?
- NetChar22: how many inject drugs among people you know that use opioids and live in NYC?
- Prop-(NetChar4$\geq$ 5): sample proportion in each cluster with NetChar4 $\geq$ 5.
- Prop-(NetChar22$\geq$ 5): sample proportion in each cluster with NetChar22 $\geq$ 5.
- Prop-Homeless: proportion of homeless people in each cluster.

Table 1.4. Sample proportion by clusters from weighted log-likelihood mixture model in each borough for the young adults opioid users' RDS data in NYC.

Cluster	Count	Prop	Prop-Manhattan	Prop-State Island	Prop-Brooklyn	Prop-Bronx	Prop-Queens
Strong	192	0.36	0.47	0.49	0.24	0.17	0.29
Moderate	110	0.21	0.04	0.3	0.26	0.17	0.51
Mild	230	0.43	0.48	0.21	0.49	0.67	0.20

likely to group people with HCV in cluster 1, which contains most heavy opioid drug users. 43.4% people in cluster 1 are HCV positive based on Table 1.3. Also, based on Table 1.3, cluster 1 has people with larger age values, more opioid and drug injectors, people who know more opioid users older than 29 and know more drug injectors, and much more homeless than cluster 2 and cluster 3. Cluster 2 contains moderately risky opioid users. Although average age in it is similar to average age in cluster 1, people in cluster 2 are much newer in terms of injection years, they know fewer 29+ years old opioid users and most of them are not homeless. Cluster 3 is the least risky opioid users group because most of them are young, do not inject, know many fewer older opioid users and injectors. Overall, these three clusters separate opioid drug users very well in terms of those characteristics and drug use behaviors.

Table 1.4 tells us that participants from Bronx and Brooklyn are more likely in the mild cluster (cluster 3), samples from Queens and State Island are more likely to be in the strong cluster (cluster 1). Participants from Manhattan are evenly clustered into strong and mild groups, which we can see from Figure 1.15 that the tree on the top has most of its samples coming from Manhattan and most of them are not homeless. Other people from Manhattan in other trees have much more homelessness. This supports the clustering result that about half participants from Manhattan are mild opioid drug users and half are strong opioid drug users.

With estimated network connection parameter $\hat{\phi}$, we can clearly see that people in the same cluster have more ties than people from different clusters. Among connections between two different clusters, people from moderate and mild clusters have

much stronger cross-cluster connections than people from strong and moderate clusters, people between strong and mild clusters are least likely to be connected. This tells us that mild opioid drug users are much more likely to be influenced by moderate opioid drug users than strong opioid drug users, which targets the population we should focus on for intervention to protect young mild and potential opioid drug users.

Meanwhile, we also applied the mixture model without weights to cluster this NYC young adults opioid users' RDS data. Its clustering result is included in Figure 1.15. From Figure 1.15 we can see that the weighted log-likelihood mixture model clusters more people in the strong opioid drug user group (cluster 1). This is because the weighted log-likelihood mixture model detects network structure better than the one without weights, which results in a clearer social connection effect in the clustering result. Capturing social connection effect is important in the NYC young adults opioid users' RDS data because it gives us guidelines for future interventions.

The network connection parameter estimation (assumed the full network size $N = 1e4$) based on the weighted log-likelihood mixture model with $\alpha = 1$ is

$$\hat{\phi} = \begin{array}{c} \begin{matrix} \textit{Strong} & \textit{Moderate} & \textit{Mild} \end{matrix} \\ \begin{bmatrix} 0.015 & 0.0005 & 0.0002 \\ 0.0005 & 0.016 & 0.001 \\ 0.0002 & 0.0014 & 0.009 \end{bmatrix} \end{array} \begin{array}{c} \textit{Strong} \\ \textit{Moderate} \\ \textit{Mild} \end{array}$$

1.8 Discussion and Conclusions

In this paper, we build a mixture model with weighted log-likelihood inference for clustering node-attributed RDS sample data. We also propose to add a tuning parameter to the weighted log-likelihood to balance contribution of node features and network structure in clustering. Node features in RDS network clustering enable us to understand how nodes differ across groups, and critically help to detect clusters

despite the multiple isolated tree structures generated by the RDS. From the simulation study with two different RDS sample sizes, we see that the clustering algorithm is robust to the sample proportion. Adding weights as inverse sampling probabilities to the log-likelihood reduces bias in parameter estimation because RDS is not simple random sampling. Edge sampling probabilities are essential to capture the truth that two un-connected nodes in the RDS data does not necessarily mean they are not connected in the full network. This relates a very sparse RDS network to a less sparse underlying network. Weighted log-likelihood inference results in better network connection parameter estimation which tells us a closer truth about how strong the connections are within and between clusters in the underlying social network. To evaluate the clustering quality and find a proper tuning parameter value, we also discussed modularity and normalized mutual information and modified it for the pseudo-population network data. We recommend using these two metrics together to select a value for the tuning parameter.

CHAPTER 2

NESTED DIRICHLET PROCESS FOR POPULATION SIZE ESTIMATION FROM MULTI-LIST RECAPTURE DATA

2.1 Abstract

Heterogeneity of response patterns is important in estimating the size of a closed population from multi-list recapture data when capture patterns are different over time and location. In this paper, we extend the one layer Dirichlet Process mixture model proposed by Manrique-Vallier (2016) to a Nested Dirichlet Process model with the first layer modeling individual heterogeneity and the second layer modeling location-time differences. In the Nested Dirichlet Process mixture model, location-time groups with similar recording patterns are in the same top layer latent class and individuals within it are dependent. The Nested Dirichlet Process mixture model incorporates hierarchical heterogeneity into the modeling to estimate population size from multi-list recapture data.

2.2 Introduction

The estimation of the size of a closed population from multi-list recapture data has been studied in many settings, for example estimation of census undercount [Chao and Tsay, 1998][Darroch et al., 1993], estimation of deaths in armed conflict [Ball et al., 2003][Manrique-Vallier et al., 2013a], fatal victims [Manrique-Vallier et al.], estimation of drug injectors [Overstall et al., 2014] and estimation of human trafficking victims [Heijden, 2016]. In general, each record in multi-list recapture data has

descriptive features, like time, location, gender, age, etc. To reduce uncertainty of list capture probabilities imposed by hierarchical structure, such as location differences, it's necessary to account for heterogeneity of response patterns in estimating the population size. One way to account for this part of heterogeneity is stratification. However, expert based stratification are too subjective. Another method is to stratify by location or time which may results in too many strata [Ball et al., 2003] [Manrique-Vallier et al.]. In this paper, we put heterogeneity caused by location and time into the model by building a non-parametric multi-layer latent class model based on the non-parametric one layer latent class model, Dirichlet process mixtures of product-Bernoulli distributions, proposed by Manrique-Vallier (2016) [Manrique-Vallier, 2016b]. In Manrique-Vallier (2016)'s paper, the latent layer models individual heterogeneity and individuals in the same latent class are independent. To reflect the hierarchical structure of the data, we add one more layer on top of the individual layer to capture the top group (location-time) differences and to allow dependence among individuals in the same top latent layer.

Many techniques estimate the population size by modeling list dependency. A class of generalized linear models, known as loglinear models [Bishop et al., 1975] assume the expected log of cell count is linearly related to a set of list interactions. Averaging over Bayesian graphical decomposable models, which represent graphical models of list dependency, is also a classical method to estimate population size from multi-list recapture data [Madigan et al., 1995]. Those methods treat all individuals the same which may not be proper in some cases, for example in our Syrian war casualties application civilian and military deaths are captured differently by some lists. Rasch models and extensions on them [Rasch, 1993] [Darroch et al., 1993] [Agresti, 1994] [Fienberg et al., 1999] incorporate individual heterogeneity into log-linear models to model list dependence. A more flexible method, mixture models have also been used to capture individual heterogeneity [Manrique-Vallier and Fienberg, 2008]

[Manrique-Vallier, 2016b]. One strong assumption in the one layer latent class model is that individuals are independent given class label, which might not be proper for data with hierarchical structure.

A popular alternative to the one layer latent class model for solving the individual dependence problem in nested data is multi-level latent class models [Vermunt, 2003] [Teh et al., 2006] [Rodrlguez et al., 2008]. However, multi-level latent class models haven't been applied in population size estimation for multi-list recapture data. In over-time and across-location multi-list recapture data, we want the top layer to capture location-times that having similar recording patterns and the bottom layer to capture hidden classes of individuals within each top layer latent class. To realize this goal, both the hierarchical Dirichlet process (HDP) [Teh et al., 2006] and the nested Dirichlet process (NDP) [Rodrlguez et al., 2008] models are great candidates. NDP allows both mixture components and weights to change within different top layer classes, but HDP components only differ in weights. Due to the complicated and potentially highly distinct list dependencies among top layer classes in our Syrian conflict application, recording patterns might differ very much between one class containing governorates with intense conflicts and one class containing governorates with much less conflict. Therefore, we choose Nested Dirichlet Process (NDP) models in this paper. NDP is usually applied in clustering nested data, like documents [Blei et al., 2010][Blei et al., 2007][Fox et al., 2011]. In this paper, we apply a NDP of product-Bernoulli mixtures to identify more accurate hidden homogeneous classes among top level groups and among individuals within top level groups to better estimate population size.

The article is organized as follows. In Section 2.3, we talk about the data and problem that motivates us for this paper. Then we introduce our proposed approach and MCMC inference for parameter estimation in Section 2.4. In Section 2.5, we do simulations to compare results from the one layer Dirichlet Process and Nested Dirichlet

Process. In Section 2.6, we apply the NDP mixtures in the Syrian conflict data to estimate population sizes. In Section 2.7, we make conclusions and discuss potential future work.

2.3 The Syrian conflict data

Human Rights Data Analysis Group (HRDAG) is a non-profit organization that applies rigorous science to the analysis of human rights violations around the world. One of its projects is to estimate the total number of killings during the Syrian conflict based on multi-list recapture data. The Syrian conflict data we are using contains identifiable people who were killed during Syrian conflict from March 2011 to March 2016. Each death record has variables describing this person, which include the person's name, death date, governornate (region in Syria), gender, age. Deaths were recorded by four data sources ($S = 4$) investigating deaths in the Syrian conflict, namely Syrian Center for Statistics and Research (SCSR), Damascus Center for Human Rights Studies (DCHRS), Syrian Network for Human Rights (SNHR) and Violations Documentation Center (VDC). Each record might be captured by more than one data source, thus the number of capture patterns is $2^S - 1 = 15$ excluding the undocumented killings, with S as number of data sources. Due to data confidentiality, in this paper we randomly generate a sample of $n = 36226$ from all the documented killings. The number of killings recorded under each pattern in this sampled Syrian conflict data is summarized in Table 3.3. We can see that $n_{1000} = 6039$ deaths are captured by VDC only, $n_{1010} = 652$ are captured by VDC and DCHRS, not by SNHR and SCSR, and $n_{1111} = 4252$ are captured by all four data sources. Estimating the number of undocumented killings is equivalent to estimating n_{0000}, and is the goal of our inference.

In the Syrian conflict data, documented killings are from 14 governorates across the country. From Figure 2.1, we can see that recording patterns within governorate

Table 2.1. Number of killings under each recording pattern in the Syrian conflict data

VDC	SNHR	DCHRS	SCSR	Num-Records
1	0	0	0	$n_{1000} = 6039$
0	1	0	0	$n_{0100} = 3273$
0	0	1	0	$n_{0010} = 1363$
0	0	0	1	$n_{0001} = 2370$
1	1	0	0	$n_{1100} = 3099$
1	0	1	0	$n_{1010} = 652$
1	0	0	1	$n_{1001} = 2060$
0	1	1	0	$n_{0110} = 921$
0	1	0	1	$n_{0101} = 1410$
0	0	1	1	$n_{0011} = 514$
1	1	1	0	$n_{1110} = 1346$
1	1	0	1	$n_{1101} = 6483$
1	0	1	1	$n_{1011} = 1572$
0	1	1	1	$n_{0111} = 872$
1	1	1	1	$n_{1111} = 4252$
0	0	0	0	$n_{0000} =?$

change overtime. For example from 04/2011 to 12/2012 deaths captured by all four sources overtake records in other patterns in Rural Damascus. From 01/2013 to 08/2014, more deaths are captured by VDC, SNHR and SCSR together, but not by DCHRS. From 03/2015 to 12/2015, n_{1101} is larger than others or more deaths were captured by VDC, DCHRS and SCSR, but not by SNHR. Some sources capture killings better than others in some governorates, for example, most killings were recorded by VDC in Tartus. Meanwhile, the documented number of killings recorded in different governorates differs much too. All those differences are not hard to explain if we think about the location of each governorate, when and where a small or a big conflict happened. With those findings, we believe that it's not a good idea to combine all the death records simply over all time and governorates like what we did in Table 3.3 to estimate the total number of killings. Due to the long period and many governorates in this data set, it's also a challenge to do proper stratification

subjectively. Therefore, our nested model is important for detecting higher level (e.g. governorate-time) strata in this problem.

2.4 Nested Dirichlet Process of product-Bernoulli mixtures

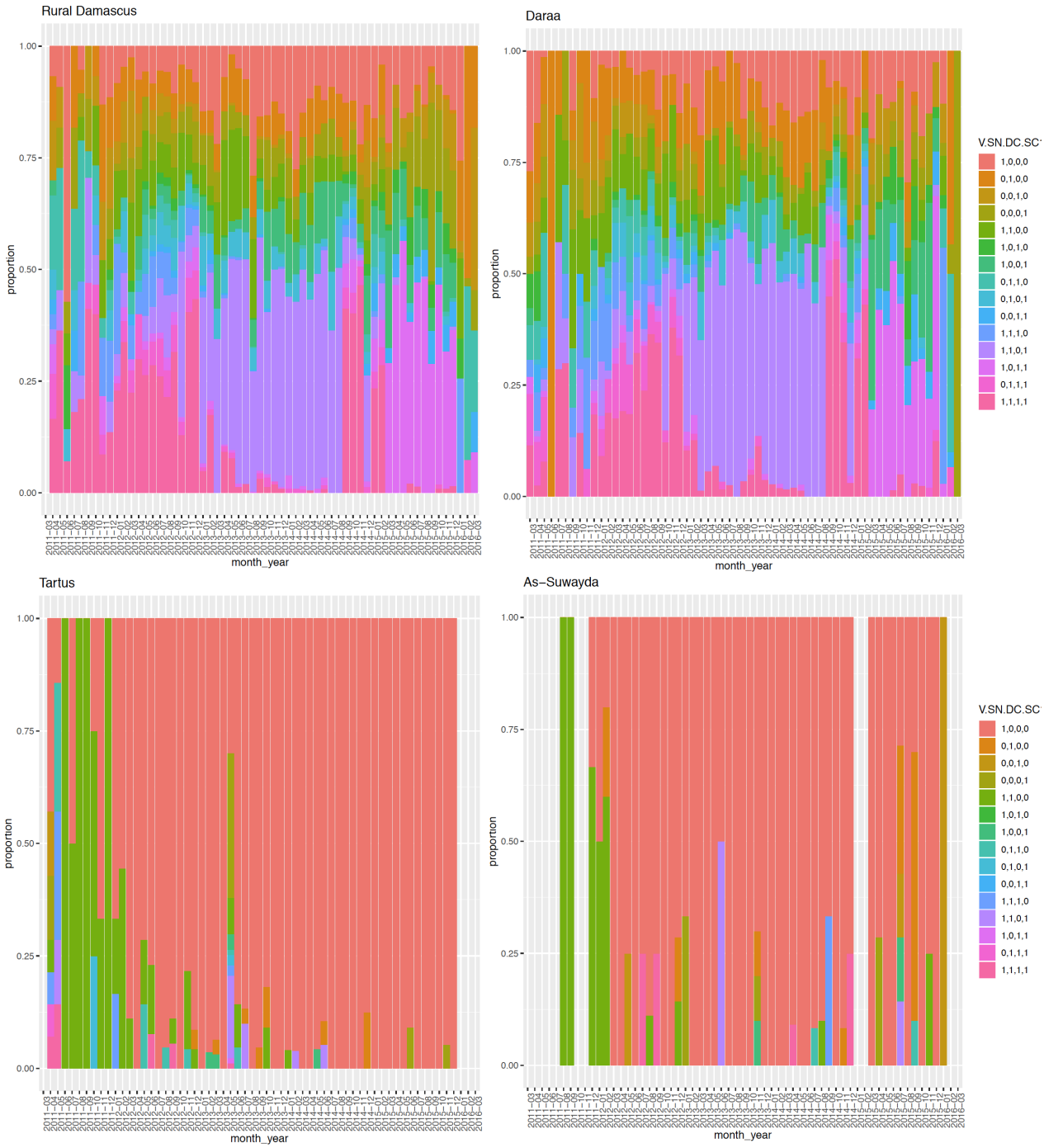
2.4.1 Bayesian non-parametric product-Bernoulli mixtures with NDP prior

In this paper, we build a product-Bernoulli mixtures model with Nested Dirichlet Process as prior for population size estimation from multi-list recapture data. Assume individuals belongs to latent classes in layer 1, and covariate groups (e.g. location-time) belong to latent classes in layer 2. Conditional on both latent layers, individual capture probabiities for each list are independent, of both other lists and other individuals. The probability that an individual is captured by the s^{th} list is denoted $\lambda_{k,l,s}$, where k is their top-layer class and l is their layer 1 class. This means this probability is influenced by both the individual's layer 1 latent class l and its top level latent class k. Meanwhile, for individual i in location-time j, its first layer latent class $z_{i,j}^{(1)}$ depends on its top layer latent class $z_j^{(2)}$. From Figure 2.4.1, we can see that individuals in the same latent class are independent given class in the one layer latent class model. From Figure 2.5, we can see a graphical model with nested structure. Its top layer latent class reflects group (e.g. location-time) heterogeneity and the first layer models individual heterogeneity within its top layer. In the two layer latent class model, we relax the local independent assumption in the one layer latent class model. Individuals in the same top layer latent class are allowed to be dependent. If our data are given by

$$y_{i,j,s} = \begin{cases} 1, & \text{if person } i, \text{in the } j^{th} \text{ top group is captured by the } s^{th} \text{ data list} \\ 0, & \text{otherwise,} \end{cases}$$

our model is:

Figure 2.1. Stacked barplots for proportion of records by 15 capture patterns over time. This plot only shows barplots from four governorates and it's based on monthly data. The recording pattern corresponds to data sources VDC, SNHR, DCHRS, SCSR in order.



$$
\begin{aligned}
&(y_{i,j,s}|z_{i,j}^{(1)} = l, z_j^{(2)} = k) \sim \text{Bernoulli}(\lambda_{k,l,s}) \\
&(z_{i,j}^{(1)}|z_j^{(2)} = k) \sim \text{Cat}(\pi_{k,1}^{(1)}, \pi_{k,2}^{(1)}, \cdots, \pi_{k,l}^{(1)}, \cdots) \\
&z_j^{(2)} \sim \text{Cat}(\pi_1^{(2)}, \pi_2^{(2)}, , \cdots, \pi_k^{(2)}, \cdots) \\
&\lambda_{k,l,s} \sim \text{Beta}(1,1) \\
&(\pi_{k,1}^{(1)}, \cdots, \pi_{k,l}^{(1)}, \cdots) \sim \text{SB}(\alpha_k),\ \alpha_k \sim \text{Gamma}(a_k, b_k) \\
&(\pi_1^{(2)}, \cdots, \pi_k^{(2)}, \cdots) \sim \text{SB}(\alpha_0),\ \alpha_0 \sim \text{Gamma}(a_0, b_0),
\end{aligned}
$$

where

- $i = 1, \cdots, N_j; j = 1, \cdots, J; s = 1, \cdots, S$; N_j and n_j are the number of total and observed individuals in the j^{th} second layer group (e.g. j^{th} location-time), J is the number of second layer groups, S is the number of data sources. Total number of observed individuals is $n = \sum_{j=1}^{J} n_j$ and the population size is $N = \sum_{j=1}^{J} N_j$.

- $(z_{i,j}^{(1)} = l|z_j^{(2)} = k)$ means the i^{th} person in the j^{th} top group falls into the l^{th} first layer latent class given its second layer latent class as k. $k, l = 1, 2, \cdots$.

- We use a stick-breaking prior, which is popularly used in non-parametric Bayesian mixture models to learn the number of mixture components from data.

$$
(\pi_1^{(2)}, \cdots, \pi_k^{(2)}, \cdots) \sim \text{SB}(\alpha_0),\ (\pi_{k,1}^{(1)}, \cdots, \pi_{k,l}^{(1)}, \cdots,) \sim \text{SB}(\alpha_k),
$$

  where $\pi_k^{(2)} = U_k^{(2)} \Pi_{h=1}^{k-1}(1 - U_h^{(2)})$, $U_k^{(2)} \sim \text{Beta}(1, \alpha_0)$ and $\pi_{k,l}^{(1)} = U_{k,l}^{(1)} \Pi_{h=1}^{l-1}(1 - U_{k,h}^{(1)})$, $U_{k,h}^{(1)} \sim \text{Beta}(1, \alpha_k)$.

  Suppose $(\pi_1^{(2)}, \cdots, \pi_k^{(2)}, \cdots) \sim \text{SB}(\alpha_0)$. For a unit-length stick, each time break a proportion $U^{(2)}$ of the remaining stick. After the $(k-1)^{th}$ break, there is $\Pi_{h=1}^{k-1}(1 - U_h^{(2)})$ left, then the k^{th} break length will be $U_k^{(2)} \Pi_{h=1}^{k-1}(1 - U_h^{(2)})$, which

Figure 2.2. One layer Latent class model: Z, individual latent class

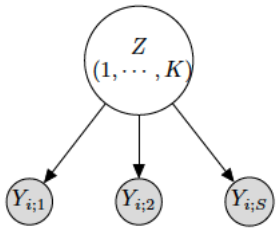
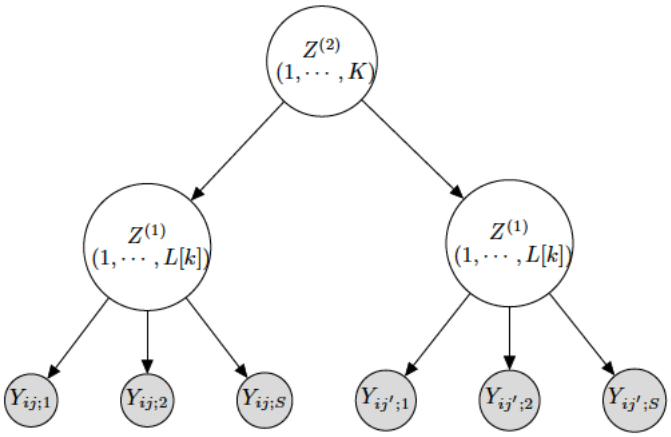


Figure 2.3. Two layer latent class model: $Z^{(1)}$, individual layer, $Z^{(2)}$, top (e.g. location-time) layer.



equals to $\pi_k^{(2)}$. Since $U_k^{(2)} \sim \text{Beta}(1, \alpha_0)$, large α_0 gives small break proportions $U_k^{(2)}$ for $k = 1, \cdots$, then small break length $\pi_k^{(2)}$ and large number of breaks. Thus, α_0 controls the number of latent classes in the second layer and α_k controls the number of latent classes in the first layer given its top layer in latent class k. Large $\pi_k^{(2)}s, k = 1, \cdots$, and $\pi_{k,l}^{(1)}s, l = 1, \cdots$, will corresponding to cluster proportions learnt from data by the model. We take large enough upper bounds K^* and L^* for number of latent classes in the second and first layers.

2.4.2 Markov Chain Monte Carlo for parameter estimation

An MCMC based Gibbs sampling procedure has been well developed for parameter estimation in the one layer mixture model in the multi-list recapture setting [Manrique-Vallier, 2016b] [Manrique-Vallier and Fienberg, 2008] [Fienberg et al.,

1999]. Meanwhile, MCMC for the Nested Dirichlet Process is also studied in the clustering nested data problem [Rodrlguez et al., 2008]. In this paper, we use data augmentation and jointly update population size N and latent variables $z^{(2),0}$ and $z^{(1),0}$ using a conditional decomposition [Manrique-Vallier, 2016b] [Basu and Ebrahimi, 2001] to update parameter N. For the Nested Dirichlet Process mixture model above, the full likelihood given latent classes $\boldsymbol{z^{(1)}, z^{(2)}}$ and parameter set $\Theta = \{\boldsymbol{\lambda, \pi^{(2)}, \pi^{(1)}}, \alpha_{k=1,\cdots}, \alpha_0, a_{k=1,\cdots}, b_{k=1,\cdots}, a_0, b_0\}$ is

$$
\begin{aligned}
P(\boldsymbol{Y, w}|\boldsymbol{z^{(1)}, z^{(2)}}, \Theta) \propto & \binom{N}{n, w_{1,1}, \cdots, w_{k,l}, \cdots} \Pi_k \Pi_l \left[\pi_k^{(2)} \pi_{k,l}^{(1)} \Pi_{s=1}^{S} (1 - \lambda_{k,l,s})\right]^{w_{k,l}} \\
& \Pi_k \Pi_l \Pi_{s=1}^{S} \left[\pi_k^{(2)} \pi_{k,l}^{(1)} (1 - \lambda_{k,l,s})\right]^{n_{k,l,s;0}} \Pi_k \Pi_l \Pi_{s=1}^{S} \left[\pi_k^{(2)} \pi_{k,l}^{(1)} \lambda_{k,l,s}\right]^{n_{k,l,s;1}} \\
& I_{(n + \sum_k \sum_l w_{k,l} = N)}
\end{aligned}
$$

where $\boldsymbol{w} = \{w_{k,l}; k = 1, \cdots, K; l = 1, \cdots, L\}$, $w_{k,l}$ = size of set $\{(y_{i,j,s=1,\cdots,S} = 0) \& (z_j^{(2)} = k) \& (z_{i,j}^{(1)} = l)\}$ is the number of un-documented records that fall into the second layer latent class k and the first layer latent class l. $n_{k,l,s;1} = ||\{(y_{i,j,s} = 1) \& (z_j^{(2)} = k) \& (z_{i,j}^{(1)} = l)\}||$ is the number of documented records falling into the second layer latent class k and the first layer latent class l and captured by the s^{th} data list. $n_{k,l,s;0} = ||\{(y_{i,j,s} = 0) \& (z_j^{(2)} = k) \& (z_{i,j}^{(1)} = l) \& (y_{i,j,s=1,\cdots,S} \text{ not all equals to } 0)\}||$ is the number of documented records falling into the second layer latent class k and the first layer latent class l and not captured by the s^{th} data list.

Instead of setting the number of latent classes to be infinity, truncated approximation is used by setting large numbers K, L to the second and first level latent classes [Ishwaran and James, 2001][Ishwaran and James, 2002]. The MCMC iterates as follows:

1. Update top layer latent class $z_j^{(2)}$:

$$
\begin{aligned}
P(z_j^{(2)} = k|Y, \pi^{(1)}) &= \sum_{z_{1,j}^{(1)}} \cdots \sum_{z_{N_j,j}^{(1)}} P(z_j^{(2)} = k, z_{1,j}^{(1)}, \cdots, z_{N_j,j}^{(1)}|Y) \\
&\propto \sum_{z_{1,j}^{(1)}} \cdots \sum_{z_{N_j,j}^{(1)}} P(Y|z_j^{(2)} = k, z_{1,j}^{(1)}, \cdots, z_{N_j,j}^{(1)}) P(z_j^{(2)} = k, z_{1,j}^{(1)}, \cdots, z_{N_j,j}^{(1)}) \\
&= \sum_{z_{1,j}^{(1)}} \cdots \sum_{z_{N_j,j}^{(1)}} P(y_{i,j}; i = 1, \cdots, N_j|z_j^{(2)} = k, z_{1,j}^{(1)}, \cdots, z_{N_j,j}^{(1)}) \\
&\qquad * P(z_{1,j}^{(1)}, \cdots, z_{N_j,j}^{(1)}|z_j^{(2)} = k) P(z_j^{(2)} = k) \\
&= \sum_{i=1}^{N_j} \sum_{z_{i,j}^{(1)}=1}^{L} \Pi_{s=1}^{S} \lambda_{k,z_{i,j}^{(1)},s}^{y_{i,j,s}} (1 - \lambda_{k,z_{i,j}^{(1)},s})^{1-y_{i,j,s}} \pi_{k,z_{i,j}^{(1)}}^{(1)} \pi_k^{(2)},
\end{aligned}
$$

where $N_j = n_j + n_{0,j}$, n_j is the number of documented recordings and $n_{0,j}$ is the estimated missing recordings in the j^{th} location-time, $n_{0,j} = \frac{n_0}{n} n_j$. n, n_0 are the total number of documented recordings and missing recordings.

2. Update first layer latent class $z_i^{(1)}$:

$$
\begin{aligned}
&P(z_{i,j}^{(1)} = l|z_j^{(2)} = k, Y, \pi^{(1)}) \\
&\propto P(y_{ij}|z_j^{(2)} = k, z_{i,j}^{(1)} = l) P(z_i^{(1)} = l|z_j^{(2)} = k) \\
&\propto \Pi_{s=1}^{S} \lambda_{k,l,s}^{y_{i,j,s}} (1 - \lambda_{k,l,s})^{1-y_{i,j,s}} \pi_{k,l}^{(1)}.
\end{aligned}
$$

3. Update list capture parameters $\lambda_{k,l,s}$:

$$
P(\lambda_{k,l,s}|\cdots) \propto (1 - \lambda_{k,l,s})^{w_{k,l}} \Pi_{j=1}^{J} \Pi_{i=1}^{n_j} \lambda_{k,l,s}^{y_{i,j,s}} (1 - \lambda_{k,l,s})^{1-y_{i,j,s}}
$$

$(\lambda_{k,l,s}|\cdots) \sim \text{Beta}(1 + n_{k,l,s;1}, 1 + n_{k,l,s;0} + w_{k,l})$.

4. Update $\pi_k^{(2)}$: $\pi_k^{(2)} = U_k^{(2)} \Pi_{h<k} (1 - U_h^{(2)})$:

since

$$P(\pi_k^{(2)}, \pi_{k,l}^{(1)}|\cdots) \propto P(y|\pi_k^{(2)}, \pi_{k,l}^{(1)}, \cdots)P(\pi_k^{(2)}, \pi_{k,l}^{(1)})$$
$$\propto \Pi_{s=1}^{S}\left[\pi_k^{(2)}\pi_{k,l}^{(1)}\right]^{n_{k,l,s}+m_{k,l,s}+w_{k,l,s}}\lambda_{k,l,s}^{n_{k,l,s}}(1-\lambda_{k,l,s})^{m_{k,l,s}+w_{k,l,s}}P(\pi_k^{(2)}, \pi_{k,l}^{(1)}),$$

changing $\pi_k^{(2)}$ to an expression with $U_k^{(2)}$ using $\pi_k^{(2)} = U_k^{(2)}\Pi_{h<k}(1-U_h^{(2)})$ and combining with Beta prior of $U_k^{(2)}$, gives a Beta posterior for $U_k^{(2)}$, which we can use the update $\pi_k^{(2)}$. Let $U_{K^*}^{(2)} = 1$, $U_k^{(2)} \sim \text{Beta}(1+u_k^{(2)}, \alpha_0 + \sum_{h>k} u_h^{(2)})$ for $k = 1, \cdots K^* - 1$, and $u_k^{(2)} = n_k^{(2)} + w_k^{(2)}$. $n_k^{(2)}, w_k^{(2)}$ are the numbers of captured and non-captured individuals whose second layer latent class is k.

5. Update α_0: $\alpha_0 \sim \text{Gamma}(a_0 - 1 + K^*, b_0 - \log\pi_{K^*}^{(2)})$.

6. Update $\pi_{kl}^{(1)}$: $\pi_{kl}^{(1)} = U_{kl}^{(1)}\Pi_{h<l}(1-U_{kh}^{(1)})$
   let $U_{kL^*[k]}^{(1)} = 1$, $U_{kl}^{(1)} \sim \text{Beta}(1+u_{kl}^{(1)}, \alpha_k + \sum_{h>l} u_{kh}^{(1)})$ for $l = 1, \cdots L^*[k]-1$, and $u_{kl}^{(1)} = n_{kl} + w_{kl}$. n_{kl}, w_{kl} are the numbers of individuals captured and non-captured in the class with it's first layer latent class l and second layer latent class k.

7. Update α_k: $\alpha_k \sim \text{Gamma}(a_k - 1 + L^*[k], b_k - \log\pi_{kL^*[k]}^{(1)})$.

8. Update N, w_{kl} for all k, l: Given $P(N) \propto 1/N$,

$$P(N, w_{kl}|\cdots) \propto \frac{(N-1)!}{\Pi_{k=1}^{K}\Pi_{l=1}^{L^*[k]}w_{kl}!(n-1)!}\Pi_{k=1}^{K}\Pi_{l=1}^{L^*[k]}\rho_{kl}^{w_{k1}}(1-\sum_{k=1}^{K}\sum_{l=1}^{L^*[k]}\rho_{kl}^{w_{k1}})^n.$$

   This is a negative multinomial distribution with $N = \sum_{k=1}^{K}\sum_{l=1}^{L^*[k]} w_{k1} + n = n_0 + n$, $\rho_{kl} = \pi_k^{(2)}\pi_{kl}^{(1)}\Pi_{s=1}^{S}(1-\lambda_{k,l,s})$.

   Then,

$$n_0 \sim \text{NegBinomial}(n, 1 - \sum_{k=1}^{K}\sum_{l=1}^{L^*[k]}\pi_k^{(2)}\pi_{kl}^{(1)}\Pi_{s=1}^{S}(1-\lambda_{kl;s})),$$
$$(w_{kl}; \text{for all } k, l) \sim \text{Multinomial}(n_0, (p_{kl}; \text{for all } k, l)),$$

   where $p_{kl} \propto \rho_{kl}$.

Table 2.2. Two layer latent class proportions and list capture probabilities

		List capture probabilities			
Layer 2 proportion	Layer 1 proportion	list 1	list 2	list 3	list 4
0.4	0.8	0.9	0.8	0.7	0.6
	0.2	0.01	0.3	0.1	0.2
0.6	0.6	0.1	0.01	0.2	0.05
	0.4	0.9	0.02	0.1	0.01

2.5 Simulation Study

In this section, we generate multiple systems recapture data from a two layer latent class model, then we estimate the population size in three different ways:

1. DP: Dirichlet Process mixture model.

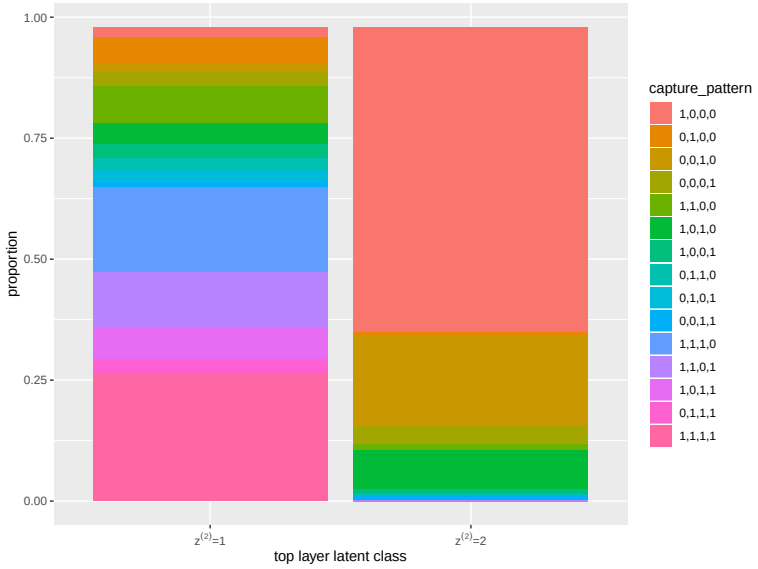
2. Multi-DP: Fit Dirichlet Process mixture model on each top layer latent class which is known in the simulated data, and then sum up population size estimations for those sub-groups to get the overall population size.

3. NDP: Nested Dirichlet Process mixture model.

For the simulated data, we use $S = 4$ data sources, $J = 100$ (e.g. 100 location-times) top layer groups, $N = 10000$, and number of individuals under each top layer group (N_j) ranging from 2 to 602 with a standard deviation of 116. Other parameters for the simulated data are listed in table 3.4. We simulate data by assuming groups within each top layer have similar recording patterns. About 40% of top groups belongs to the first latent class and 60% in the second latent class. From both table 3.4 and figure 2.4, we can see that all four data sources have high capture probabilities and they have many overlapping records when the top layer latent class is 1 ($z^{(2)} = 1$). Individuals are most likely to be captured by the first data source only when they are in the top layer latent class 2 ($z^{(2)} = 2$). Therefore, we can see an obvious nested structure in our simulated data.

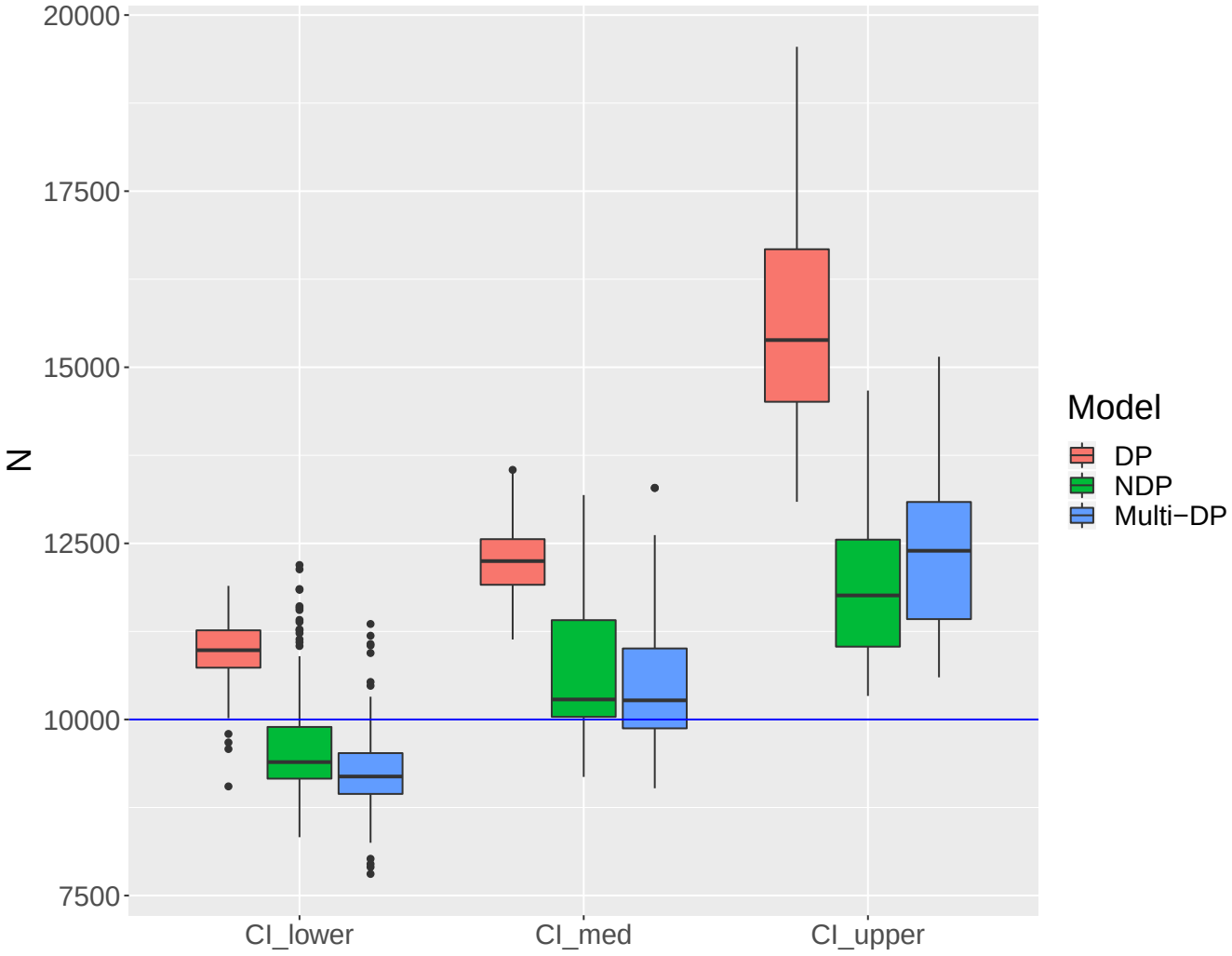
Figure 2.5 is boxplot of posterior estimation of population size under three different

Figure 2.4. Stacked bar-plot of capture pattern proportions by top layer latent class



models: DP, Multi-DP and NDP. The purple line is the true population size. We can see that the estimation in DP (red boxplot) is severely biased. However, if we use DP to estimate the population size separately by the top layer latent classes, the estimation is much better. This result makes sense because the data within each top layer latent class is from a one layer latent class model. If we stratify the data based on the true top layer latent class, we'll get very good population size estimation using DP for each strata, thus ideal estimation for the overall population size. The estimation from NDP (blue boxplot) is very close to the result from the Multi-DP method. This means that NDP successfully detected the capture pattern differences among top level groups and dependency among individuals in the same top layer latent class. We can also see that NDP gives much smaller uncertainty for the population size estimation than NP. All these comparisons strongly suggest the importance of accounting for hierarchical structure in multi-list recapture data when true hierarchy is present.

Figure 2.5. Posterior quantiles for Population size estimation under three different models. These boxplots are based on 100 replicates, the center boxes summarize point estimates, and the left and right sets summarize lower and upper boundaries of the 95% posterior credible intervals.



2.6 Application

From the description in Section 2.3, we know that the Syrian conflict multiple systems recapture data has a hierarchical structure over time and governornate. From the simulation results we know that hierarchical structure is important to consider when estimating the population size. Therefore, we take governornate-time as the top group (or second layer) and individuals as the first layer. Then we apply the Nested Dirichlet Process mixture model to estimate the total number of killings based on sampled Syrian Conflict data. We also apply DP (or LCMCR) and Bayesian model averaging over decomposable graphical models (BMA-DG) to compare the results, which are summarized in Table 2.3.

From Table 2.3 we see that the population size estimation results from the NDP and DP are similar, but NDP has a litter higher estimate, smaller credible interval and higher lower credible interval bound. Model averaging Bayesian decomposable graphical models (BMA-DG) give much smaller estimates.

Figure 2.6 summarizes clustering results of the top layer (location-time). We can see that most records from Tartus, As-Suwayda and Latakia are clustered into the circle class. From Figure 2.1 we know that it is because they have similar recording patterns and most deaths recorded in those governorates were from VDC. If we look at Figure 2.7, larger black triangles are mainly from Tartus, As-Suwayda, Latakia, Quneitra, AI-Hasaka and Ar-Raqqah. From Table 2.4, we can find this group corresponding to the second gov-time layer and the first individual layer with four small list capture probabilities and VDC capture probability (0.39) relatively larger than the other three (0.22 for SNHRS, 0.006 for DCHRS and 0.14 for SCSR). Larger green colored triangles are mainly from the other governorates and from around 09/2012 to 09/2014. Individuals in those groups are much more likely to be captured by VDC, SNHRS, and SCSR together, but not by DCHRS, which is also reflected in Table 2.4 which shows list capture probability less than 0.1 for DCHRS, but more than 0.85 for the

Table 2.3. Estimated number of killings and its 95% credible intervals based on the sampled Syrian Conflict data from 03/2011 to 03/2016

Model	n	$\hat{N}$	$\hat{N}_L$	$\hat{N}_U$
NDP	36226	53405	51605	56094
DP	36226	53069	47389	69848
BMA-DG	36226	38302	36534	43530

Table 2.4. Parameter estimates and 95% credible intervals for the sampled Syrian Conflict data

		List capture probabilities			
gov-time layer prop	individual layer prop	VDC	SNHRS	DCHRS	SCSR
0.47 (0.46,0.48)	0.19 (0.17,0.23)	0.62 (0.54,0.67)	0.93 (0.85,0.99)	0.55 (0.45,0.61)	0.32 (0.25,0.39)
	0.23 (0.21,0.27)	0.88 (0.86,0.90)	0.70 (0.68,0.72)	0.87 (0.90,0.93)	0.90 (0.97,0.99)
	0.57 (0.53,0.60)	0.30 (0.27,0.33)	0.16 (0.11,0.20)	0.17 (0.15,0.19)	0.19 (0.17,0.21)
0.43 (0.42,0.44)	0.59 (0.45,0.61)	0.39 (0.32,0.47)	0.22 (0.18,0.25)	0.006 (0.0005,0.01)	0.14 (0.10,0.17)
	0.02 (0.008,0.15)	0.075 (0.003,0.19)	0.25 (0.16,0.35)	0.44 (0.08,0.96)	0.27 (0.19,0.38)
	0.39 (0.37,0.40)	0.90 (0.89,0.91)	0.88 (0.86,0.90)	0.09 (0.08,0.10)	0.87 (0.84,0.90)

other three data sources. Similarly, we can see from Figure 2.8 that most individuals captured after 09/2014 (larger green circles) are clustered into the first gov-time layer and the third individual layer. In this group, all four data sources have small capture probabilities, which means that many killings are not documented in this period. Individuals under larger black circles are mainly from Latakia, Quenitra, AI-Hasaka and Ar-Raqqah before 06/2012, and they are more likely to be captured by VDC and SNHR, not by the other two data sources. This group corresponding to the first gov-time layer and the first individual layer in Table 2.4. Overall, from our proposed nested Dirichlet process for estimating population size of the capture-recapture data, we are able to group location-times with similar recording patterns into the same top layer latent class and then cluster individuals within it into proper sub-groups within which the probabilities of being captured are similar.

Figure 2.6. Clustering of the location-time group

2.7 Discussion and Conclusions

In this paper, we find similar capture patterns in some governorates and months in the sampled Syrian conflict recapture data, which means that the heterogeneity is clear in this dataset. In order to combine heterogeneity with modeling and allow information sharing across and within strata, we extend the Dirichlet Process mixture model for multi-list capture data to a Nested Dirichlet Process mixture model to estimate population size from multi-list capture data. In clustering problems, NDP is preferred when the data has a hierarchical structure as it allows dependence for objects within the same top layer latent class. In our multi-list capture setting, NDP retains its flexible property and uses heterogeneity from both top groups (e.g. location-time) and individuals to detect better latent classes in the data and thus gives more reasonable population size estimation with smaller uncertainty.

Figure 2.7. Proportion of individuals by the individual layer for each gov-time within the first gov-time layer; colored by individual layer, sized by proportion

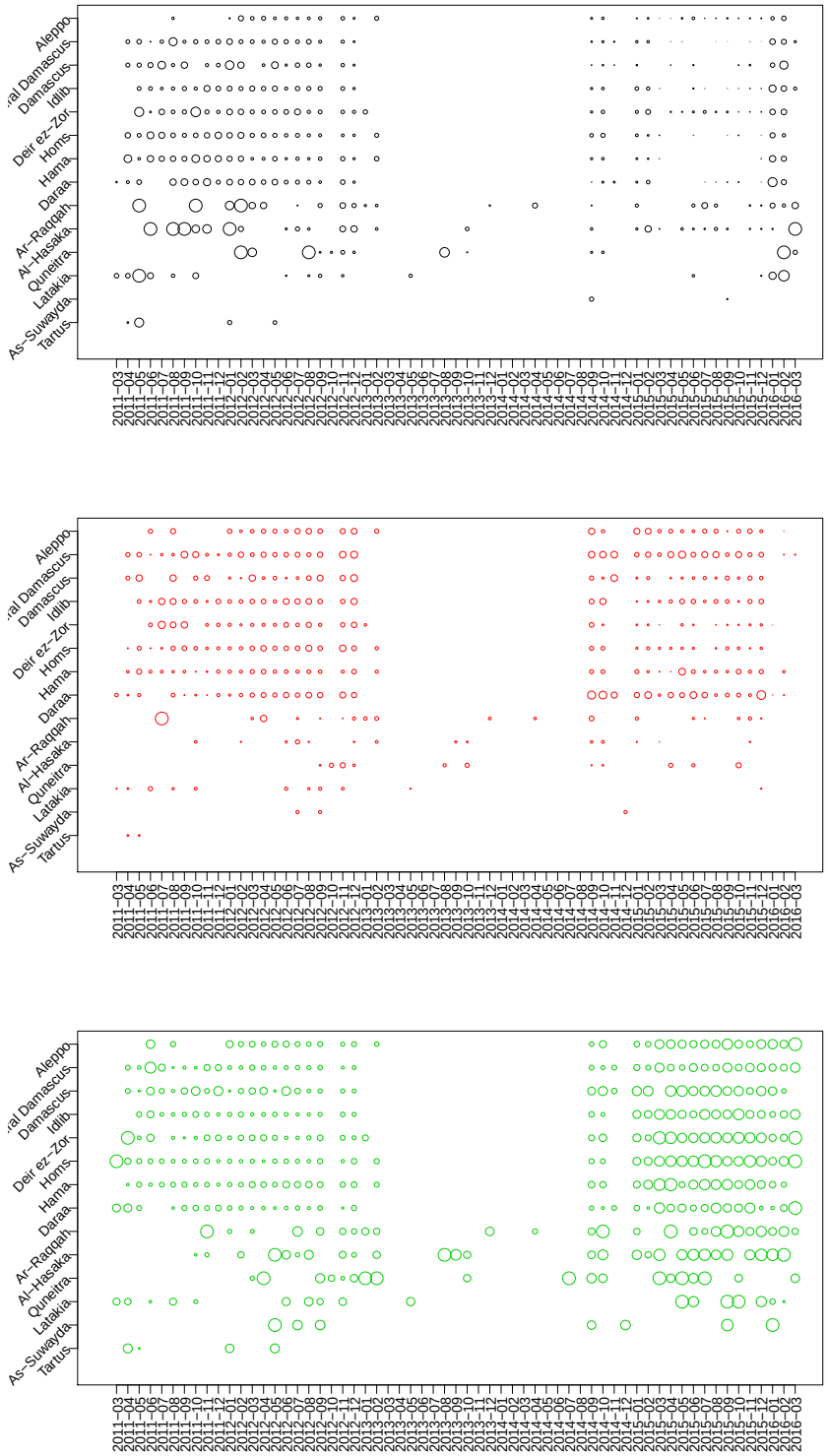
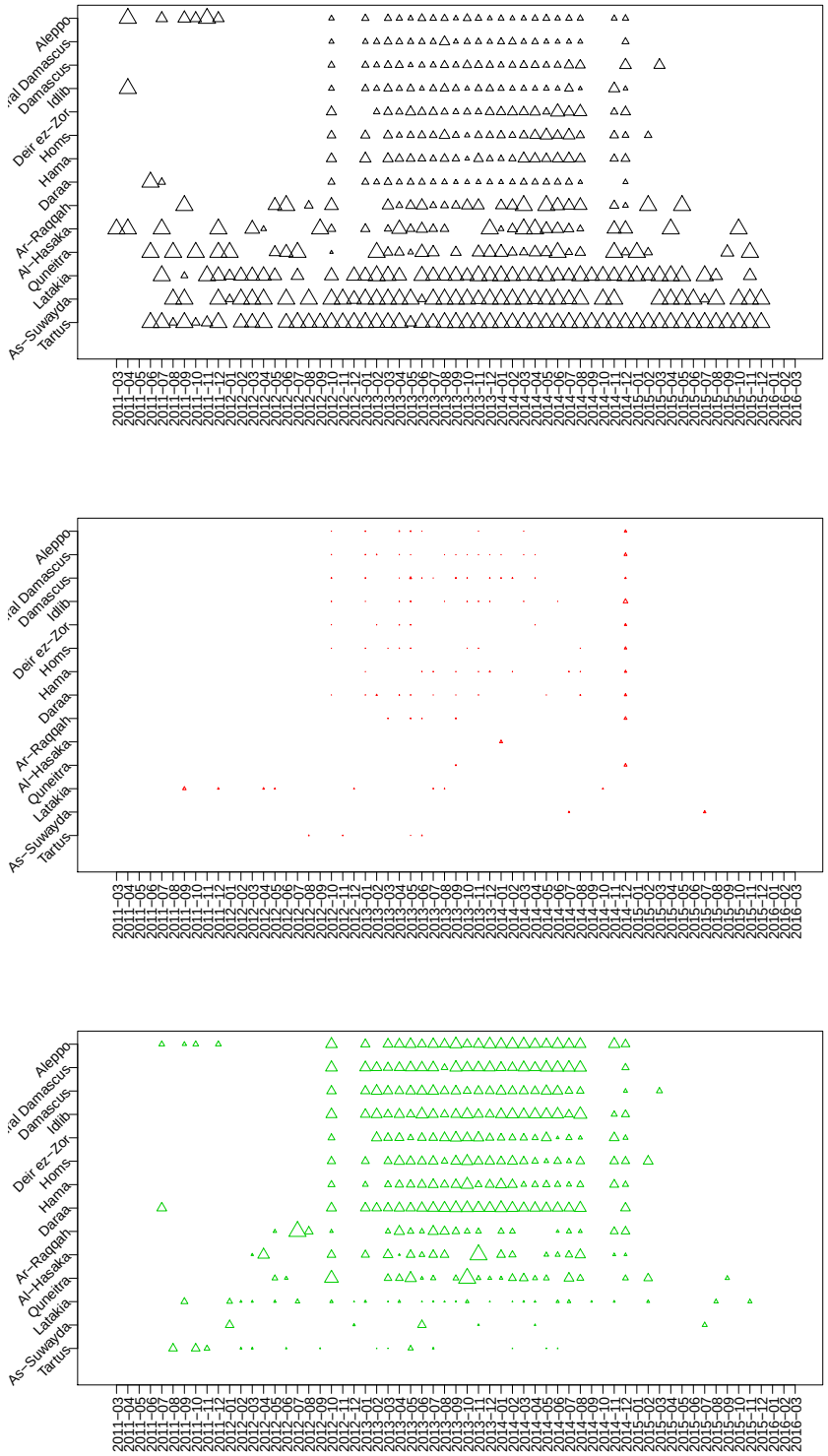


Figure 2.8. Proportion of individuals by the individual layer for each gov-time within the second gov-time layer; colored by individual layer, sized by proportion



CHAPTER 3

BAYESIAN NON-PARAMETRIC LATENT CLASS MODEL FOR POPULATION SIZE ESTIMATION AND MISSING COVARIATE IMPUTATION IN MULTI-SOURCE RECAPTURE DATA

3.1 Abstract

In the Bayesian non-parametric latent class model for multi-list recapture data (LCMCR), different recording patterns across latent classes are used to reflect individual heterogeneity when covariates are not available. In this paper, we add covariates, assuming capture patterns and covariates are independent given the latent classes. In this way, individuals in each latent class are similar in capture patterns and also in covariate distributions. When they have strong association, individual attributes reduce uncertainly of the latent classes and thus uncertainly of the population size estimation. Comparing those latent classes, we can better understand how capture patterns relate with individual characteristics. Meanwhile, there are missing covariate values. We apply data augmentation to impute missing values during MCMC for parameter estimation.

3.2 Introduction

In multi-list recapture data, multiple data sources record partial data from a target population, for example victims in a conflict [Ball et al., 2003][Price et al., 2015], diabetic persons [Seber et al., 2000], patients with alcohol related problems [Corrao et al., 2000], Lethal Violence in Casanare during the Colombian conflict [Guberek

et al., 2020]. To estimate the population size of the target population, we need to estimate the number of un-observed records. Whether a person is captured or not may relate with their characteristics, for example adults are more likely to be recorded than young people, and people living in cities are more likely to be captured than people in rural areas. Therefore, probabilities of being captured are different among individuals. This difference is called individual heterogeneity and models considering it have been well studied in Rasch models, log-linear models, and Bayeisan models [Sanathanan, 1972][Darroch et al., 1993][George and Robert, 1992][Fienberg et al., 1999][Manrique-Vallier, 2016a].

When people's traits are not available, the Bayesian non-parametric latent class model (LCMCR) can be applied to find hidden strata, within which individuals have similar probabilities of being captured or have similar capture patterns, thus reflecting similar individual characteristics. Often, multi-list recapture data has features describing each record, for example, time, location, gender and age. Stratification based on categories of those discrete attributes can be made to reduce bias induced by strata heterogeneity [Guberek et al., 2020][Manrique-Vallier et al., 2013b]. The drawback of this method is that it divides individuals into strata with smaller sample size which may reduce model power and cause identifiability problems [Manrique-Vallier et al.]. Also, this method cannot handle covariate missing values. EM algorithm was proposed for the incomplete stratification problem in multi-list recapture data and used in a Log-linear model with list indicators, categorical variables and their interactions [Sutherland et al., 2007]. However, Log-linear models are not flexible enough to handle complex dependencies. Manrique-Vallier et al. (2019) accommodates the stratification into a flexible model, Bayesian non-parametric latent class model for multi-list recapture data. They assign a discrete distribution to the categorical variable and the individual's capture probability depends on the latent class and value of the attribute [Manrique-Vallier et al.]. In this way, covariate missing values can

be imputed using data augmentation. However, the number of capture probability parameters increases exponentially with the number of attributes and this model does not apply to continuous attributes. In this paper, we propose another way to deal with covariates and missing values in them. We extend the Bayesian non-parametric latent class model for multi-list recapture data (LCMCR) by modeling covariates in parallel with capture pattern variables and assuming they're conditionally independent given the latent class. In this way, latent classes detected by the proposed Bayesian non-parametric latent class model with covariates reflect capture pattern differences and attribute heterogeneity together, based on which we'll have a better idea about which attribute combinations correspond to which capture patterns.

A latent class model for multiple imputation of incomplete categorical data is proposed by Vermunt et al. (2008) and has been used to impute missing values in a life questionnaire [Peyre et al., 2011], and impute missing values in a large-scale survey [Si and Reiter, 2013], etc. It is preferred over log-linear models when the variables have complex association structures and the number of variables is large. Due to differences among lists, among individuals and possible copying between lists, complexity is not rare in the multi-list recapture data problem. Modeling multivariate categorical data using non-parametric Bayesian models is also well studied by Dunson and Xing (2009), which handles attributes with more than two categories and can easily combine with latent class model imputation if there are missing covariate values [Dunson and Xing, 2009]. Even though we focus on categorical variables in this paper, the method can be extended to include continuous variables.

This paper is organized as follows. In Section 3.3, we review the Bayesian method proposed by Manrique-Vallier et al. (2019) and propose our conditionally independent version with covariates. In Section 3.4, we introduce the data augmentation method used for missing covariate imputation and parameter estimation. In Section 3.5, we do simulation to compare the model with and without covariates. We apply

our model to the sampled Syrian conflict data in Section 3.6. In Section 3.7, we discuss our model and make conclusions.

3.3 Bayesian non-parametric latent class model with covariates

The Bayesian non-parametric latent class model for multiple recapture data (LCMCR) finds homogeneous individuals by grouping individuals with similar capture patterns into the same latent class. It infers that individual heterogeneity affects the probability of being captured when individual characteristics are not available. In this paper, we add information on individual characteristics and given by categorical variables. The straightforward way to deal with heterogeneity caused by categorical variables is stratification by their categories. However, this results in too many strata if several categorical variables are available, for example $2^4 = 16$ strata if we have four categorical variables and each has two categories. With so many strata, the sample size in each strata will reduce sharply which may results in identifiability problems [Manrique-Vallier et al., 2019]. Manrique-Vallier et al. (2019) proposed a hierarchical way to impute covariate missing values when one categorical variable is available. It suffers similar problems with stratification because it gives one set of parameters for each category combination. Let

$$y_{i,j} = \begin{cases} 1, & \text{if person } i, \text{ is captured by the } j^{th} \text{ data list} \\ 0, & \text{otherwise,} \end{cases}$$

such that

$$
\begin{aligned}
(y_{i,j}|z_i = k, x_{i,1}, \cdots, x_{i,R}) &\sim \text{Bernoulli}(\lambda_{k,j;x_{i,1},\cdots,x_{i,R}}) \\
z_i &\sim \text{Cat}(\pi_1, \cdots, \pi_k, \cdots) \\
x_i^{(r)} &\sim \text{Cat}(\theta_1^{(r)}, \cdots, \theta_{M_r}^{(r)}) \\
(\theta_1^{(r)}, \cdots, \theta_{M_r}^{(r)}) &\sim \text{Dirichlet}(1, \cdots, 1) \\
\lambda_{k,j;x_{i,1},\cdots,x_{i,R}} &\sim \text{Beta}(1, 1) \\
(\pi_1, \cdots, \pi_k, \cdots) &\sim \text{SB}(\alpha) \\
\alpha &\sim \text{Gamma}(a, b),
\end{aligned}
$$

where

- $\lambda_{k,j;x_{i,1},\cdots,x_{i,R}}$ is the probability of being captured by the j^{th} data source if a individual is in latent class k and has attributes $x_{i,1}, \cdots, x_{i,R}$.
- $x_i^{(r)}$ is the r^{th} covariate with number of categories M_r. $\sum_{m=1}^{M_r} \theta_m^{(r)} = 1$. $r = 1, \cdots, R$, where R is the number of covariates.
- The number of possible values for the capture probability λ is $K * J * \Pi_{r=1}^{R} M_r$ for this model and number of parameters for covariates is $\sum_{r=1}^{R}(M_r - 1)$.
- z_i is the latent class for the i^{th} person. It has a categorical distribution with infinite number of categories, $\sum_{k=1}^{\infty} \pi_k = 1$.
- The latent class proportion parameters $\pi_k, k = 1, \cdots,$ has a stick-breaking prior $(\pi_1, \cdots, \pi_k, \cdots) \sim \text{SB}(\alpha)$, which enables a non-parametric way to learn the number of latent classes from the data. Often, a large K^* is given instead of using ∞. The model will learn the number of latent classes as K $(K < K^*)$ which corresponds to the first K largest latent class proportions. The prior for α is a Gamma distribution, $a = 0.25$ and $b = 0.25$ as usually used in Dirichlet process mixture model [Manrique-Vallier, 2016a][Dunson and Xing, 2009][Si and Reiter, 2013].

In this paper, we add covariates to the Bayesian non-parametric latent class model (LCMCR) by assuming capture patterns and covariates are independent given the latent class. In this way, we put individuals having similar characteristics with homogeneous capture patterns in the same latent class. Then our proposed model is

$$
\begin{aligned}
(y_{i,j}|z_i = k) &\sim \text{Bernoulli}(\lambda_{k,j}) \\
(x_i^{(r)}|z_i = k) &\sim \text{Cat}(\theta_{k,1}^{(r)}, \cdots, \theta_{k,M_r}^{(r)}) \\
z_i &\sim \text{Cat}(\pi_1, \cdots, \pi_k, \cdots) \\
(\theta_{k,1}^{(r)}, \cdots, \theta_{k,M_r}^{(r)}) &\sim \text{Dirichlet}(1, \cdots, 1) \\
\lambda_{k,j} &\sim \text{Beta}(1, 1) \\
(\pi_1, \cdots, \pi_k, \cdots) &\sim \text{SB}(\alpha) \\
\alpha &\sim \text{Gamma}(a, b),
\end{aligned}
$$

where

- $\lambda_{k,j}$ is the probability of captured by the j^{th} data source if an individual is in latent class k.
- $\theta_{k,m}^{(r)}$ is the probability that an individual's r^{th} covariate is m, $x_i^{(r)} = m$, given this individual is in the k^{th} latent class. $\sum_{m=1}^{M_r} \theta_{k,m}^{(r)} = 1$.

In our model, the number of parameters for capture probabilities and covariates is $K * J + K * \sum_{r=1}^{R}(M_r - 1)$ which is much less than those of Manrique-Vallier et al. (2019) when the number of categorical variables or the number of categories is large. This is because this model does not do strict stratification, but combines individuals with similar covariate distributions and having similar recording patterns. By comparing latent classes from this model, we'll know how covariate differences relate with capture pattern differences. Meanwhile, if groups have similar characteristic distributions but different recording patterns, it may also give us information about other

potential factors that influence recording patterns, for example other characteristics not included in the model or factors relating with list dependence.

If we have missing values in the covariates, this conditional independent Bayesian non-parametric latent class model with covariates (LCMCR-cov) still holds by assuming the missing data are MAR and people in the same latent class have the same distribution whether their covariates are observed or not, which is $(x_i^{(r)}|z_i = k) \sim \text{Cat}(\theta_{k,1}^{(r)}, \cdots, \theta_{k,M_r}^{(r)})$ for $r \in \mathcal{O}^i$ or $r \in \mathcal{M}^i$, where $\mathcal{O}^i$ is the observed covariate index set for the i^{th} person, and $\mathcal{M}^i$ is the missing covariate index set for the i^{th} person, $i = 1, \cdots, N$.

3.4 Data Augmentation and Gibbs Sampler

The MCMC algorithm to estimate model parameters and population size in the Bayesian non-parametric latent class model for multi-list recapture data (LCMCR) is developed by Manrique-Vallier (2016). In this paper, we will do two main extensions: update covariate parameters and impute missing covariates, for our Bayesian non-parametric latent class model with missing covariates (LCMCR-cov). Data augmentation is used to impute missing covariates and update parameters. The main idea of data augmentation is to impute missing values and then update parameters based on the observed data and the imputed data.

For the i^{th} person:

$$
\begin{aligned}
&P(x_i^{(r)}, r \in \mathcal{M}_i | z_i = k, y_{i,j}, x_i^{(r')}, r' \in \mathcal{O}_i, j = 1, \cdots, J, \Theta) \\
&\approx P(x_i^{(r)}, r \in \mathcal{M}_i, y_{i,j}, x_i^{(r')}, r' \in \mathcal{O}_{r^i}, j = 1, \cdots, J | z_i = k, \Theta) \\
&= C * P(x_i^{(r)}, r \in \mathcal{M}_i | z_i = k, \theta_{k,1}^{(r)}, \cdots, \theta_{k,M_r}^{(r)})
\end{aligned}
$$

thus, $(x_i^{(r)}, r \in \mathcal{M}_i | z_i = k, ..) \sim \text{Cat}(\theta_{k,1}^{(r)}, \cdots, \theta_{k,M_r}^{(r)})$, where constant

$$
C = P(y_{i,j}, x_i^{(r')}, r' \in \mathcal{O}_i, j = 1, \cdots, J | z_i = k, \Theta)
$$

which does not relate with $\theta_{k,m}^{(r)}$ for $r \in \mathcal{M}_i$ and $m = 1, \cdots, M_r$.

The Gibbs sampler procedures for updating parameters and imputing missing attributes are as follows:

- Impute missing attributes: $(x_i^{(r)}, r \in \mathcal{M}_i | z_i = k, ..) \sim \text{Cat}(\theta_{k,1}^{(r)}, \cdots, \theta_{k,M_r}^{(r)})$, for $i = 1, \cdots, n$. Denote the imputed covariates as X_{imp}.

  Note: The initial values for z's and θ's can be obtained by randomly assign or selection or by running the algorithm with observed covariates only for some iterations, and using its result to initialize the algorithm with imputation [Vidotto et al., 2018].

- Update z_i for $i = 1, \cdots, n$:

$$
\begin{aligned}
&P(z_i = k | y_{i,j}, j = 1, \cdots, J; X_{imp}, ..) \\
&\approx P(y_{i,j}, j = 1, \cdots, J; X_{imp} | z_i = k, ..) P(z_i = k) \\
&= \Pi_{j=1}^{J} \lambda_{k,j}^{y_{i,j}} (1 - \lambda_{k,j})^{1 - y_{i,j}} \Pi_{r=1}^{R} \Pi_{m=1}^{M_r} \theta_{k,m}^{(r)} * I(x_{i,imp}^{(r)} == m) \pi_k
\end{aligned}
$$

- Update list capture parameters $\lambda_{k,j}$: $\lambda_{k,j} \sim \text{Beta}(1 + n_{k,j;1}, 1 + n_{k,j;0} + w_{k,j})$ where $n_{k,j;1} = ||\{i = 1, \cdots, n; z_i = k, y_{i,j} = 1\}||$, $n_{k,j;0} = ||\{i = 1, \cdots, n; z_i = k, y_{i,j} = 0\}||$, and $w_{k,j} = ||\{i = n+1, \cdots, N; z_i = k, y_{i,j} = 0\}||$, where persons $i = 1, \cdots, n$ are captured and persons $i = (n+1), \cdots, N$ are unobserved.

- Update attribute parameters $\theta_{k,m}^{(r)}$ for $m = 1, \cdots, M_r$ and $r = 1, \cdots, R$:

  Similar to $\lambda_{k,j}$,

$$
(\theta_{k,1}^{(r)}, \cdots, \theta_{k,M_r}^{(r)}) \sim \text{Dirichlet}(1 + n_{k,1}^{(r)} + w_{k,1}^{(r)}, \cdots, 1 + n_{k,M_r}^{(r)} + w_{k,M_r}^{(r)})
$$

  , where $n_{k,m}^{(r)} = ||\{i = 1, \cdots, n; z_i = k, x_{i,imp}^{(r)} = m\}||$, and $w_{k,m}^{(r)} = ||\{i = (n+1), \cdots, N; z_i = k, x_i^{(r)} = m\}||$.

- Update $P(\pi_k|...)$:

$$\pi_k = U_k \Pi_{h<k}(1-U_h), k = 1, \cdots, K^*,$$

where $U_k \sim \text{Beta}(1+u_k, \alpha + \sum_{h=k+1}^{K^*} u_k)$, for $k = 1, \cdots, K^*-1$ and $U_{K^*} = 1$.

- Update $P(\alpha|..)$: $\alpha \sim \text{Gamma}(a-1+K^*, b-\log\pi_{K^*})$.

- Update $P(N, w|...)$:

As in Manrique-Vallier (2016), given N has vague prior $P(N) \propto 1/N$, joint distribution of $N, w_1, \cdots, w_{K^*}$ has a negative multinomial distribution. The number of undocumented records $n_0 = N - n$ has a negative binomial distribution:

$$n_0 \sim \text{NegBinomial}(n, 1 - \sum_{k=1}^{K^*} \rho_k),$$

where $\rho_k = \pi_k \Pi_{j=1}^{J}(1-\lambda_{k,j})$.

In each latent class, the number of unobserved people:

$$(w_1, \cdots, w_{K^*}) \sim \text{Multinomial}(n_0; p_1, \cdots, p_{K^*}), p_k \propto \rho_k.$$

In the k^{th} latent class, the number of unobserved people with different r^{th} attribute values is:

$$(w_{k,1}^{(r)}, \cdots, w_{k,M_r}^{(r)}) \sim \text{Multinomial}(w_k; \theta_{k,1}^{(r)}, \cdots, \theta_{k,M_r}^{(r)}),$$

where $k = 1, \cdots, K^*$, $m = 1, \cdots, M_r$ and $r = 1, \cdots, R$.

3.5 Simulation Study

In this section, we generate 100 replicates from the latent class model with covariates. The population size $N = 5000$, and other parameters are listed in Tables 3.1 and

Table 3.1. Parameters for recapture latent class model with covariates

	List capture probabilities				covariate parameters	
π	list 1	list 2	list 3	list 4	$\theta^{(1)}_{1:K,1:V[1]}$	$\theta^{(2)}_{1:K,1:V[2]}$
0.1	0.01	0.05	0.15	0.1	(0.7 ,0.3)	(0.8,0.2)
0.3	0.9	0.3	0.4	0.7	(0.1 ,0.9)	(0.6,0.4)
0.6	0.1	0.4	0.05	0.01	(0.6,0.4)	(0.1,0.9)

Table 3.2. Missing proportions by recording patterns

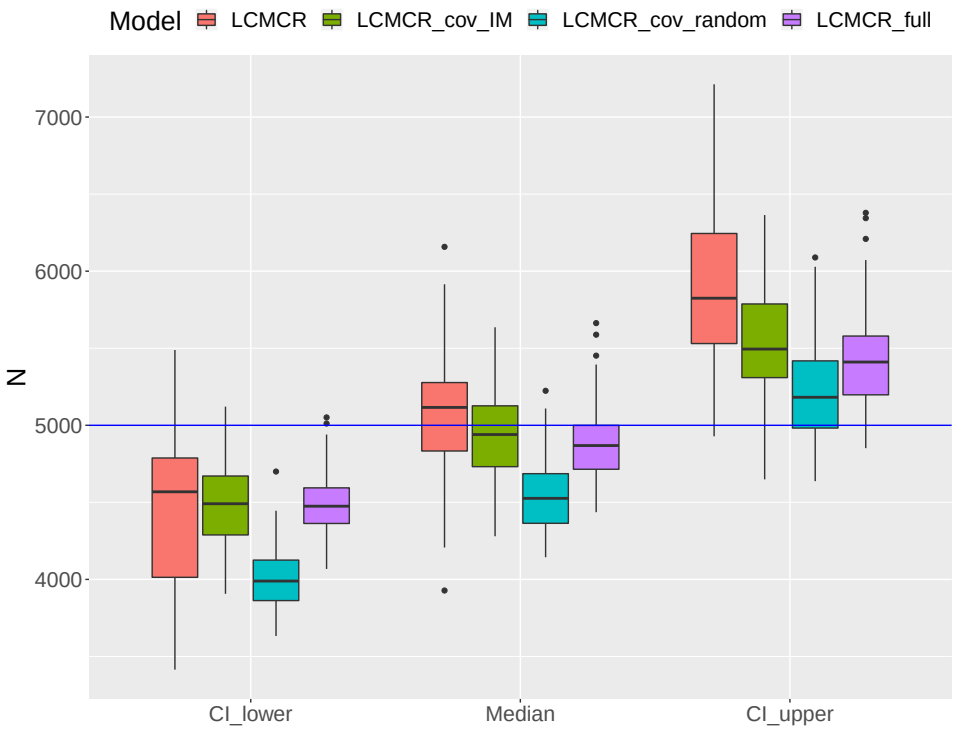
	missing proportions by recording pattern														
	1000	0100	0010	0001	1100	1010	1001	0110	0101	0011	1110	1101	1011	0111	1111
$X^{(1)}$	0	0.04	1	1	0	0.51	0.28	0.47	0.39	1.00	0.11	0.14	0.60	0.79	0.22
$X^{(2)}$	0	0.034	0	0.004	0	0	0	0	0	0	0	0	0	0	0

3.2. We generate missing covariates based on recording patterns so that the missing data are MAR. $X^{(1)}$ has about 40% missing values and $X^{(2)}$ has a small number of missing values. Then we use the Bayesian non-parametric latent class models with and without covariates to estimate the population size. From figure 3.1 we can see that our model LCMCR-cov-IM, Bayesian non-parametric latent class model for multi-list recapture data with missing covariates, gets population size estimation with less bias and uncertainty than models LCMCR and LCMCR-cov-random (missing values are imputed randomly). The model fitting covariates without missing values (LCMCR-full) gives the best estimation which makes sense because there is no bias and errors induced by missing covariates. The estimation from our model LCMCR-cov-IM is much closer to the result got from LCMCR-full. Those indicate that LCMCR-cov-IM helps estimate population size in multi-list recapture data through latent class imputation.

3.6 Application

The Syrian conflict data contains deaths during the Syrian conflict. Those deaths are recorded by four data centers, Violations Documentation Center (VDC), Syrian Network for Human Rights (SNHR), Damascus Center for Human Rights Studies (DCHRS), and Syrian Center for Statistics and Research (SCSR). We take a subset of

Figure 3.1. Population size estimation using Bayesian non-parametric latent class model with and without covariates. LCMCR: use LCMCR model for data without covariates; LCMCR-cov-missing: use LCMCR-cov model for data with missing covariate values; LCMCR-cov-full: use LCMCR-cov model for full data or data without missing values.



the Syrian conflict data for application in this paper due to confidentiality. This subset data is death records from governorates Damascus, Rural Damascus, Latakia and Tartus in 2014. The total number of death records is $n = 10412$ in our Syrian conflict sample data and the number of records under each recording pattern is summarized in Table 3.3. Each death record also has covariates: gender, age-group (adult or child), civilian status (Civilian or Military), and under-torture (Yes or No). As we can see from Figure 3.2, among documented records, about $\frac{7014}{10412} * 100\% = 67.4\%$ are missing under-torture values, about $\frac{5277}{10412} * 100\% = 50.7\%$ are missing age-group information, $\frac{1839}{10412} = 17.7\%$ are missing civilian status and only $\frac{26}{10412} * 100\% = 0.2\%$ do not have gender values. For the under-torture and age-group variables, most missing values come from the $'1, 0, 0, 0'$ recording pattern which corresponds to captured by the VDC only. For the civilian status attribute, most missing values come from death records only captured by the SCSR. Therefore, missing covariates relate with capture patterns. We assume the missing pattern is MAR that $P(x_i^{(r)}, r \in \mathcal{M}_i | y_{i,j=1,\cdots,J}, x_i^{(r')}, r' \in \mathcal{O}_i, \text{other factors}) = P(x_i^{(r)}, r \in \mathcal{M}_i | y_{i,j=1,\cdots,J}, x_i^{(r')}, r' \in \mathcal{O}_i)$. Due to the high missing proportion (67.4%) for under-torture, we don't use this covariate in the model.

As we can see from the estimation, the LCMCR with covariates (LCMCR-cov) gives a slightly larger estimate than the Benchmark Bayeisan model averaging of decomposable graphical models (BMA-DG) and the Bayeisan non-parametric latent class model (LCMCR). Moreover, Table 3.5 gives us list capture probabilities and covariate parameters in each latent class based on the LCMCR-cov model, as we can see that the first class mainly captures Male, Military and Adults and VDC has higher capture ability than other data sources in this group of people. Deaths captured in the second class are mainly Male, Adults and about 60% are civilians. The capture probabilities for VDC, SNHR, and SCSR are all pretty high. Deaths in the third class are mostly Male, Civilian, and Adults. Comparing the first three classes, we see that civilians are more likely to be captured by SNHR and military are more

Table 3.3. Number of deaths under each recording pattern in the Syrian conflict sample data

VDC	SNHR	DCHRS	SCSR	Num-Records
1	0	0	0	$n_{1000} = 2820$
0	1	0	0	$n_{0100} = 814$
0	0	1	0	$n_{0010} = 123$
0	0	0	1	$n_{0001} = 870$
1	1	0	0	$n_{1100} = 466$
1	0	1	0	$n_{1010} = 48$
1	0	0	1	$n_{1001} = 826$
0	1	1	0	$n_{0110} = 63$
0	1	0	1	$n_{0101} = 423$
0	0	1	1	$n_{0011} = 56$
1	1	1	0	$n_{1110} = 72$
1	1	0	1	$n_{1101} = 2750$
1	0	1	1	$n_{1011} = 158$
0	1	1	1	$n_{0111} = 82$
1	1	1	1	$n_{1111} = 841$
0	0	0	0	$n_{0000} =?$

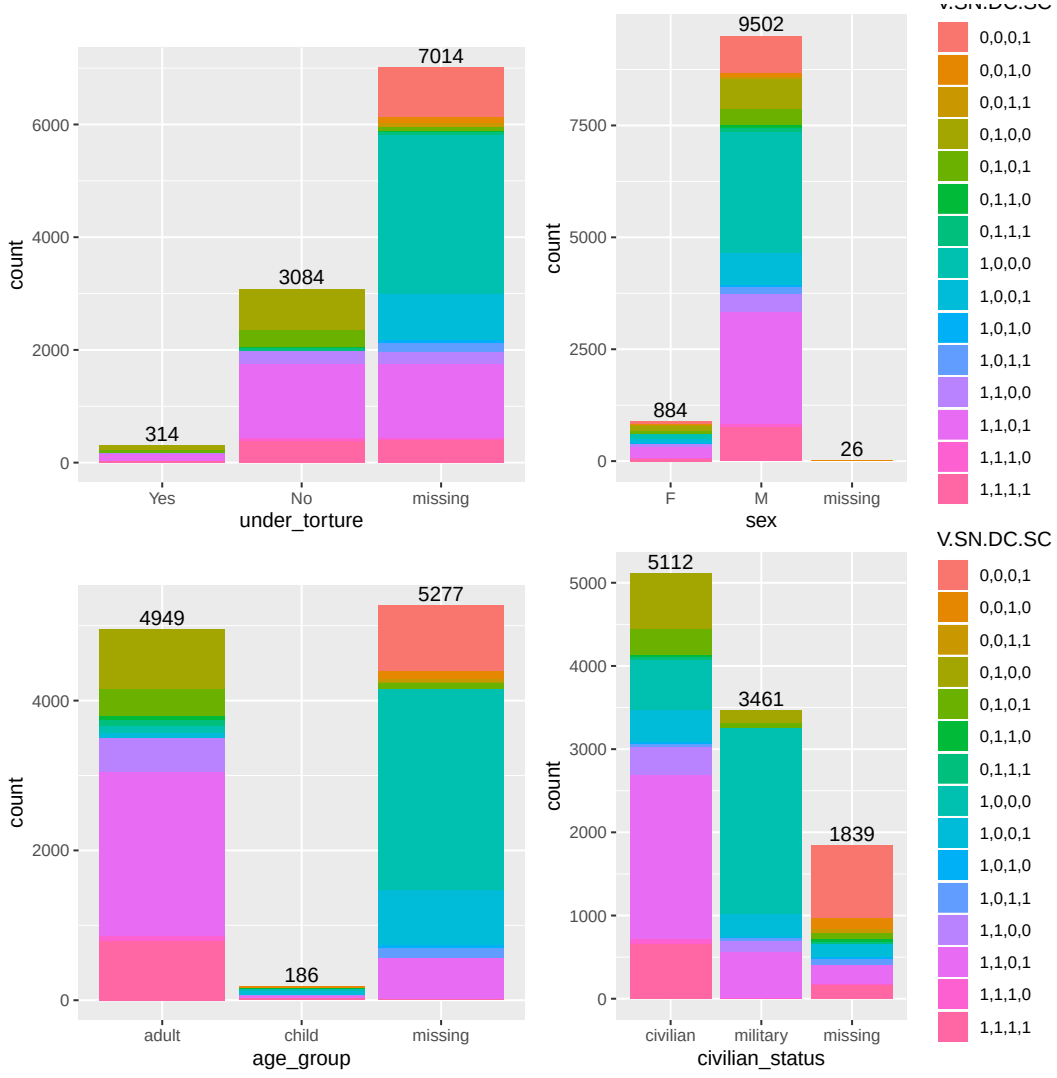
Table 3.4. Estimated total number of deaths and 95% posterior credible intervals in the Syrian conflict sample data

	n	$\hat{N}$	$\hat{N}_L$	$\hat{N}_U$
LCMCR	10412	16591	14039	19615
LCMCR-cov	10412	17903	15454	22481
BMA-DG	10412	17684	14821	20484

Table 3.5. Estimation and 90% credible intervals of latent class proportions, list capture probabilities and covariate proportions

	List capture probabilities				covariate parameters		
π	VDC	SNHR	DCHRS	SCSR	$\theta^{(sex)}_{1:5,\text{Female}}$	$\theta^{(civilian)}_{1:5,\text{Civilian}}$	$\theta^{(age-group)}_{1:5,\text{Adults}}$
0.55	0.27	0.27	0.0008	0.07	0.0009	0.12	0.99
(0.48,0.63)	(0.19,0.35)	(0.01,0.03)	(0.0001,0.003)	(0.05,0.08)	(0.0001,0.003)	(0.08,0.14)	(0.98,0.9995)
0.15	0.92	0.89	0.005	0.89	0.0021	0.69	0.997
(0.06,0.19)	(0.90,0.94)	(0.85,0.98)	(0.0004,0.016)	(0.85,0.92)	(0.0002,0.007)	(0.26,0.75)	(0.992,0.9996)
0.07	0.15	0.84	0.05	0.23	0.19	0.985	0.998
(0.05,0.10)	(0.04,0.27)	(0.51,0.98)	(0.004,0.08)	(0.10,0.31)	(0.16,0.22)	(0.96,0.996)	(0.993,0.9998)
0.06	0.93	0.98	0.89	0.94	0.07	0.997	0.96
(0.04,0.075)	(0.91,0.96)	(0.94,0.998)	(0.70,0.99)	(0.91,0.96)	(0.04,0.09)	(0.990,0.9998)	(0.94,0.97)
0.05	0.94	0.91	0.04	0.92	0.36	0.995	0.995
(0.02,0.14)	(0.86,0.99)	(0.85,0.99)	(0.004,0.15)	(0.84,0.99)	(0.13,0.91)	(0.98,0.9995)	(0.97,0.9996)

Figure 3.2. Number of death record by covariate and stacked by recording pattern; Four covariates: under-torture, sex, age-group and civilian-status. The missing category represents missing values



likely to be captured by VDC. The fourth and fifth latent classes capture more child deaths. VDC and SCSR have relatively higher capture probabilities in these clusters. Even though covariate distributions are similar in latent classes two and four, the list capture proabilities differ very much. This tells us that other heterogeneity besides those three attributes impacts list captures. It is possible, for example, that copying patterns between data sources create different capture patterns over time.

3.7 Discussion and Conclusions

In this paper, we extend the Bayesian non-parametric latent class model for multi-list recapture data by adding individual attributes into the model and assuming conditional independence between capture patterns and individual covariates given the latent class. The latent class detects attribute differences and capture patterns relating with it. Comparing latent classes, we are able to see which data sources capture which groups of people. Meanwhile, our model handles missing values in covariates through data augmentation by assuming MAR. In this paper, we focus on categorical variables with a small number of categories only. We can easily extend it to continuous variables, for example if we use numerical age instead of the age-group category, we can assign a mixture of normals or truncated normal distribution to it. If we have categorical variables with many categories, such as time and location, we recommend use the Nested Bayesian non-parametric latent class model, grouping by time and location as the top (or second) layer and modelings variables with less categories, like sex or civilian-status, in the first layer together with the capture pattern variables. These methods help understand individual attribute-related heterogeneity and how attributes relate with capture patterns. We can see from our application that there are other potential sources of heterogeneity besides individual attributes. List heterogeneity and other factors, like copying among lists, may also cause cap-

ture pattern differences. Adding those sources of heterogeneity into the model is a potential direction for further work.

BIBLIOGRAPHY

Alan Agresti. Simple capture-recapture models permitting unequal catchability and variable sampling effort. *Biometrics*, 50(2):494–500, 1994. ISSN 0006341X, 15410420. URL http://www.jstor.org/stable/2533391.

Edo M Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, June 2008. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1390681.1442798.

Patrick Ball, Jana Asher, David Sulmont, and Daniel Manrique. How many Peruvians have died?, 01 2003.

Seema Bandyopadhyay and Edward Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. volume 3, pages 1713 – 1723 vol.3, 04 2003. ISBN 0-7803-7752-4. doi: 10.1109/INFCOM.2003.1209194.

Sanjib Basu and Nader Ebrahimi. Bayesian capture-recapture methods for error detection and estimation of population size: Heterogeneity and dependence. *Biometrika*, 88(1):269–279, 2001. ISSN 00063444. URL http://www.jstor.org/stable/2673684.

N. Binkiewicz, J. T. Vogelstein, and K. Rohe. Covariate-assisted spectral clustering. *Biometrika*, 104(2):361–377, 03 2017. ISSN 0006-3444. doi: 10.1093/biomet/asx008. URL https://doi.org/10.1093/biomet/asx008.

Yvonne M. Bishop, Paul W. Holland, and Stephen E. Fienberg. Discrete multivariate analysis: Theory and practice. 1975.

David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *J. ACM*, 57:7:1–7:30, 2007.

David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *J. ACM*, 57(2), February 2010. ISSN 0004-5411. doi: 10.1145/1667053.1667056. URL https://doi.org/10.1145/1667053.1667056.

Anne Chao and P. K. Tsay. A sample coverage approach to multiple-system estimation with application to census undercount. *Journal of the American Statistical Association*, 93(441):283–293, 1998. doi: 10.1080/01621459.1998.10474109. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1998.10474109.

Yiu-ming Cheung. Maximum weighted likelihood via rival penalized em for density mixture clustering with automatic model selection. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):750–761, 2005.

G Corrao, V Bagnardi, G Vittadini, and S Favilli. Capture-recapture methods to size alcohol related problems in a population. *Journal of epidemiology and community health*, 54:603–10, 09 2000. doi: 10.1136/jech.54.8.603.

John N. Darroch, Stephen E. Fienberg, Gary F. V. Glonek, and Brian W. Junker. A three-sample multiple-recapture approach to census population estimation with heterogeneous catchability. *Journal of the American Statistical Association*, 88(423):1137–1148, 1993. doi: 10.1080/01621459.1993.10476387. URL https://amstat.tandfonline.com/doi/abs/10.1080/01621459.1993.10476387.

Jean-Jacques Daudin, Franck Picard, and Stéphane Robin. A mixture model for random graphs. *Statistics and Computing*, 18(2):173–183, Jun 2008. ISSN 1573-1375. doi: 10.1007/s11222-007-9046-7. URL https://doi.org/10.1007/s11222-007-9046-7.

D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

Chris Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 107–114, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-1119-8. URL http://dl.acm.org/citation.cfm?id=645496.658058.

J. C. Dunn†. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4(1):95–104, 1974. doi: 10.1080/01969727408546059. URL https://doi.org/10.1080/01969727408546059.

David B. Dunson and Chuanhua Xing. Nonparametric Bayes modeling of multivariate categorical data. *Journal of the American Statistical Association*, 104(487):1042–1051, 2009. doi: 10.1198/jasa.2009.tm08439. URL https://doi.org/10.1198/jasa.2009.tm08439.

S. E. Fienberg, M. S. Johnson, and B. W. Junker. Classical multilevel and Bayesian approaches to population size estimation using multiple lists. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 162(3):383–405, 1999. doi: 10.1111/1467-985X.00143. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-985X.00143.

Emily B. Fox, Erik B. Sudderth, Michael I. Jordan, and Alan S. Willsky. A sticky HDP-HMM with application to speaker diarization. *The Annals of Applied Statistics*, 5(2A):1020–1056, 2011. ISSN 19326157. URL http://www.jstor.org/stable/23024915.

Edward I. George and Christian P. Robert. Capture-recapture estimation via Gibbs sampling. *Biometrika*, 79(4):677–683, 1992. ISSN 00063444. URL http://www.jstor.org/stable/2337223.

Krista J. Gile and Mark S. Handcock. Respondent-driven sampling: An assessment of current methodology. *Sociological Methodology*, 40 (1):285–327, 2010. doi: 10.1111/j.1467-9531.2010.01223.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9531.2010.01223.x.

Tamy Guberek, Daniel Guzmán, Megan Price, Kristian Lum, and Patrick Ball. To count the uncounted: An estimation of lethal violence in Casanare. 03 2020.

Greg Hamerly and Charles Elkan. Alternatives to the k-means algorithm that find better clusterings. pages 600–607, 01 2002. doi: 10.1145/584792.584890.

Douglas Heckathorn. Respondent-driven sampling: A new approach to the study of hidden populations. *Social Problems*, 44(2):174–199, 1997. ISSN 00377791, 15338533. URL http://www.jstor.org/stable/3096941.

Peter G.M. Heijden. Multiple systems estimation for estimating the number of victims of human trafficking across the world. 06 2016.

Hemant Ishwaran and Lancelot F James. Gibbs sampling methods for Stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001. doi: 10.1198/016214501750332758. URL https://doi.org/10.1198/016214501750332758.

Hemant Ishwaran and Lancelot F. James. Approximate Dirichlet process computing in finite normal mixtures: Smoothing and prior information. *Journal of Computational and Graphical Statistics*, 11(3):508–532, 2002. ISSN 10618600. URL http://www.jstor.org/stable/1391111.

Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Phys. Rev. E*, 83:016107, Jan 2011. doi: 10.1103/PhysRevE.83.016107. URL https://link.aps.org/doi/10.1103/PhysRevE.83.016107.

Stephen Kobourov, Sergey Pupyrev, and Paolo Simonetto. Visualizing Graphs as Maps with Contiguous Regions. In N. Elmqvist, M. Hlawitschka, and J. Kennedy, editors, *EuroVis - Short Papers*. The Eurographics Association, 2014. ISBN 978-3-905674-69-9. doi: 10.2312/eurovisshort.20141153.

Bjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 16–22, New York, NY, USA, 1999. Association for Computing Machinery. ISBN 1581131437. doi: 10.1145/312129.312186. URL https://doi.org/10.1145/312129.312186.

Zhiguo Li, Peter Gilbert, and Bin Nan. Weighted likelihood method for grouped survival data in case–cohort studies with application to HIV vaccine trials. *Biometrics*, 64(4):1247–1255, 2008. doi: 10.1111/j.1541-0420.2008.00998.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1541-0420.2008.00998.x.

David Madigan, Jeremy York, and Denis Allard. Bayesian graphical models for discrete data. *International Statistical Review / Revue Internationale de Statistique*, 63(2):215–232, 1995. ISSN 03067734, 17515823. URL http://www.jstor.org/stable/1403615.

Daniel Manrique-Vallier. Bayesian population size estimation using Dirichlet process mixtures. *Biometrics*, 72(4):1246–1254, 2016a. doi: 10.1111/biom.12502. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/biom.12502.

Daniel Manrique-Vallier. Bayesian population size estimation using Dirichlet process mixtures. *Biometrics*, 72(4):1246–1254, 2016b. doi: 10.1111/biom.12502. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/biom.12502.

Daniel Manrique-Vallier and Stephen E. Fienberg. Population size estimation using individual level mixture models. *Biometrical Journal*, 50(6):1051–1063, 2008. doi: 10.1002/bimj.200810448. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/bimj.200810448.

Daniel Manrique-Vallier, Patrick Ball, and David Sulmont. Estimating the number of fatal victims of the Peruvian internal armed conflict, 1980-2000: an application of modern multi-list Capture-Recapture techniques.

Daniel Manrique-Vallier, Megan Price, and Anita Gohdes. *Multiple Systems Estimation Techniques for Estimating Casualties in Armed Conflicts*, pages 165–181. 06 2013a. ISBN 9780199977307. doi: 10.1093/acprof:oso/9780199977307.003.0009.

Daniel Manrique-Vallier, Megan Price, and Anita R. Gohdes. Multiple systems estimation techniques for estimating casualties in armed conflicts. 2013b.

Daniel Manrique-Vallier, Patrick Ball, and Mauricio Sadinle. Capture-recapture for casualty estimation and beyond: Recent advances and research directions. 2019.

Marianthi Markatou. Mixture models, robustness, and the weighted likelihood methodology. *Biometrics*, 56(2):483–486, 2000. doi: 10.1111/j.0006-341X.2000.00483.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.0006-341X.2000.00483.x.

M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006. ISSN 0027-8424. doi: 10.1073/pnas.0601602103. URL https://www.pnas.org/content/103/23/8577.

A.Y. Ng, Michael Jordan, and Y Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pages 849–856, Cambridge, MA, USA, 2001. MIT Press. URL http://dl.acm.org/citation.cfm?id=2980539.2980649.

Krzysztof Nowicki and Tom A. B Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001. doi: 10.1198/016214501753208735. URL https://doi.org/10.1198/016214501753208735.

Miles Ott and Krista Gile. Unequal edge inclusion probabilities in link-tracing network sampling with implications for Respondent-driven sampling. *Electronic Journal of Statistics*, 10:1109–1132, 01 2016. doi: 10.1214/16-EJS1138.

Antony M. Overstall, Ruth King, Sheila M. Bird, Sharon J. Hutchinson, and Gordon Hay. Incomplete contingency tables with censored cells with application to estimating the number of people who inject drugs in Scotland. *Statistics in Medicine*, 33(9):1564–1579, 2014. doi: 10.1002/sim.6047. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.6047.

Hugo Peyre, Alain Leplège, and Joël Coste. Missing data methods for dealing with missing items in quality of life questionnaires. a comparison by simulation of personal mean score, full information maximum likelihood, multiple imputation, and hot deck techniques applied to the sf-36 in the french 2003 decennial health survey. *Quality of Life Research*, 20(2):287–300, 2011. ISSN 09629343, 15732649. URL http://www.jstor.org/stable/41488074.

Megan Price, Anita Gohdes, and Patrick Ball. Documents of war: Understanding the Syrian conflict. *Significance*, 12(2): 14–19, 2015. doi: 10.1111/j.1740-9713.2015.00811.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1740-9713.2015.00811.x.

Guo-jun Qi Qi, Charu Aggarwal, and Thomas Huang. Community detection with edge content in social media networks. In *2012 IEEE 28th International Conference on Data Engineering*, pages 534–545, April 2012. doi: 10.1109/ICDE.2012.77.

G.E. Rasch. *Probabilistic Models for Some Intelligence and Attainment Tests*, volume 1. 01 1993.

Eréndira Rendón, Itzel Abundez, A. Arizmendi, and E.M. Quiroz. Internal versus external cluster validation indexes. *International Journal of Computers and Communications*, 5:27–34, 01 2011.

Abel Rodrlguez, David B Dunson, and Alan E Gelfand. The nested Dirichlet process. *Journal of the American Statistical Association*, 103(483):1131–1154, 2008. doi: 10.1198/016214508000000553. URL https://doi.org/10.1198/016214508000000553.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987. ISSN 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7. URL http://www.sciencedirect.com/science/article/pii/0377042787901257.

Takumi Saegusa and Jon A. Wellner. Weighted likelihood estimation under two-phase sampling. *Ann. Statist.*, 41(1):269–295, 02 2013. doi: 10.1214/12-AOS1073. URL https://doi.org/10.1214/12-AOS1073.

Lalitha Sanathanan. Estimating the size of a multinomial population. *The Annals of Mathematical Statistics*, 43, 02 1972. doi: 10.1214/aoms/1177692709.

Satu Schaeffer. Graph clustering. *Computer Science Review*, 1:27–64, 08 2007. doi: 10.1016/j.cosrev.2007.05.001.

George A. F. Seber, John T. Huakau, and David Simmons. Capture-recapture, epidemiology, and list mismatches: Two lists. *Biometrics*, 56(4):1227–1232, 2000. doi: 10.1111/j.0006-341X.2000.01227.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.0006-341X.2000.01227.x.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 05 2002. doi: 10.1109/34.868688.

Motoki Shiga, Ichigaku Takigawa, and Hiroshi Mamitsuka. A spectral clustering approach to optimally combining numericalvectors with a modular network. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, page 647–656, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595936097. doi: 10.1145/1281192.1281262. URL https://doi.org/10.1145/1281192.1281262.

Yajuan Si and Jerome P. Reiter. Nonparametric Bayesian multiple imputation for incomplete categorical variables in large-scale assessment surveys. *Journal of Educational and Behavioral Statistics*, 38(5):499–521, 2013. doi: 10.3102/1076998613480394. URL https://doi.org/10.3102/1076998613480394.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3(null): 583–617, March 2003. ISSN 1532-4435. doi: 10.1162/153244303321897735. URL https://doi.org/10.1162/153244303321897735.

Jason M. Sutherland, Carl James Schwarz, and Louis-Paul Rivest. Multilist population estimation with incomplete and partial stratification. *Biometrics*, 63(3):910–916, 2007. doi: 10.1111/j.1541-0420.2007.00767.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1541-0420.2007.00767.x.

Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006. doi: 10.1198/016214506000000302. URL https://doi.org/10.1198/016214506000000302.

Alexander Topchy, Behrouz Minaei, Anil Jain, and William Punch. Adaptive clustering ensembles. volume 1, pages 272–275, 01 2004. doi: 10.1109/ICPR.2004.1334105.

Jeroen K. Vermunt. 7. multilevel latent class models. *Sociological Methodology*, 33(1):213–239, 2003. doi: 10.1111/j.0081-1750.2003.t01-1-00131.x. URL https://doi.org/10.1111/j.0081-1750.2003.t01-1-00131.x.

Davide Vidotto, Jeroen K. Vermunt, and Katrijn Van Deun. Bayesian multilevel latent class models for the multiple imputation of nested categorical data. *Journal of Educational and Behavioral Statistics*, 43(5):511–539, 2018. doi: 10.3102/1076998618769871. URL https://doi.org/10.3102/1076998618769871.

Zhiqiang Xu, Yiping Ke, Yi Wang, Hong Cheng, and James Cheng. A model-based approach to attributed graph clustering. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pages 505–516, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1247-9. doi: 10.1145/2213836.2213894. URL http://doi.acm.org/10.1145/2213836.2213894.

Jaewon Yang, Julian McAuley, and Jure Leskovec. Community detection in networks with node attributes. In *2013 IEEE 13th International Conference on Data Mining*, pages 1151–1156, Dec 2013. doi: 10.1109/ICDM.2013.167.

Bin Zhang. Generalized K-Harmonic means – dynamic weighting of data in unsupervised learning. 04 2001. doi: 10.1137/1.9781611972719.6.