

# Random Weighting in LASSO Regression and in Discrete Mixture Models

by

Tun Lee Ng

A dissertation submitted in partial fulfillment of  
the requirements for the degree of

Doctor of Philosophy

(Statistics)

at the

UNIVERSITY OF WISCONSIN-MADISON

2022

Date of final oral examination: 5/5/2022

The dissertation is approved by the following members of the Final Oral Committee:

Michael A. Newton, Professor, Statistics

Christina Kendzierski, Professor, Biostatistics and Medical Informatics

Vikas Singh, Professor, Biostatistics and Medical Informatics

Qiongshi Lu, Assistant Professor, Biostatistics and Medical Informatics

Sameer Deshpande, Assistant Professor, Statistics

© Copyright by Tun Lee Ng 2022

All Rights Reserved

# Acknowledgments

First, I want to express my deep gratitude to my advisor Professor Michael A. Newton for his expert guidance and mentoring. The vast majority of this thesis has grown out of regular discussions with Professor Newton. I am inspired by his passion for research and his many deep insights in statistical methods. I am also indebted to his character reference and his consistent funding support, which have led to exciting collaborative opportunities in several applied statistical genomics projects with Oncology faculty Professors James Shull and Douglas McNeel and their respective lab members. It is my honor and pleasure to work with Professor Newton and the PIs during my Ph.D study in Madison, WI.

I am also thankful to my committee members Drs. Christina Kendziorski, Vikas Singh, Qiongshi Lu and Sameer Deshpande for their many helpful comments and insights.

Special thanks are due to Professors Michael A. Newton and Christina Kendziorski for their encouragement and emotional support during the difficult COVID-19 lockdown period.

My appreciation also extends to Drs. Naijun Sha, Ori Rosen, Panagis Moschopoulos and Shuanming Li for their past guidance that has helped me in pursuing Statistics graduate study in the United States.

Finally, I want to thank my family and friends for their unconditional love and unwavering support throughout all these years, which have meant the world to me.

# Contents

Contents . . . . .	ii
List of Tables . . . . .	iv
List of Figures . . . . .	v
Abstract . . . . .	viii
<b>1</b> Introduction . . . . .	1
<b>2</b> Random Weighting in LASSO Regression . . . . .	5
2.1 <i>Preamble</i> . . . . .	5
2.2 <i>Problem Setup</i> . . . . .	8
2.3 <i>Main Results</i> . . . . .	13
2.4 <i>Numerical Experiments</i> . . . . .	24
2.5 <i>Discussion</i> . . . . .	37
<b>3</b> Technical Details for Chapter 2 . . . . .	43
<b>4</b> Random Weighting in Discrete Mixture Models . . . . .	82
4.1 <i>Framework</i> . . . . .	82
4.2 <i>Methodology</i> . . . . .	88
4.3 <i>Numerical Experiments</i> . . . . .	101
4.4 <i>Theoretical Properties</i> . . . . .	114

<b>5</b>	<b>Supplementary Material for Chapter 4</b>	<b>125</b>
5.1	<i>Implementation details of DP-rich</i>	125
5.2	<i>Additional details of RW SDP-rich</i>	127
5.3	<i>Additional details of RW K-means</i>	136
5.4	<i>Additional details for theoretical properties</i>	137
5.5	<i>Additional information for numerical experiments</i>	145
5.6	<i>Variational Inference</i>	153
	<b>References</b>	<b>159</b>

# List of Tables

2.1	Simulation Settings . . . . .	26
2.2	Empirical coverage $\hat{q}_j$ and average width $\hat{l}_j$ (in parentheses) of the two-sided 90% CI for the first 10 variables in Simulation Setting 8, using the five approaches: MCMC via BLASSO, two-step random-weighting approach using weighting schemes (2.4) (denoted RW1), (2.5) (denoted RW2) and (2.6) (denoted RW3), and LASSO residual bootstrap (denoted RB). . . . .	31
2.3	Variables in Boston Housing Data Set . . . . .	36
5.1	Average (across $T$ simulated data sets) computational times for various methods in our simulations. The proportion of average computational time (as a percentage of that of MCMC) for each method in each simulation setting is presented in parenthesis. Unit ‘s’ stands for seconds. . . . .	150
5.2	Computational times for various methods in our benchmark and motivating data examples. The proportion of computational time (as a percentage of that of MCMC) for each method in each data set is presented in parenthesis. Units ‘s’, ‘h’ and ‘d’ represent seconds, hours and days respectively. . . . .	151

# List of Figures

- 2.1 Simulation Part I: Sampling distribution of total variation distance between the random-weighting distribution and a Bayesian posterior (averaged across all  $\beta$ 's) among  $T = 500$  simulated data sets in 8 simulation settings between ecdf of MCMC samples and ecdf of samples from each of the 4 methods: two-step random-weighting approach using weighting schemes (2.4) (denoted RW1), (2.5) (denoted RW2) and (2.6) (denoted RW3), and LASSO residual bootstrap (denoted RB). . . . . 27
- 2.2 Simulation Part I: Sampling distribution of conditional (on data) probabilities of selecting  $\beta_1$  and  $\beta_7$  among  $T = 500$  simulated data sets in 8 simulation settings by the 5 methods: MCMC via Bayesian LASSO, two-step random-weighting approach using weighting schemes (2.4) (denoted RW1), (2.5) (denoted RW2) and (2.6) (denoted RW3), and LASSO residual bootstrap (denoted RB). . . . . 29
- 2.3 Simulation Part II: Sampling distribution of total variation distance between random-weighting distribution and a Bayesian posterior (averaged across all  $\beta$ 's) among  $T = 500$  simulated data sets in Simulation Setting 2 between ecdf of MCMC samples and ecdf of the two-step random-weighting samples, computed with  $\lambda_n$  obtained via 1-step cross validation or 2-step cross validation, using weighting schemes (2.4) (2.5) and (2.6) (denoted RW1, RW2 and RW3 respectively). . . . . 34

2.4	Simulation Part II: Sampling distribution of conditional (on data) probabilities of selecting $\beta$ 's among $T = 500$ simulated data sets in Simulation Setting 2 by the two-step random-weighting approach, computed with $\lambda_n$ obtained via 1-step cross validation or 2-step cross validation, using weighting schemes (2.4) (2.5) and (2.6) (denoted RW1, RW2 and RW3 respectively) . . . . .	35
2.5	Boston Housing data example: Marginal posterior/conditional distribution plots for $\beta = (\beta_1, \dots, \beta_{13})'$ sampled from the 5 methods – MCMC via Bayesian LASSO, the two-step random-weighting approach using weighting schemes (2.4) (2.5) and (2.6) (denoted RW1, RW2 and RW3 respectively), as well as the parametric residual bootstrap (denoted RB). . . . .	42
4.1	K-means clustering for data points which are uniformly distributed on an equilateral triangle with vertices $\{(0, 0), (1, 0), (0.5, \sqrt{3}/2)\}$ . The black dots represent the centroids obtained by the K-means algorithm specified with $K = 3$ and the data points are colored according to their respective clusters. . . . .	83
4.2	Cluster partitions obtained by Dahl (2009)'s algorithm, DP-rich and DP-means approaches for the 1-dimensional Galaxy data set (Roeder, 1990), where red color indicates that the pair of observations is clustered together and navy-blue color otherwise. The observations in the data set are arranged in ascending order. . . . .	91
4.3	Schematic depicting different variations of the random-weighting models. . . . .	97
4.4	Sampling distribution for 4 comparison measurements among $T = 10$ simulated data sets in 3 simulation settings: $TV_{\phi(\cdot)}^{(t)}$ (Criterion (1)), $TV_{\hat{\eta}(\cdot)}^{(t)}$ (Criterion (4)), $\tilde{p}_{(\cdot)}^{(t)}$ (Criterion (3)), and $\tilde{g}_{(\cdot)}^{(t)}$ (Criterion (2)). . . . .	106



- 4.5 The ecdf curves of CoV of cluster sizes (see, Equation (4.22)) and the ecdf curves of NMI (see, Equation (4.25)) comparing randomly-sampled pairs of cluster assignments for all 6 methods – MCMC (solid black), VI (solid green), RW DP-means (dashed light-blue), RW DP-rich (solid dark-blue), RW SDP-means (dashed orange) and RW SDP-rich (solid red), as well as the barplots depicting mean absolute differences (in comparison with MCMC) of pairwise probabilities of clustering any two observations together (see, Equation (4.24)) for the other 5 methods, among the 3 benchmark and motivating data examples. . . . . 110
- 5.1 Comparing performances of RW DP-rich and RW SDP-rich using different  $rgr$  tuning parameters. For RW DP-rich, we specify  $\lambda_2^{\text{rwDP-rich}}$  to be 0 (denoted `rwDPmeans`), 0.5 (denoted `rwDPrich1`), 1 (denoted `rwDPrich2`) and 2 (denoted `rwDPrich3`). For RW SDP-rich, we specify  $\lambda_2^{\text{rwSDP-rich}}$  to be 0 (denoted `rwSDPmeans`), 0.5 (denoted `rwSDPrich1`) and 1 (denoted `rwSDPrich2`). . . . . 134
- 5.2 Sampling distribution of average NMI  $\eta_{(\cdot)}^{(b,t)}$  in comparison with ground-truth cluster assignments (Equation (5.19)) among  $T = 10$  simulated data sets in 3 simulation settings for each of the 6 methods: MCMC, VI and the 4 random-weighting setups. . . . . 147
- 5.3 Sampling distribution of average (over  $B$  random-weighting draws) computational times for RW SDP-means and RW SDP-rich across  $T = 10$  simulated data sets in the 3 simulation settings. . . . . 150
- 5.4 Trace plots for posterior number of clusters obtained by MCMC for all benchmark and motivating data sets. . . . . 152
- 5.5 PCA scree plots for wine data set depicting percentage of variance explained in the data across the principal components. The blue dashed lines represent linear mapping of the original data set to a 5-dimensional subspace. The gray dashed lines only serve as interpolation between the points to ease visual inspection. . . . . 153

# Abstract

We consider a general-purpose approximation approach to Bayesian inference in which repeated optimization of a randomized objective function provides surrogate samples from the joint posterior distribution. Our motivation stems from the need for computationally efficient uncertainty quantification in contemporary settings.

This thesis consists of two main parts: In the first part, we establish statistical properties of random-weighting methods in LASSO regression under different regularization parameters and suitable regularity conditions. In Chapter 2, we show that existing approaches have conditional model selection consistency and conditional asymptotic normality at different growth rates of regularization parameters as sample size increases. We propose an extension to the available random-weighting methods and establish that the resulting samples attain conditional sparse normality and conditional consistency in a growing-dimension setting. We illustrate the proposed methodology using synthetic and benchmark data sets, and we discuss the relationship of the results to approximate Bayesian analysis and to perturbation bootstrap methods. Relevant technical details for Chapter 2 are collected in Chapter 3.

The second part of the thesis concerns with random-weighting discrete mixture models under the Bayesian nonparametric learning (NPL) framework. Specifically, in Chapter 4, we first develop new asymptotics for a Dirichlet Process Mixture (DPM) model – the DP-rich algorithm. Unlike the DP-means approach that arises as small-variance-asymptotics of the DPM, our DP-rich setup retains the rich-gets-richer property of the DPM. We then apply the random-weighting mechanism under the Bayesian NPL framework on an extended version of the DP-rich setup that leads to our main random-weighting discrete mixture model: the

random-weighting scaled DP-rich (RW SDP-rich) approach. We develop a scalable algorithm (which is trivially parallelizable over multiple computing nodes) that ensures local convergence of solutions, and explore various related random-weighting mixture models via simplifications of our RW SDP-rich setup. We illustrate, via various simulations and benchmark data examples, that our RW SDP-rich approach provides reasonable approximation to MCMC posterior clustering for the DPM model. Finally, we establish several appealing theoretical properties of our random-weighting models under the Bayesian NPL framework. Additional details for our random-weighting mixture models are collected as supplementary material in Chapter 5.

# Chapter 1

## Introduction

Computational and modeling considerations in contemporary Bayesian analysis have led to renewed interest in a class of weighted bootstrap algorithms for posterior inference. An important example in this class is the weighted likelihood bootstrap (WLB), which was designed to yield approximate posterior samples in parametric models (M. A. Newton & Raftery, 1994). Compared to Markov chain Monte Carlo (MCMC), for example, WLB provides computationally efficient approximate posterior samples in cases where likelihood optimization is relatively easy. Asymptotic arguments demonstrate that for sufficiently regular models WLB samples provide a valid posterior approximation as the amount of data increases. However, the utility of WLB approximation is in doubt as we consider the availability of sophisticated MCMC schemes and codes that are simulation consistent: i.e., they produce exact posterior summaries on a fixed data set as the amount of computing resources increases without bound (e.g., Carpenter et al., 2017). Contemporary models also strain the validity of regularity conditions that support existing WLB asymptotic analysis.

The decades since publication of WLB have seen dramatic improvements in algorithms and code systems for optimization, as well as the interpenetration of these techniques into statistics (e.g., Bhadra, Datta, Polson, & Willard, 2019; Duchi, Jordan, Wainwright, & Wibisono, 2015; R. J. Tibshirani & Taylor, 2011). This period has likewise seen improvements in Bayesian analysis, but there continue to be difficulties with posterior computation in

some settings, especially given the problem to assure Monte Carlo error bounds with MCMC (e.g., Mossel and Vigoda (2006)), the increased size of data sets (e.g., Welling & Teh, 2011), the increased complexity of modeling techniques (e.g., Jordan, 2013), and the growing emphasis on tools that are not overly sensitive to modeling assumptions. The search for scalable, accurate posterior inference tools continues to be an important challenge in computational statistics.

Framing WLB in a contemporary context, M. Newton, Polson, and Xu (2021) extended the posterior approximation scheme to a class of penalized likelihood objective functions. They saw good performance of the proposed Weighted Bayesian Bootstrap (WBB) extension in high-dimensional regression, trend filtering, and deep-learning applications. Others have recognized the utility of weighted bootstrap computations beyond the realm of parametric posterior approximation. A critical perspective was provided by Bissiri, Holmes, and Walker (2016) with the concept of generalized Bayesian inference. Rather than constructing a fully specified probabilistic model for data, as in traditional Bayesian analysis, the authors told us to focus on an objective function for a parameter of interest, sidestepping the marginal posterior inference on this parameter by creating a generalized Bayesian posterior defined directly using this objective function. Lyddon, Holmes, and Walker (2019) discovered a key connection between the generalized Bayesian posterior and WLB sampling, and constructed a modification called the loss-likelihood bootstrap to leverage this connection. Further links to nonparametric Bayesian inference were recently reported in Lyddon, Walker, and Holmes (2018) and Fong, Lyddon, and Holmes (2019), who introduced the concept of Bayesian Nonparametric Learning (Bayesian NPL). These works demonstrate renewed interest in the operating characteristics of weighted bootstrap computation.

Whether we aim for approximate parametric Bayes, generalized Bayes, or model-guided nonparametric Bayes, it is important to understand the distributional properties of these *random-weighting* procedures. Precise answers are difficult, even with simple loss functions (e.g., Hjort & Ongaro, 2005), and so asymptotic methods are helpful to study the conditional distribution of parameters of interest given data. Adopting a Dirichlet prior on the sampling

distribution, Fong et al. (2019) pointed out that WBB sampling is consistent under suitable regularity conditions, due to posterior consistency property of the Dirichlet process (e.g., Ghosal, Ghosh, & Ramamoorthi, 1999; Ghosal, Ghosh, & van der Vaart, 2000). M. A. Newton and Raftery (1994)'s first-order analysis of the weighted bootstrap samples yields the same Gaussian limits as the standard Bernstein-von-Mises results (e.g., van der Vaart, 1998) under a correctly-specified Bayesian parametric model. Under model misspecification setting, Lyddon et al. (2019) showed that the Gaussian limits of weighted bootstrap sampling do not coincide with their Bayesian counterparts in Kleijn and van der Vaart (2012). Instead, they mimic the Gaussian limits in Huber (1967) – the asymptotic covariance matrix of the weighted bootstrap sampling is in fact the well-known sandwich covariance matrix in robust statistics literature.

With the work reported here, we first aim to extend asymptotic analysis for weighted bootstrap distributions to high-dimensional regression models. The first part of our work, namely Chapters 2 and 3, adapts frequentist-theory asymptotic arguments, notably the works of Knight and Fu (2000), Zhao and Yu (2006) and Liu and Yu (2013), to the present context.

Subsequently, in Chapters 4 and 5, we further explore random weighting in discrete mixture models. Quantifying the uncertainty in clustering is a difficult but important inference problem that arises in many statistical applications. It could be an end in itself (e.g., Wade & Ghahramani, 2018, and references therein), or it might be relevant when clustering is one element in sequence of data-analysis steps (e.g., Ma, Korthauer, Kendzioriski, & Newton, 2021). An important general approach to address the clustering inference problem is to invert through some computational means (e.g., Markov chain Monte Carlo, or variational Bayes) a fully specified prior and generative statistical model in order to access the posterior distribution of the clustering object (e.g. Müller, Quintana, Jara, & Hanson, 2015; Scrucca, Fop, Murphy, & Raftery, 2016). An alternative approach could use models to guide optimization-based computations, such as in generalized Bayesian inference (Bissiri et al., 2016) or Bayesian nonparametric learning (NPL) (Lyddon et al., 2019, 2018). This

approach has potential benefits that are both computational (e.g., leveraging optimization tools; no MCMC diagnosis) as well as statistical (e.g., less reliance on model assumptions). Discrete mixing has long provided a model-based approach to clustering (e.g., McLachlan, Lee, & Rathnayake, 2019), and by using such models to guide Bayesian NPL, we find new and potentially useful schemes for clustering inference.

One popular class of Bayesian nonparametric discrete mixture models is the Dirichlet Process Mixture (DPM) models, due to its appealing theoretical properties (e.g., strong consistency, exchangeability) and practicality (e.g., DPM readily models uncertainty about the number clusters without the need for additional model selection procedures). Whilst various standard MCMC procedures have been developed for implementing the DPM models (e.g., Müller et al., 2015), computational challenges remain in terms of poor scalability and difficulty in assessing chain mixing. While many approximate Bayesian procedures are available for finite-mixture-models (e.g., Nemeth & Fearnhead, 2021, and references therein), to the best of our knowledge, Blei and Jordan (2006)'s Variational Inference (VI) approach remains the preferred approximate posterior inference tool for the DPM to date, albeit its own limitations such as underestimation of posterior uncertainty (Fong et al., 2019). Meanwhile, other authors were concerned with posterior point estimation (e.g., Karabatsos, 2020; Zuanetti, Muller, Zhu, Yang, & Ji, 2019) instead of uncertainty quantification. Consequently, we want to further explore approximate posterior inference for countable discrete mixture models using the Bayesian NPL approach in Chapters 4 and 5.

## Chapter 2

# Random Weighting in LASSO Regression

### 2.1. Preamble

Consider the well-studied linear regression model with fixed design

$$\mathbf{Y} = \beta_\mu \mathbf{1}_n + X\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad (2.1)$$

where  $\mathbf{Y} = (y_1, \dots, y_n)' \in \mathbb{R}^n$  is the response vector,  $\mathbf{1}_n$  is a  $n \times 1$  vector of ones,  $X \in \mathbb{R}^{n \times p_n}$  is the design matrix,  $\boldsymbol{\beta}$  is the vector of regression coefficients, and  $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_n)'$  is the vector of independent and identically distributed (i.i.d.) random errors with mean 0 and variance  $\sigma_\epsilon^2$ . Without loss of generality, we assume that the columns of  $X$  are centered, and take  $\hat{\beta}_\mu = \bar{Y}$ , in which case we can replace  $\mathbf{Y}$  in (2.1) with  $\mathbf{Y} - \bar{Y}\mathbf{1}_n$ , and concentrate on inference for  $\boldsymbol{\beta}$ . Again, without loss of generality, we also assume  $\bar{Y} = 0$ . Let  $\boldsymbol{\beta}_0 \in \mathbb{R}^{p_n}$  be the true model coefficients with  $q$  non-zero components, where  $q \leq \min(p_n, n)$ . Note that  $\mathbf{Y}$ ,  $X$  and  $\boldsymbol{\epsilon}$  are all indexed by sample size  $n$ , but we omit the subscript whenever this does not cause confusion.



Recall, the LASSO estimator is given by

$$\widehat{\beta}_n^{\text{LAS}} := \arg \min_{\beta} \sum_{i=1}^n (y_i - \mathbf{x}'_i \beta)^2 + \lambda_n \sum_{j=1}^{p_n} |\beta_j|, \quad (2.2)$$

for a scalar penalty  $\lambda_n$  (R. Tibshirani, 1996), where  $\mathbf{x}'_i$  is the  $i^{\text{th}}$  row of  $X$ . The LASSO is a canonical example in the broad class of penalized inference procedures; for the purpose of uncertainty quantification in such models, M. Newton et al. (2021) developed the random-weighting approach as a straightforward technique to leverage advances in optimization. They reported good performance in high-dimensional regression, trend-filtering and deep learning applications. In particular, their random-weighting version of (2.2) is

$$\widehat{\beta}_n^w := \arg \min_{\beta} \left\{ \sum_{i=1}^n W_i (y_i - \mathbf{x}'_i \beta)^2 + \lambda_n \sum_{j=1}^{p_n} W_{0,j} |\beta_j| \right\}, \quad (2.3)$$

where the analyst first chooses a distribution  $F_W$  with  $P(W > 0) = 1$  and  $\mathbb{E}(W^4) < \infty$ , and constructs  $W_i \stackrel{iid}{\sim} F_W$  for all  $i = 1, 2, \dots, n$ . The precise treatment of penalty-associated weights  $\mathbf{W}_0 = (W_{0,1}, \dots, W_{0,p_n})$  induces several random-weighting variations, the simplest of which has

$$W_{0,j} = 1 \quad \forall j, \quad (2.4)$$

or the penalty terms all share a common random weight

$$W_{0,j} = W_0 \quad \forall j, \quad \text{where } (W_0, W_i) \stackrel{iid}{\sim} F_W \quad \forall i, \quad (2.5)$$

and the most elaborate of which has all entries

$$(W_{0,j}, W_i) \stackrel{iid}{\sim} F_W \quad \forall i, j. \quad (2.6)$$

Regardless of our treatment of the weights, (2.3) yields independent and identically

distributed draws from the conditional distribution of  $\widehat{\beta}_n^w$  given data when we repeatedly realize weight vectors *in silico* by one of the random-weighting mechanisms. A computational benefit for uncertainty quantification is that random weighting is readily parallelized. Though useful inference tools already exist for LASSO regression (e.g., Friedman, Hastie, & Tibshirani, 2010), we focus on this well-studied model in order to extend random-weighting theory and also to guide work for more complex settings where random weighting may be readily applied (M. Newton et al., 2021). In the present study we investigate the asymptotic properties of (2.3), with attention on properties of the conditional distribution given data. By allowing different rates of growth of the regularization parameter  $\lambda_n$ , and under suitable regularity conditions, we prove that the random-weighting method has the following properties:

- conditional model selection consistency (for both growing  $p_n$  and fixed  $p$ )
- conditional consistency (for fixed  $p_n = p$ )
- conditional asymptotic normality (for fixed  $p_n = p$ )

for all three weighting schemes (2.4), (2.5) and (2.6). We find there is no common  $\lambda_n$  that would allow random-weighting samples to have conditional sparse normality (i.e., simultaneously to enjoy conditional model selection consistency and to achieve conditional asymptotic normality on the true support of  $\beta$ ) even under fixed  $p_n = p$  setting. Consequently, we propose an extension to the random-weighting framework (2.3) by adopting a two-step procedure in the optimization step as laid out in Algorithm 2. We prove that a common regularization rate  $\lambda_n$  allows two-step random-weighting samples to achieve conditional sparse normality and conditional consistency properties under growing  $p_n$  setting.

After setting regularity conditions and notation in Section 2.2, we report our main distributional results for random weighting in Section 2.3. Asymptotic techniques from Knight and Fu (2000), Zhao and Yu (2006) and Liu and Yu (2013) guide our calculations. Extensive simulations and application to a benchmark data set illustrate how two-step

random weighting under schemes (2.4), (2.5) and (2.6) compares with both Bayesian and bootstrap methods for uncertainty quantification (Section 2.4). In Section 2.5 we comment on our findings in relation to the perturbation bootstrap (e.g., Das & Lahiri, 2019) and also to recent nonparametric Bayesian work that has renewed interest in the operating characteristics of random-weighting (Fong et al., 2019; Lyddon et al., 2019, 2018). Detailed proofs are presented in Chapter 3.

## 2.2. Problem Setup

We assume throughout that the unknown number of truly relevant predictors,  $q$ , is fixed, that

$$\mathbb{E}(\epsilon_i^4) < \infty \quad \forall i, \quad (2.7)$$

and all  $p_n$  predictors are bounded, i.e.  $\exists M_1 > 0$  such that

$$|x_{ij}| \leq M_1 \quad \forall i = 1, \dots, n; j = 1, \dots, p_n, \quad (2.8)$$

where  $x_{ij}$  refers to the  $(i, j)^{th}$  element of  $X$ .

Without loss of generality, we partition  $\beta_0$  into

$$\beta_0 = \begin{bmatrix} \beta_{0(1)} \\ \beta_{0(2)} \end{bmatrix},$$

where  $\beta_{0(1)}$  refers to the  $q \times 1$  vector of non-zero true regression parameters, and  $\beta_{0(2)}$  is a  $(p_n - q) \times 1$  zero vector. Similarly, we partition the columns of the design matrix  $X$  into

$$X = \begin{bmatrix} X_{(1)} & X_{(2)} \end{bmatrix}$$

which corresponds to  $\beta_{0(1)}$  and  $\beta_{0(2)}$  respectively.

We consider both fixed-dimensional ( $p_n = p$ ) and growing-dimensional ( $p_n$  increases with  $n$ ) settings. In the growing dimensional setting, we assume that for some  $M_2 > 0$ ,

$$\boldsymbol{\alpha}' \left[ \frac{X'_{(1)} X_{(1)}}{n} \right] \boldsymbol{\alpha} \geq M_2 \quad \forall \quad \|\boldsymbol{\alpha}\|_2 = 1. \quad (2.9)$$

Note that assumptions (2.8) and (2.9), coupled with the fact that  $q$  is fixed, ensure that  $\frac{1}{n} X'_{(1)} X_{(1)}$  is invertible  $\forall n$ , a fact that we rely on in this paper.

Meanwhile, for fixed-dimensional ( $p_n = p$ ) setting, we assume that  $\text{rank}(X) = p$  and there exists a non-singular matrix  $C$  such that

$$\frac{1}{n} X' X = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}'_i \rightarrow C \quad \text{as } n \rightarrow \infty, \quad (2.10)$$

where  $\mathbf{x}_i$  is the  $i^{\text{th}}$  row of the design matrix  $X$ .

**Comments on assumptions:** The fixed- $q$  assumption is commonly found in Bayesian linear-model literature, such as Johnson and Rossell (2012), and Narisetty and He (2014). Since we intend to compare the random-weighting approach with posterior inference, we make the fixed- $q$  assumption to align with existing Bayesian theory. The finite-moment assumption (2.7) of  $\epsilon$  is commonly found in literature (e.g., Camponovo, 2015; Das & Lahiri, 2019) is weaker than the normality assumption commonly specified under a Bayesian approach (e.g., Johnson & Rossell, 2012; Narisetty & He, 2014; Park & Casella, 2008). Assumption (2.8) can also be found in some seminal papers, such as Zhao and Yu (2006) and A. Chatterjee and Lahiri (2011b), and in fact, can be (trivially) achieved by standardizing the covariates. Assumption (2.9) is equivalent to providing a lower bound to the minimum eigenvalue of  $\frac{1}{n} X'_{(1)} X_{(1)}$ . This eigenvalue assumption is very common in both frequentist and Bayesian literature, such as Zhao and Yu (2006) and Narisetty and He (2014). Finally, assumption (2.10) is common in the LASSO literature under fixed  $p$  setting, which can be traced back to Knight and Fu (2000) and Zhao and Yu (2006). This assumption basically explains the relationship between the predictors under a fixed design model, and can be interpreted as

the direct counterpart to the variance-covariance matrix of  $X$  under a random design model. For the case of growing  $p_n$ , assumption (2.10) is no longer appropriate since the dimension of  $\frac{1}{n}X'X$  grows.

**Probability Space:** There are two sources of variation in the random-weighting setup (2.3), namely the error terms  $\epsilon$  and the user-defined weights  $\mathbf{W}$ . In this paper, we consider a common probability space with the common probability measure  $P = P_D \times P_W$ , where  $P_D$  is the probability measure of the observed data  $Y_1, Y_2, \dots$ , and  $P_W$  is the probability measure of the triangular array of random weights (e.g., Mason & Newton, 1992). The use of product measure reflects the independence of user-defined  $\mathbf{W}$  and data-associated  $\epsilon$ . We focus on the conditional probabilities given data, that is, given the sigma-field  $\mathcal{F}_n$  generated by  $\epsilon$ :

$$\mathcal{F}_n := \sigma(Y_1, \dots, Y_n) = \sigma(\epsilon_1, \dots, \epsilon_n).$$

The study of convergence of these conditional probabilities  $P(\cdot | \mathcal{F}_n)$  under a weighted bootstrap framework is not new; see, for example, Mason and Newton (1992) and Lyddon et al. (2019). We now outline some definitions and notations in this respect.

**Conditional Convergence Notations:** Let random variables (or vectors)  $U, V_1, V_2, \dots$  be defined on  $(\Omega, \mathcal{A})$ . We say  $V_n$  converges in conditional probability *a.s.*  $P_D$  to  $U$  if for every  $\delta > 0$ ,

$$P(\|V_n - U\| > \delta | \mathcal{F}_n) \rightarrow 0 \quad a.s. P_D$$

as  $n \rightarrow \infty$ . The notation *a.s.*  $P_D$  is read as *almost surely under  $P_D$* , and means *for almost every infinite sequence of data  $Y_1, Y_2, \dots$* . For brevity, this convergence is denoted

$$V_n \xrightarrow{\text{c.p.}} U \quad a.s. P_D.$$

Similarly, we say  $V_n$  converges in conditional distribution *a.s.*  $P_D$  to  $U$  if for any Borel set  $A \subset \mathbb{R}$ ,

$$P(V_n \in A | \mathcal{F}_n) \rightarrow P(U \in A) \quad a.s. P_D$$

as  $n \rightarrow \infty$ . For brevity, this convergence is denoted

$$V_n \xrightarrow{\text{c.d.}} U \quad a.s. P_D.$$

In addition, for random variables (or vectors)  $V_1, V_2, \dots$  and random variables  $U_1, U_2, \dots$ , we say

$$V_n = O_p(U_n) \quad a.s. P_D$$

if and only if, for any  $\delta > 0$ , there is a constant  $C_\delta > 0$  such that *a.s.*  $P_D$ ,

$$\sup_n P \left( \|V_n\| \geq C_\delta |U_n| \mid \mathcal{F}_n \right) < \delta;$$

whereas

$$V_n = o_p(U_n) \quad a.s. P_D$$

if and only if

$$\frac{V_n}{U_n} \xrightarrow{\text{c.P.}} 0 \quad a.s. P_D.$$

**Other Notation:** Following the usual convention, denote  $\Phi\{\cdot\}$  as the cumulative distribution function of the standard normal distribution. For two random variables  $U$  and  $V$ , the expression  $U \perp V$  is read as “ $U$  is independent of  $V$ ”. Denote  $\|\cdot\|_2$  and  $\|\cdot\|_F$  as the  $l_2$  norm and Frobenius norm respectively. Let  $\mathbf{1}_k$  and  $I_k$  be  $k \times 1$  vector of ones and  $k \times k$  identity matrix respectively for some integer  $k \geq 2$ . Besides that, for any two vectors  $\mathbf{u}$  and  $\mathbf{v}$  of the same dimension, we denote  $\mathbf{u} \circ \mathbf{v}$  as the Hadamard (entry-wise) product of the two vectors. In addition, define

$$\begin{bmatrix} C_{n(11)} & C_{n(12)} \\ C_{n(21)} & C_{n(22)} \end{bmatrix} := \frac{1}{n} X'X = \frac{1}{n} \begin{bmatrix} X'_{(1)}X_{(1)} & X'_{(1)}X_{(2)} \\ X'_{(2)}X_{(1)} & X'_{(2)}X_{(2)} \end{bmatrix}.$$

Notice that an immediate consequence of Assumption (2.10) is that

$$C_{n(ij)} \rightarrow C_{ij} \quad \forall i, j = 1, 2,$$

where  $C_{11}$  is invertible. Furthermore, denote  $\mu_W$  and  $\sigma_W^2$  as the mean and variance of the random weight distribution  $F_W$ . Let  $D_n = \text{diag}(W_1, \dots, W_n)$ , and define

$$\begin{bmatrix} C_{n(11)}^w & C_{n(12)}^w \\ C_{n(21)}^w & C_{n(22)}^w \end{bmatrix} := \frac{1}{n} X' D_n X = \frac{1}{n} \begin{bmatrix} X'_{(1)} D_n X_{(1)} & X'_{(1)} D_n X_{(2)} \\ X'_{(2)} D_n X_{(1)} & X'_{(2)} D_n X_{(2)} \end{bmatrix}.$$

Notice that  $D_n$  does not contain any penalty weights  $W_{0,j}$ . For weighting scheme (2.6), the penalty weights  $\mathbf{W}_0 = (W_{0,1}, \dots, W_{0,p_n})$  could also be partitioned into

$$\mathbf{W}_0 = \begin{bmatrix} \mathbf{W}_{0(1)} \\ \mathbf{W}_{0(2)} \end{bmatrix},$$

which corresponds to the partition of  $\beta_0$ . For ease of notation, define

$$\begin{aligned} \mathbf{Z}_{n(1)}^w &= \frac{1}{\sqrt{n}} X'_{(1)} D_n \epsilon, \\ \mathbf{Z}_{n(2)}^w &= \frac{1}{\sqrt{n}} X'_{(2)} D_n \epsilon, \\ \mathbf{Z}_{n(3)}^w &= C_{n(21)} C_{n(11)}^{-1} \mathbf{Z}_{n(1)}^w - \mathbf{Z}_{n(2)}^w, \\ \tilde{C}_n^w &= C_{n(21)}^w \left( C_{n(11)}^w \right)^{-1} - C_{n(21)} C_{n(11)}^{-1}. \end{aligned}$$

Finally, the function  $\text{sgn}(\cdot)$  maps positive entry to 1, negative entry to -1 and zero to zero.

An estimator  $\hat{\beta}$  is said to be equal in sign to the true parameter  $\beta_0$ , if

$$\text{sgn}(\hat{\beta}) = \text{sgn}(\beta_0),$$

and is denoted as

$$\hat{\beta} \stackrel{s}{=} \beta_0.$$

## 2.3. Main Results

### 2.3.1. One-step Procedure

We investigate the asymptotic properties of random-weighting draws (2.3) obtained from Algorithm 1, which coincides with the weighted Bayesian bootstrap method considered by M. Newton et al. (2021). For convenience, we shall call this the “one-step procedure” to distinguish it from the extended framework that we shall discuss in Section 2.3.2.

---

#### Algorithm 1 Random-Weighting in LASSO regression

---

**Require:** data  $D = (\mathbf{y}, X)$ , regularization parameter  $\lambda_n$ , number of draws  $B$ , choice of random weight distribution  $F_W$ , choice of weighting schemes (2.4), (2.5) or (2.6)

1: **for**  $b = 1$  to  $B$  **do**

2:     Draw i.i.d. random weights from  $F_W$  and substitute them into (2.3).

3:     Store  $\hat{\beta}_n^{w,b}$  obtained by optimizing (2.3).

4: **end for**

**Ensure:**  $B$  parameter samples  $\{\hat{\beta}_n^{w,b}\}_{b=1}^B$

---

First, we establish the property of conditional model selection given data. In particular, we are interested in the conditional probability of the random-weighting samples matching the signs of  $\beta_0$ . Notably, sign consistency is stronger than variable selection consistency, which requires only matching of zeros. Nevertheless, we agree with Zhao and Yu (2006)’s argument of considering sign consistency – it allows us to avoid situations where models have matching zeroes but reversed signs, which hardly qualify as correct models. We begin with a result that establishes the lower bound for this conditional probability.

**Proposition 2.1.** *Suppose  $p_n \leq n$  and  $\text{rank}(X) = p_n$ . Assume (2.7), (2.8) and (2.9). Furthermore, assume the **strong irrepresentable condition** (Zhao & Yu, 2006): there exists a positive constant vector  $\boldsymbol{\eta}$  such that*

$$\left| C_{n(21)} (C_{n(11)})^{-1} \text{sgn}(\beta_{0(1)}) \right| \leq \mathbf{1}_{p_n-q} - \boldsymbol{\eta}, \quad (2.11)$$



where  $0 < \eta_j \leq 1 \forall j = 1, \dots, p_n - q$ , and the inequality holds element-wise. Then, for all  $n \geq p_n$ ,

$$P\left(\widehat{\beta}_n^w(\lambda_n) \stackrel{s}{=} \beta_0 | \mathcal{F}_n\right) \geq P\left(A_n^w \cap B_n^w | \mathcal{F}_n\right),$$

where

(a) for weighting scheme (2.4),

$$\begin{aligned} A_n^w &\equiv \left\{ \left| \left( C_{n(11)}^w \right)^{-1} \left( \mathbf{Z}_{n(1)}^w - \frac{\lambda_n}{2\sqrt{n}} \text{sgn} [\beta_{0(1)}] \right) \right| \leq \sqrt{n} |\beta_{0(1)}| \text{ element-wise} \right\} \\ B_n^w &\equiv \left\{ \left| \widetilde{C}_n^w \left( \mathbf{Z}_{n(1)}^w - \frac{\lambda_n}{2\sqrt{n}} \text{sgn} [\beta_{0(1)}] \right) + \mathbf{Z}_{n(3)}^w \right| \leq \frac{\lambda_n}{2\sqrt{n}} \boldsymbol{\eta} \text{ element-wise} \right\}; \end{aligned}$$

(b) for weighting scheme (2.5),

$$\begin{aligned} A_n^w &\equiv \left\{ \left| \left( C_{n(11)}^w \right)^{-1} \left( \mathbf{Z}_{n(1)}^w - \frac{\lambda_n W_0}{2\sqrt{n}} \text{sgn} [\beta_{0(1)}] \right) \right| \leq \sqrt{n} |\beta_{0(1)}| \text{ element-wise} \right\} \\ B_n^w &\equiv \left\{ \left| \widetilde{C}_n^w \left( \mathbf{Z}_{n(1)}^w - \frac{\lambda_n W_0}{2\sqrt{n}} \text{sgn} [\beta_{0(1)}] \right) + \mathbf{Z}_{n(3)}^w \right| \leq \frac{\lambda_n W_0}{2\sqrt{n}} \boldsymbol{\eta} \text{ element-wise} \right\}; \end{aligned}$$

(c) for weighting scheme (2.6),

$$\begin{aligned} A_n^w &\equiv \left\{ \left| \left( C_{n(11)}^w \right)^{-1} \left( \mathbf{Z}_{n(1)}^w - \frac{\lambda_n}{2\sqrt{n}} \mathbf{W}_{0(1)} \circ \text{sgn} [\beta_{0(1)}] \right) \right| \right. \\ &\quad \left. \leq \sqrt{n} |\beta_{0(1)}| \text{ element-wise} \right\} \\ B_n^w &\equiv \left\{ \left| \widetilde{C}_n^w \left( \mathbf{Z}_{n(1)}^w - \frac{\lambda_n}{2\sqrt{n}} \mathbf{W}_{0(1)} \circ \text{sgn} [\beta_{0(1)}] \right) + \mathbf{Z}_{n(3)}^w \right| \right. \\ &\quad \left. \leq \frac{\lambda_n}{2\sqrt{n}} \left( \mathbf{W}_{0(2)} - \left| C_{n(21)} (C_{n(11)})^{-1} \mathbf{W}_{0(1)} \circ \text{sgn} [\beta_{0(1)}] \right| \right) \text{ element-wise} \right\}. \end{aligned}$$

The  $\text{rank}(X) = p_n \leq n$  assumption in Proposition 2.1 ensures that the random-weighting setup (2.3) has a unique solution (Osborne, Presnell, & Turlach, 2000). For a random-design setting, the  $\text{rank}(X) = p_n \leq n$  assumption can be replaced with the assumption that  $X$  is drawn from a joint continuous distribution (R. J. Tibshirani, 2013).

The strong irrepresentable condition (2.11) can be seen as a constraint on the relationship between active covariates and inactive covariates, that is, the total amount of an irrelevant covariate “represented” by a relevant covariate must be strictly less than one. Similar to Zhao and Yu (2006)’s argument,  $A_n^w$  refers to recovery of the signs of coefficients for  $\beta_{0(1)}$ , and  $B_n^w$  further implies obtaining  $\hat{\beta}_{n(2)}^w = \mathbf{0}$  given  $A_n^w$ . The regularization parameter  $\lambda_n$  continues to play the role of trade-off between  $A_n^w$  and  $B_n^w$ : higher  $\lambda_n$  leads to larger  $B_n^w$  but smaller  $A_n^w$ , which forces the random-weighting method to drop more covariates, and vice versa. Meanwhile, larger  $\eta$  in (2.11), which could be interpreted as lower “correlation” between active covariates and inactive covariates, increases  $B_n^w$  but does not affect  $A_n^w$ , thus allowing the random-weighting method to better select the true model. Zhao and Yu (2006) also gave a few sufficient conditions that ensure the following designs of  $X$  satisfy condition (2.11):

- constant positive correlation,
- bounded correlation,
- power-decay correlation,
- orthogonal design, and
- block-wise design.

Again, we would like to highlight the fact that conditional on  $\mathcal{F}_n$ , the randomness of  $A_n^w$  and  $B_n^w$  derives from the random weights instead of  $\epsilon$ . Besides that, notice how the presence of different penalty weights in weighting scheme (2.6) affects the strong irrepresentable condition (2.11) in  $B_n^w$ . We will see how these different weighting schemes affect the constraints on  $p_n$  and  $\lambda_n$  in order to achieve conditional model selection consistency.

**Theorem 2.2.** (*Conditional Model Selection Consistency*) *Assume assumptions in Proposition 2.1.*

- (a) Under weighting schemes (2.4) and (2.5), if there exists  $\frac{1}{2} < c_1 < c_2 < 1.5 - c_1$  and  $0 \leq c_3 < \min\{2(c_2 - c_1), 2c_1 - 1\}$  for which  $\lambda_n = \mathcal{O}(n^{c_2})$  and  $p_n = \mathcal{O}(n^{c_3})$ , then as  $n \rightarrow \infty$ ,

$$P\left(\widehat{\beta}_n^w(\lambda_n) \stackrel{s}{=} \beta_0 | \mathcal{F}_n\right) \rightarrow 1 \quad a.s. P_D.$$

- (b) Under weighting scheme (2.6), if  $(W_i, W_{0,j}) \stackrel{iid}{\sim} \text{Exp}(\theta_w)$  for some  $\theta_w > 0$ , and if  $\boldsymbol{\eta} = \mathbf{1}_{p_n - q}$ , and if there exists  $\frac{1}{2} < c_1 < c_2 < 1.5 - c_1$  and  $0 \leq c_3 < \min\{\frac{2}{3}(c_2 - c_1), 2c_1 - 1\}$  for which  $\lambda_n = \mathcal{O}(n^{c_2})$  and  $p_n = \mathcal{O}(n^{c_3})$ , then as  $n \rightarrow \infty$ ,

$$P\left(\widehat{\beta}_n^w(\lambda_n) \stackrel{s}{=} \beta_0 | \mathcal{F}_n\right) \rightarrow 1 \quad a.s. P_D.$$

Theorem 2.2 could be interpreted as the ‘‘concentration’’ of the conditional distribution of signs of  $\widehat{\beta}_n^w$  around the neighborhood of the true signs of  $\beta$  as  $n \rightarrow \infty$ . Comparing the three weighting schemes, we can see that assigning random weights on the penalty term further impedes how fast  $p_n$  could increase with  $n$  while achieving conditional model selection consistency, especially when the penalty terms do not share a common random weight in weighting scheme (2.6). This adversely affects/violates the strong irrepresentable assumption (2.11), unless under a stringent condition where  $\boldsymbol{\eta} = \mathbf{1}$ . One sufficient condition for  $\boldsymbol{\eta} = \mathbf{1}$  would be zero correlation between any relevant predictor and any irrelevant predictor, i.e.  $C_{n(21)} = \mathbf{0}$  for all  $n$ .

We also point out that the conditional model selection consistency property under a fixed dimensional ( $p_n = p$ ) setting could be easily obtained by taking  $c_3 = 0$  in Theorem 2.2.

The next two results concern with the properties of conditional consistency and conditional asymptotic normality of the random-weighting samples under a fixed-dimension ( $p_n = p$ ) setting.

**Theorem 2.3.** *Suppose  $p_n = p$  is fixed. Assume (2.7), (2.8) and (2.10).*

- (a) **(Conditional Consistency)** *If  $\frac{\lambda_n}{n} \rightarrow 0$ , then for all three weighting schemes (2.4), (2.5)*

and (2.6),

$$\widehat{\beta}_n^w \xrightarrow{c.p.} \beta_0 \quad a.s. P_D.$$

(b) If  $\frac{\lambda_n}{n} \rightarrow \lambda_0 \in (0, \infty)$ , then

$$\left(\widehat{\beta}_n^w - \beta_0\right) \xrightarrow{c.d.} \arg \min_{\mathbf{u}} g(\mathbf{u}) \quad a.s. P_D,$$

where

$$g(\mathbf{u}) = \mu_W \mathbf{u}' C \mathbf{u} + \lambda_0 \sum_{j=1}^p W_j |\beta_{0,j} + u_j|$$

and

- (i)  $W_j = 1$  for all  $j$  under weighting scheme (2.4),
- (ii)  $W_j = W_0$  for all  $j$  and  $W_0 \sim F_W$  under weighting scheme (2.5),
- (iii)  $W_j \stackrel{iid}{\sim} F_W$  under weighting scheme (2.6).

In other words, the conditional distribution of  $\widehat{\beta}_n^w$  concentrates in the neighborhood of  $\arg \min_{\mathbf{u}} g(\mathbf{u})$  as the sample size increases. In fact, for part (b)(i) of Theorem 2.3, conditional convergence in probability takes place since  $g(\mathbf{u})$  is not a random function (i.e., does not involve any non-degenerate random variables).

**Theorem 2.4. (Asymptotic Conditional Distribution)** Suppose  $p_n = p$  is fixed. Assume (2.7), (2.8) and (2.10). Let  $\widehat{\beta}_n^{\text{SC}}$  be a strongly consistent estimator of  $\beta$  in the linear model (2.1) such that for  $\mathbf{e}_n = \mathbf{Y} - X\widehat{\beta}_n^{\text{SC}}$ ,

$$\frac{1}{\sqrt{n}} X' \mathbf{e}_n \rightarrow \mathbf{0} \quad a.s. P_D. \quad (2.12)$$

If  $q = p$  and  $\frac{\lambda_n}{\sqrt{n}} \rightarrow \lambda_0 \in [0, \infty)$ , then

$$\sqrt{n} \left(\widehat{\beta}_n^w - \widehat{\beta}_n^{\text{SC}}\right) \xrightarrow{c.d.} \arg \min_{\mathbf{u}} V(\mathbf{u}) \quad a.s. P_D,$$

where

$$V(\mathbf{u}) = -2\mathbf{u}'\Psi + \mu_W \mathbf{u}'C\mathbf{u} + \lambda_0 \sum_{j=1}^p W_j [u_j \operatorname{sgn}(\beta_{0,j})],$$

for  $\Psi \sim N(\mathbf{0}, \sigma_W^2 \sigma_\epsilon^2 C)$ , and

- (i)  $W_j = 1$  for all  $j$  under weighting scheme (2.4),
- (ii)  $W_j = W_0$  for all  $j$ ,  $W_0 \sim F_W$  and  $W_0 \perp \Psi$  under weighting scheme (2.5),
- (iii)  $W_j \stackrel{iid}{\sim} F_W$  and  $W_j \perp \Psi$  for all  $j$  under weighting scheme (2.6).

In particular, if  $\lambda_0 = 0$ , then for all three weighting schemes (2.4), (2.5) and (2.6),

$$\sqrt{n} \left( \widehat{\beta}_n^w - \widehat{\beta}_n^{\text{SC}} \right) \xrightarrow{c.d.} N \left( \mathbf{0}, \frac{\sigma_W^2 \sigma_\epsilon^2}{\mu_W^2} C^{-1} \right) \quad a.s. P_D.$$

The OLS estimator  $\widehat{\beta}_n^{\text{OLS}}$  and the standard LASSO estimator  $\widehat{\beta}_n^{\text{LAS}}(\lambda_n^*)$  with  $\lambda_n^* = o(\sqrt{n})$  are two qualified candidates for  $\widehat{\beta}_n^{\text{SC}}$  to satisfy the conditions in Theorem 2.4. (Note that  $\lambda_n^*$  does not necessarily have to be the same as the  $\lambda_n$  that we use for the random-weighting approach.) Firstly, due to Assumption (2.10),  $\widehat{\beta}_n^{\text{OLS}}$  is strongly consistent (Lai, Robbins, & Wei, 1978), and

$$X' \mathbf{e}_n^{\text{OLS}} = (X'Y - X'X(X'X)^{-1}X'Y) = \mathbf{0}.$$

Meanwhile, since  $\mathbb{E}(|\epsilon_i|) < \infty$  for all  $i$  and  $\lambda_n^* = o(\sqrt{n})$ ,  $\widehat{\beta}_n^{\text{LAS}}(\lambda_n^*)$  is strongly consistent (A. Chatterjee & Lahiri, 2011b), and the KKT conditions ensure that

$$\frac{1}{\sqrt{n}} \left\| X' \mathbf{e}_n^{\text{LAS}} \right\|_2 = \frac{1}{\sqrt{n}} \left\| X' \left( \mathbf{y} - X \widehat{\beta}_n^{\text{LAS}} \right) \right\|_2 \leq \frac{\lambda_n^* \sqrt{p}}{\sqrt{n}} \rightarrow 0 \quad a.s. P_D.$$

We also point out that centering on the true regression parameter

$$\sqrt{n} \left( \widehat{\beta}_n^w - \beta_0 \right).$$

results in additional terms that depend on the sample path of realized data  $\{y_1, y_2, \dots\}$ . Consequently, convergence in conditional distribution almost surely under  $P_D$  (just like the

result in Theorem 2.4) could not be achieved. We refer readers to Remark 3.1 in Chapter 3 for more details.

On the other hand, a more sophisticated argument is needed to establish the asymptotic conditional distribution for the case of  $0 < q < p$ . First, note that for  $j \in \{j : \beta_{0,j} = 0\}$ ,  $\sqrt{n}\widehat{\beta}_{n,j}^{SC}$  has an asymptotic normal distribution (denoted  $Z_j$ ) under  $P_D$ . By the Skorokhod representation theorem, there exists random variables  $U_{n,j}$  and  $U_j$  such that  $U_{n,j} \stackrel{d}{=} \sqrt{n}\widehat{\beta}_{n,j}^{SC}$ ,  $U_j \stackrel{d}{=} Z_j$ , and  $U_{n,j} \rightarrow U_j$  *a.s.*  $P_D$ . Then, for  $(\lambda_n/\sqrt{n}) \rightarrow \lambda_0 \in [0, \infty)$ ,

$$\sqrt{n} \left( \widehat{\beta}_n^w - \widehat{\beta}_n^{SC} \right) \xrightarrow{\text{c.d.}} \arg \min_{\mathbf{u}} V^*(\mathbf{u}) \quad \text{a.s. } P_D, \quad (2.13)$$

where

$$\begin{aligned} V^*(\mathbf{u}) = & -2\mathbf{u}'\Psi + \mu_W \mathbf{u}'C\mathbf{u} \\ & + \lambda_0 \sum_{j=1}^p W_j \left[ u_j \operatorname{sgn}(\beta_{0,j}) \mathbb{1}_{\{\beta_{0,j} \neq 0\}} + (|U_j + u_j| - |U_j|) \mathbb{1}_{\{\beta_{0,j} = 0\}} \right], \end{aligned}$$

for  $\Psi$  and  $\{W_j\}_{1 \leq j \leq p}$  defined in Theorem 2.4.

The results presented above fulfill our first objective to study and extend the asymptotic properties of the “one-step” random-weighting procedure that was considered by M. Newton et al. (2021). However, we also recognize that the current “one-step” random-weighting setup (2.3) in Algorithm 1 does not produce random-weighting samples that have conditional sparse normality property. From Theorems 2.2 and 2.4, it is evident that even under a fixed dimensional ( $p_n = p$ ) setting, the random weighting samples achieve conditional model selection consistency when  $\lambda_n = \mathcal{O}(n^c)$  for some  $\frac{1}{2} < c < 1$ , whereas conditional asymptotic normality happens when  $\lambda_n = o(\sqrt{n})$ .

Unsurprisingly, this finding about (lack of) conditional sparse normality approximation coincides with many existing Bayesian and frequentist results. For instance, in the Bayesian framework, Theorem 7 of Castillo, Schmidt-Hieber, and van der Vaart (2015) proved that the Bayesian LASSO approach (Park & Casella, 2008) could not achieve asymptotic sparse

normality for any one given  $\lambda_n$  due to the conflicting demands of sparsity-inducement and normality approximation on the regularization parameter  $\lambda_n$ . In the frequentist setting, Liu and Yu (2013) pointed out that there does not exist one  $\lambda_n$  that allows a standard LASSO estimator (2.2) to simultaneously achieve model selection and asymptotic normality. Consequently, many variations of “two-step” LASSO estimators (e.g., Zou (2006)’s ALasso), and their corresponding bootstrap procedures (e.g., Das, Gregory, and Lahiri (2019)’s perturbation bootstrap of ALasso) were introduced to overcome this shortcoming.

### 2.3.2. Two-step Procedure

To overcome the regularization problem, we propose an extension to random weighting in LASSO regression. We retain the random-weighting framework of repeatedly assigning random-weights and optimizing the objective function (2.3), except we propose optimization in two-steps: In step one, we optimize

$$\min_{\beta} \left\{ \sum_{i=1}^n W_i (y_i - \mathbf{x}'_i \beta)^2 + \lambda_n \sum_{j=1}^{p_n} W_{0,j} |\beta_j| \right\} \quad (2.14)$$

to select variables. Let  $\widehat{S}_n^w \subseteq \{1, \dots, p_n\}$  be the set of variables being selected in (2.14), and let  $(\widehat{S}_n^w)^c$  be the set of discarded variables. In addition, denote  $X_{\widehat{S}_n^w}$  as the  $n \times |\widehat{S}_n^w|$  submatrix of  $X$  whose columns correspond to the selected variables in (2.14). Then, in step two, we obtain our random-weighting samples by solving

$$\widehat{\beta}_n^w := \begin{bmatrix} \widehat{\beta}_{n, \widehat{S}_n^w}^w \\ \widehat{\beta}_{n, (\widehat{S}_n^w)^c}^w \end{bmatrix} := \begin{bmatrix} \left( X'_{\widehat{S}_n^w} D_n X_{\widehat{S}_n^w} \right)^{-1} X'_{\widehat{S}_n^w} D_n Y \\ \mathbf{0} \end{bmatrix}, \quad (2.15)$$

where the partition of  $\widehat{\beta}_n^w$  corresponds to  $\widehat{S}_n^w$  and  $(\widehat{S}_n^w)^c$ .

For convenience, we shall refer to this proposed extension as a “two-step procedure”, which is laid out in detail in Algorithm 2. This extension can be seen as the random-

---

**Algorithm 2** Random-Weighting in LASSO+LS regression
 

---

**Require:** data  $D = (\mathbf{y}, X)$ , regularization parameter  $\lambda_n$ , number of draws  $B$ , choice of random weight distribution  $F_W$ , choice of weighting schemes: (2.4), (2.5) or (2.6)

1: **for**  $b = 1$  to  $B$  **do**

2:   Draw i.i.d. random weights from  $F_W$  and substitute them into (2.3).

3:   Optimize (2.14) to obtain  $\widehat{S}_n^{w,b}$ .

4:   Based on the selected set of variables  $\widehat{S}_n^{w,b}$ , obtain  $\widehat{\beta}_n^{w,b}$  by solving (2.15).

5: **end for**

**Ensure:**  $B$  sets of selected variables  $\{\widehat{S}_n^{w,b}\}_{b=1}^B$ ,  $B$  parameter samples  $\{\widehat{\beta}_n^{w,b}\}_{b=1}^B$

---

weighting version of Liu and Yu (2013)'s LASSO+LS procedure, i.e., a LASSO step (2.2) for variable selection followed by a least-square estimation for the selected variables. (Belloni and Chernozhukov (2013) had also studied the finite-sample and asymptotic properties of the post-LASSO OLS estimator.) We shall denote this unweighted two-step LASSO+LS estimator as  $\widehat{\beta}_n^{LAS+LS}$ , and let  $\widehat{S}_n$  be the set of variables selected (in the first step) by this estimator. Notice that  $\widehat{S}_n$  and  $\widehat{S}_n^w$  may be different due to the presence of random-weights in the selection step of (2.14). The superscript  $w$  of  $\widehat{S}_n^w$  helps to remind readers that the set of selected variables in (2.14) could change with different sets of assigned random weights.

In this subsection, we adopt the same assumptions as we did in Theorem 2.2, including the fact that  $p_n \leq n$  and  $X$  is full rank for all  $n$ . Thus  $X_{\widehat{S}_n^w}$  is full rank and consequently,

$$X'_{\widehat{S}_n^w} D_n X_{\widehat{S}_n^w}$$

is also full rank and is invertible for all  $n$ .

For ease of presentation, we introduce a bit of additional notation. Let  $S_0$  be the true set of relevant variables. To be consistent with our previous notation, we remind readers that  $S_0 = \{1, \dots, q\}$  without loss of generality, and  $X_{S_0} = X_{(1)}$ . We also partition  $\widehat{\beta}_n^w$  and  $\widehat{\beta}_n^{LAS+LS}$  into

$$\widehat{\beta}_n^w = \begin{bmatrix} \widehat{\beta}_{n(1)}^w \\ \widehat{\beta}_{n(2)}^w \end{bmatrix} \quad \text{and} \quad \widehat{\beta}_n^{LAS+LS} = \begin{bmatrix} \widehat{\beta}_{n(1)}^{LAS+LS} \\ \widehat{\beta}_{n(2)}^{LAS+LS} \end{bmatrix}$$



respectively, which correspond to the partition of  $\beta_0 = [\beta_{0(1)} \ \beta_{0(2)}]'$ . We observe that if  $\widehat{S}_n^w = S_0$ , then

$$\widehat{\beta}_{n, \widehat{S}_n^w}^w = \widehat{\beta}_{n(1)}^w \quad \text{and} \quad \widehat{\beta}_{n, (\widehat{S}_n^w)^c}^w = \widehat{\beta}_{n(2)}^w = \beta_{0(2)} = \mathbf{0}.$$

Similarly, if  $\widehat{S}_n = S_0$ , then

$$\widehat{\beta}_{n, \widehat{S}_n}^{LAS+LS} = \widehat{\beta}_{n(1)}^{LAS+LS} \quad \text{and} \quad \widehat{\beta}_{n, (\widehat{S}_n)^c}^{LAS+LS} = \widehat{\beta}_{n(2)}^{LAS+LS} = \beta_{0(2)} = \mathbf{0}.$$

We are now ready to establish the conditional sparse normality property of the two-step random-weighting samples (2.15) under growing  $p_n$  setting with appropriate regularity conditions.

**Theorem 2.5. (Conditional Sparse Normality)** *Adopt all regularity assumptions as stated in Theorem 2.2 (including assumptions about the different rates of  $\lambda_n$  and  $p_n$  for weighting schemes (2.4), (2.5) and (2.6)). Furthermore, assume  $\mu_W = 1$  and  $C_{n(11)} \rightarrow C_{11}$  for some nonsingular matrix  $C_{11}$ . Let  $\widehat{\beta}_n^w$  be the two-step random-weighting samples defined in (2.15), and let  $\widehat{\beta}_n^{LAS+LS}$  be the unweighted two-step LASSO+LS estimator (i.e. a LASSO variable selection step (2.2) followed by least-squares estimation for the selected variables). Then,*

$$P\left(\widehat{S}_n^w = S_0 \mid \mathcal{F}_n\right) \rightarrow 1 \quad a.s. \ P_D,$$

and

$$\sqrt{n} \left( \widehat{\beta}_{n(1)}^w - \widehat{\beta}_{n(1)}^{LAS+LS} \right) \xrightarrow{c.d.} N_q \left( \mathbf{0}, \sigma_W^2 \sigma_\epsilon^2 C_{11}^{-1} \right) \quad a.s. \ P_D.$$

Theorem 2.5 highlights the improvement brought about by the extended random-weighting framework as compared to the original “one-step” procedure considered by M. Newton et al. (2021). With a common regularization parameter  $\lambda_n$  (and all regularity conditions that apply), the two-step random-weighting samples attain conditional model selection consistency and achieve conditional asymptotic normality (by centering at the unweighted two-step LASSO+LS estimator) on the true support  $S_0$  under growing  $p_n$  setting.

We acknowledge that the convergence rate of  $P\left(\widehat{S}_n^w = S_0 | \mathcal{F}_n\right)$  is rather slow; see Lemma 3.11 for more details.

We conclude this section by establishing that the random-weighting samples from the two-step procedure also achieve the conditional consistency property under growing  $p_n$  setting. This could be viewed as an improvement to the result that we have in Theorem 2.3(a) which applies to fixed dimensional setting only.

**Theorem 2.6. (Conditional Consistency)** *Adopt all regularity assumptions as stated in Theorem 2.2 (including assumptions about the different rates of  $\lambda_n$  and  $p_n$  for weighting schemes (2.4), (2.5) and (2.6)). Let  $\widehat{\beta}_n^w$  be the two-step random-weighting samples defined in (2.15). Then*

$$\left\| \widehat{\beta}_n^w - \beta_0 \right\|_2 \xrightarrow{c.p.} 0 \quad a.s. \ P_D.$$

Theorem 2.6 indicates a concentration of the conditional distribution of  $\widehat{\beta}_n^w$  near  $\beta_0$  with increasing sample size given almost any data set.

### 2.3.3. Remarks

The two-step random-weighting procedure is a valid bootstrap procedure for Liu and Yu (2013)'s LASSO+LS estimator  $\widehat{\beta}_n^{LAS+LS}$  under growing  $p_n$  setting. Using very similar regularity assumptions, Liu and Yu (2013) showed that their LASSO+LS method gives consistent model selection under  $P_D$ , and

$$\sqrt{n} \left( \widehat{\beta}_{n(1)}^{LAS+LS} - \beta_{0(1)} \right)$$

converges to  $N(\mathbf{0}, \sigma_\epsilon^2 C_{11}^{-1})$  under  $P_D$ . Hence, based on Theorem 2.5, by fulfilling the appropriate regularity assumptions and drawing random weights from  $F_W$  with unitary mean and variance ( $\mu_W = \sigma_W^2 = 1$ ), the conditional distribution of the two-step random-weighting samples  $\widehat{\beta}_n^w$  converges to the same distributional limit of the LASSO+LS estimator under  $P_D$ . This enables the two-step random-weighting procedure to produce bootstrap

samples that provide valid distributional approximation to the LASSO+LS estimator.

We point out that by capitalizing on the sub-Gaussian nature of  $\epsilon$ , Liu and Yu (2013)'s proposed residual bootstrap procedure for their LASSO+LS estimator works under high-dimensional setting where  $p_n$  grows nearly exponential with sample size  $n$ . On the other hand, in this paper, we only require finite fourth moment assumptions for both error term  $\epsilon$  and random weights  $\mathbf{W}$ , and our random-weighting procedure only allows  $p_n$  to grow at a polynomial rate of  $o(\sqrt{n})$ .

Similarly, under fixed dimensional ( $p_n = p$ ) setting where  $\beta_0$  is not sparse (i.e.  $q = p$ ), our one-step random-weighting approach in Algorithm 1 could also be a valid bootstrap procedure for the standard LASSO estimator  $\widehat{\beta}_n^{\text{LAS}}(\lambda_n)$ . Specifically, Knight and Fu (2000) proved that for  $(\lambda_n/\sqrt{n}) \rightarrow \lambda_0 \in [0, \infty)$ ,

$$\sqrt{n} \left( \widehat{\beta}_n^{\text{LAS}}(\lambda_n) - \beta_0 \right)$$

converges to the same distributional limit stated in Theorem 2.4 under  $P_D$ . However, for the case where  $q < p$ , the one-step random-weighting procedure no longer provides valid distributional approximation to  $\widehat{\beta}_n^{\text{LAS}}(\lambda_n)$ , as evident from the Skorokhod argument. This mimics the asymptotic conditional distribution of the LASSO parametric residual bootstrap (Knight & Fu, 2000).

## 2.4. Numerical Experiments

We perform simulation studies and data analysis using R (R Core Team, 2019); all source code is available at the Github public repository: <https://github.com/wiscstatman/optimizetointegrate/tree/master/Tun>.

### 2.4.1. Simulation: Part I

A simulation study of one-step random-weighting procedures (Algorithm 1) was previously reported (M. Newton et al., 2021), and so here we study performance of the two-step random-

weighting procedure (Algorithm 2) for all three weighting schemes (2.4), (2.5) and (2.6) – denoted RW1, RW2 and RW3 respectively – in several experimental settings, and compare it with:

- Bayesian LASSO (Park & Casella, 2008), which can be easily implemented with R package `monomvn` (Gramacy, Moler, & Turlach, 2019)
- parametric residual bootstrap (Knight & Fu, 2000), which is a very common and easily implementable bootstrap procedure in LASSO regression. We denote this method as RB thereafter.

We drew inspiration from Das and Lahiri (2019), Liu and Yu (2013) and M. Newton et al. (2021) in setting up our simulation schemes. Specifically, we consider 8 simulation settings as tabulated in Table 2.1. In all settings, the generative state  $\beta_0 = (\beta_{0,1}, \dots, \beta_{0,p})'$  is defined as  $\beta_{0,j} = (3/4) + (1/4)j$  for  $j = 1, \dots, q$  and  $\beta_{0,j} = 0$  for  $j = q + 1, \dots, p$ . The predictors  $\mathbf{x}_i$  are drawn from  $p$ -variate normal distribution with different covariance structures.  $\Sigma^{(1)}$  has the following structure

$$\Sigma_{i,j}^{(1)} = \mathbb{1}_{\{i=j\}} + \mathbb{1}_{\{i \neq j\}} \times \left( 0.3^{|i-j|} \mathbb{1}_{\{i \leq q\}} \mathbb{1}_{\{j \leq q\}} \right) \quad \text{for } 1 \leq i, j \leq 10. \quad (2.16)$$

$\Sigma^{(3)}$  also has the same structure as (2.16), except that it has larger dimension  $p = 50$ . Meanwhile,  $\Sigma^{(2)}$  has the following structure: for  $1 \leq i, j \leq 10$ ,

$$\Sigma_{i,j}^{(2)} = \mathbb{1}_{\{i=j\}} + \mathbb{1}_{\{i \neq j\}} \times \left[ 0.4 \mathbb{1}_{\{i \leq q\}} \mathbb{1}_{\{j \leq q\}} + 0.5 (1 - \mathbb{1}_{\{i \leq q\}} \mathbb{1}_{\{j \leq q\}}) \right].$$

We verify that only simulation settings 5 and 6 violate the strong irrerepresentable condition (2.11), whereas the other six simulation settings satisfy assumption (2.11). By simulating i.i.d.  $\epsilon_i$  and  $\mathbf{x}_i$ , we generate  $y_i = \mathbf{x}_i \beta_0 + \epsilon_i$  for  $i = 1, \dots, n$ .

**Purpose of simulation setup:** The even-numbered simulation settings share the same specifications as their odd-numbered counterparts except with larger sample size  $n$  (e.g. Setting 2 versus Setting 1, Setting 4 versus Setting 3, et cetera). Simulation Settings 3 and 4

Table 2.1: Simulation Settings

Setting	$n$	$p$	$q$	$\epsilon_i$	$\mathbf{x}_i \sim N_p(\mathbf{0}, \Sigma)$
1	100	10	6	$N(0, 1)$	$\Sigma = \Sigma^{(1)}$
2	500	10	6	$N(0, 1)$	$\Sigma = \Sigma^{(1)}$
3	100	10	6	$\chi_2^2 - 2$	$\Sigma = \Sigma^{(1)}$
4	500	10	6	$\chi_2^2 - 2$	$\Sigma = \Sigma^{(1)}$
5	100	10	6	$N(0, 1)$	$\Sigma = \Sigma^{(2)}$
6	500	10	6	$N(0, 1)$	$\Sigma = \Sigma^{(2)}$
7	100	50	6	$N(0, 1)$	$\Sigma = \Sigma^{(3)}$
8	500	50	6	$N(0, 1)$	$\Sigma = \Sigma^{(3)}$

are used as an example of cases where the error term  $\epsilon$  is no longer normally distributed, whereas Simulation Settings 5 and 6 are set up to illustrate the situations where the strong irrerepresentable condition (2.11) is violated. Finally, we increase the dimension  $p$  of predictors by five-fold in Settings 7 and 8 to compare performances in higher-dimensional setting.

For each simulation setting, we generate  $T = 500$  independent datasets. For each simulated data set, we draw  $B = 1000$  posterior/bootstrap samples from the 5 aforementioned methods: Bayesian LASSO (BLASSO), two-step random-weighting with schemes (2.4), (2.5) and (2.6), and residual bootstrap. For the Bayesian LASSO procedure, we specify a 2000 burn-in period. In addition, Bayesian LASSO imposes a noninformative marginal prior on  $\sigma_\epsilon^2$ ,  $\pi(\sigma_\epsilon^2) \sim 1/\sigma_\epsilon^2$ , and a Jeffrey's prior on  $\lambda_n$ . To induce sparsity in the MCMC samples of  $\beta$ , the posterior distribution is sampled by a Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm (Green, 1995), with a uniform prior specified on the number of non-zero coefficients to be included in the model. For the three random-weighting schemes, all i.i.d. random weights are drawn from a standard exponential distribution. The regularization parameter  $\lambda_n$  is chosen via cross-validation using Liu and Yu (2013)'s (unweighted)

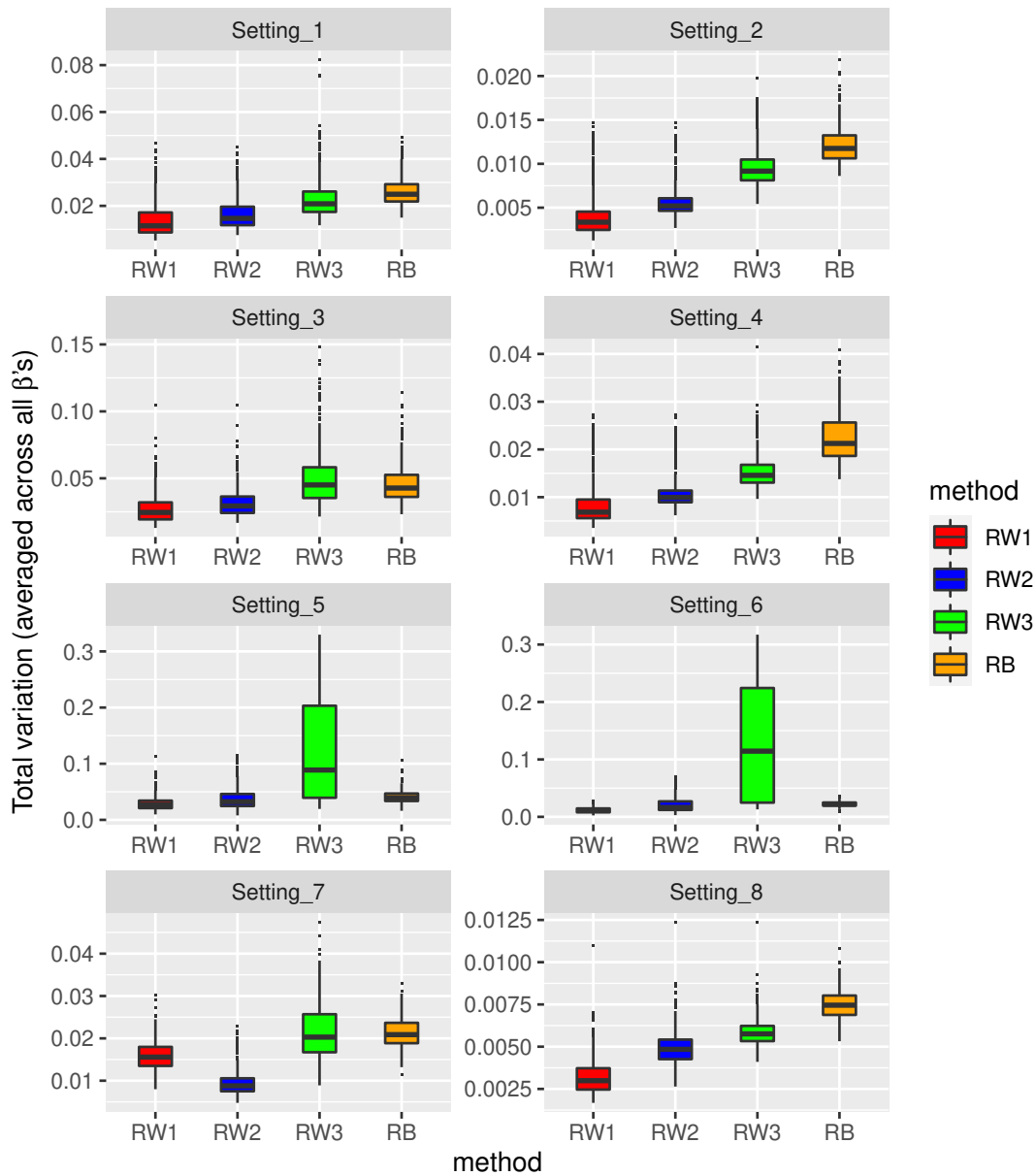


Figure 2.1: Simulation Part I: Sampling distribution of total variation distance between the random-weighting distribution and a Bayesian posterior (averaged across all  $\beta$ 's) among  $T = 500$  simulated data sets in 8 simulation settings between ecdf of MCMC samples and ecdf of samples from each of the 4 methods: two-step random-weighting approach using weighting schemes (2.4) (denoted RW1), (2.5) (denoted RW2) and (2.6) (denoted RW3), and LASSO residual bootstrap (denoted RB).

LASSO+LS procedure, and then the same  $\lambda_n$  is used to draw the 1000 random-weighting samples according to Algorithm 2. We note that the optimization step (2.14) can be easily computed using R package `glmnet` (Friedman et al., 2010). Meanwhile for residual bootstrap, its regularization parameter  $\lambda_n^{\text{RB}}$  is chosen via cross-validation using standard LASSO, and values of  $\lambda_n^{\text{RB}}$  are thereafter fixed for all bootstrap computations on the same dataset.

For each of the five aforementioned methods, we obtain  $\{\widehat{\beta}_j^{(b,t)}\}$  that represents the  $j^{\text{th}}$  component of sampled/bootstrapped  $\beta$  in the  $b^{\text{th}}$  iteration for the  $t^{\text{th}}$  simulated data set, where  $j = 1, \dots, p$ , and  $b = 1, \dots, B$ , and  $t = 1, \dots, T$ . To be precise, we have

$$\left\{ \widehat{\beta}_{j(\text{MCMC})}^{(b,t)}, \widehat{\beta}_{j(\text{RW1})}^{(b,t)}, \widehat{\beta}_{j(\text{RW2})}^{(b,t)}, \widehat{\beta}_{j(\text{RW3})}^{(b,t)}, \widehat{\beta}_{j(\text{RB})}^{(b,t)} \right\}$$

that correspond to the sampled/bootstrapped  $\beta$ 's of the five aforementioned methods, but for brevity we drop the subscripts whenever it does not cause any confusion, since each method is subject to the same performance evaluation. We then assess the performances of each of these five methods – BLASSO, RW1, RW2, RW3 and RB – in each of the 8 simulation settings using the following comparison criteria:

- Estimation MSE of coefficients. Specifically, for each simulated data set  $t = 1, \dots, T$ , we keep track of

$$\text{MSE}^{(t)} = \frac{1}{B} \sum_{b=1}^B \left\| \mathbf{Y}^{(t)} - X^{(t)} \widehat{\beta}^{(b,t)} \right\|_2^2.$$

- Out-of-sample prediction MSE (abbreviated as MSPE thereafter), where test sets are of the same size as the corresponding training sets. Similarly, for each simulated data set  $t = 1, \dots, T$ , we keep track of

$$\text{MSPE}^{(t)} = \frac{1}{B} \sum_{b=1}^B \left\| \mathbf{Y}_{\text{test}}^{(t)} - X_{\text{test}}^{(t)} \widehat{\beta}^{(b,t)} \right\|_2^2.$$

- Conditional (on data) probability of selecting the  $j^{\text{th}}$  variable where  $j = 1, \dots, p$ .

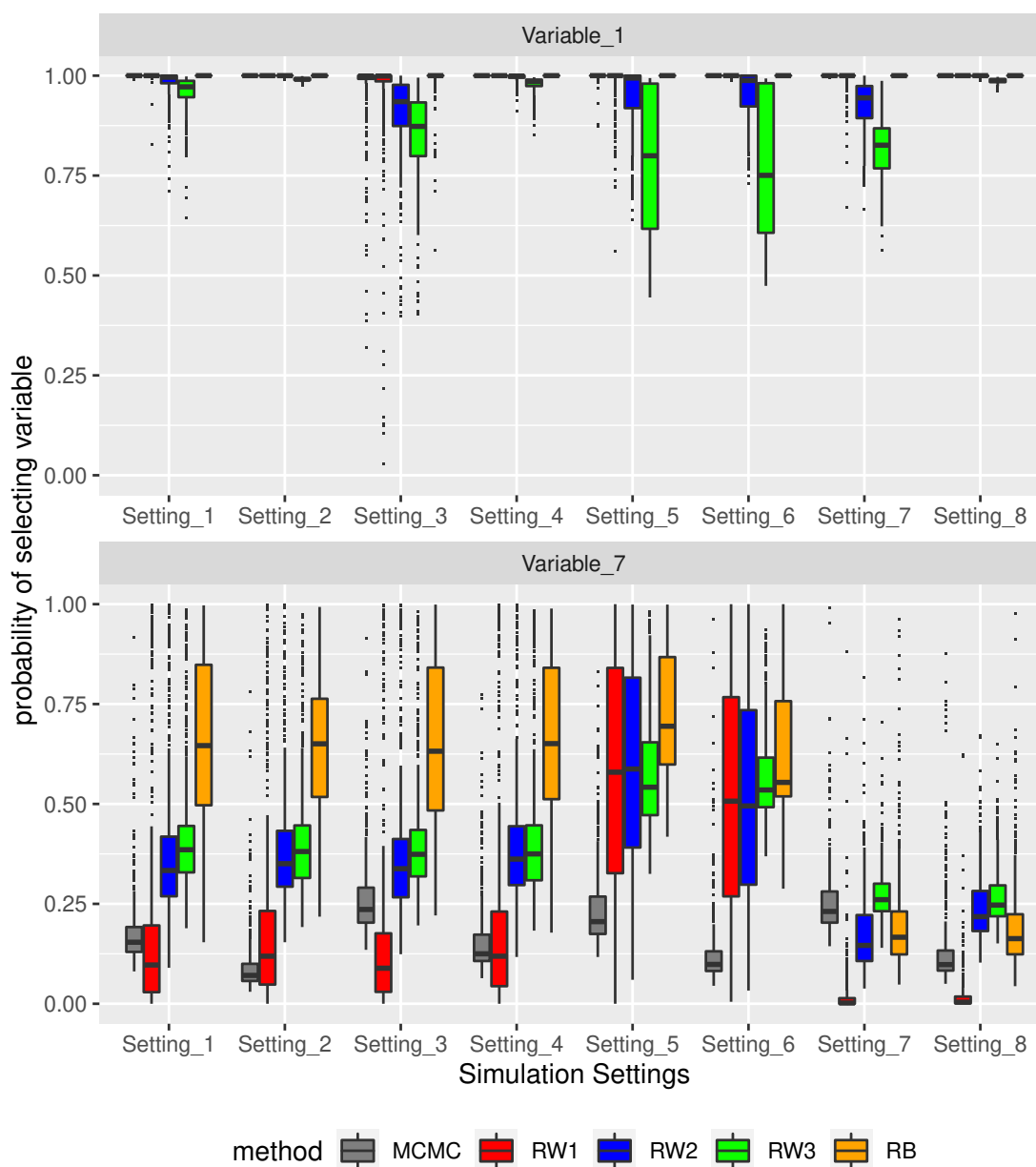


Figure 2.2: Simulation Part I: Sampling distribution of conditional (on data) probabilities of selecting  $\beta_1$  and  $\beta_7$  among  $T = 500$  simulated data sets in 8 simulation settings by the 5 methods: MCMC via Bayesian LASSO, two-step random-weighting approach using weighting schemes (2.4) (denoted RW1), (2.5) (denoted RW2) and (2.6) (denoted RW3), and LASSO residual bootstrap (denoted RB).



Specifically, for each simulated data set  $t = 1, \dots, T$ , we keep track of

$$\hat{p}_j^{(t)} := \frac{1}{B} \left| \left\{ b : \widehat{\beta}_j^{(b,t)} \neq 0 \right\} \right|.$$

We note that the computation of  $\hat{p}_j^{(t)}$  is sensible because all the five methods (including BLASSO with RJMCMC implementation) induce sparsity in the sampled/bootstrapped  $\beta$ 's.

- Coverage and average width of the two-sided 90% credible/confidence interval (CI) for the  $j^{\text{th}}$  variable where  $j = 1, \dots, p$ . Specifically, denote  $\hat{r}_{0.05,j}^{(t)}$  and  $\hat{r}_{0.95,j}^{(t)}$  as the 5<sup>th</sup> percentile and 95<sup>th</sup> percentile of the empirical distribution of  $\{\widehat{\beta}_j^{(b,t)}\}_{1 \leq b \leq B}$ . Then, the average width (across  $T = 500$  simulated data sets) of the two-sided 90% CI for the  $j^{\text{th}}$  variable is computed as

$$\hat{l}_j := \frac{1}{T} \sum_{t=1}^T \left( \hat{r}_{0.95,j}^{(t)} - \hat{r}_{0.05,j}^{(t)} \right),$$

and its corresponding empirical coverage is calculated as

$$\hat{q}_j := \frac{1}{T} \left| \left\{ t : \hat{r}_{0.05,j}^{(t)} \leq \beta_{0,j} \leq \hat{r}_{0.95,j}^{(t)} \right\} \right|.$$

In addition, we obtain the total variation distance between empirical cumulative distribution function (ecdf) of MCMC samples and ecdf of samples produced by one of the other four methods – the two-step random-weighting (RW1, RW2 and RW3) and residual bootstrap (RB). The intent is to assess how well the random-weighting methods approximate the MCMC-approximated posterior. Specifically, for the  $j^{\text{th}}$  variable in the  $t^{\text{th}}$  simulated data set, let

$$\hat{F}_{j(MCMC)}^{(t)} = \text{ecdf of } \left\{ \widehat{\beta}_{j(MCMC)}^{(b,t)} \right\}_{1 \leq b \leq B},$$

and let  $\hat{F}_{j(\cdot)}^{(t)}$  be the ecdf of samples produced by one of the other 4 methods: RW1, RW2, RW3 or RB. Note that the ecdf's are easily obtained via the function `ecdf` in R base package

(R Core Team, 2019). Then, for each of the 4 methods, we keep track of the total variation (averaged across all  $p$  variables) for each simulated data set  $t = 1, \dots, T$ :

$$\text{TV}^{(t)} = \frac{1}{p} \sum_{j=1}^p \frac{1}{2} \sum_{\omega \in \Omega} \left| \hat{F}_{j(MCMC)}^{(t)}(\omega) - \hat{F}_{j(\cdot)}^{(t)}(\omega) \right|,$$

where the inner summation is approximated using a trapezoidal rule with an interval width of 0.001.

Table 2.2: Empirical coverage  $\hat{q}_j$  and average width  $\hat{l}_j$  (in parentheses) of the two-sided 90% CI for the first 10 variables in Simulation Setting 8, using the five approaches: MCMC via BLASSO, two-step random-weighting approach using weighting schemes (2.4) (denoted RW1), (2.5) (denoted RW2) and (2.6) (denoted RW3), and LASSO residual bootstrap (denoted RB).

$\beta_{0,j}$	MCMC	RW1	RW2	RW3	RB
1.00	0.918 (0.161)	0.878 (0.152)	0.882 (0.152)	0.906 (0.16)	0.344 (0.153)
1.25	0.908 (0.169)	0.88 (0.158)	0.876 (0.159)	0.904 (0.168)	0.588 (0.16)
1.50	0.894 (0.168)	0.864 (0.158)	0.868 (0.158)	0.886 (0.165)	0.578 (0.16)
1.75	0.918 (0.168)	0.886 (0.159)	0.892 (0.159)	0.9 (0.165)	0.596 (0.16)
2.00	0.922 (0.168)	0.894 (0.159)	0.882 (0.159)	0.898 (0.164)	0.556 (0.16)
2.25	0.886 (0.161)	0.866 (0.151)	0.872 (0.152)	0.874 (0.157)	0.35 (0.153)
0.00	1 (0.04)	1 (0.016)	1 (0.096)	1 (0.099)	0.998 (0.023)
0.00	1 (0.041)	0.998 (0.018)	1 (0.097)	1 (0.1)	1 (0.024)
0.00	1 (0.04)	1 (0.015)	1 (0.097)	1 (0.099)	1 (0.023)
0.00	0.998 (0.04)	1 (0.015)	1 (0.097)	1 (0.1)	1 (0.023)

Firstly, as expected, performance improves with larger sample size  $n$ , such as smaller

MSE's, smaller MSPE's, higher coverage probabilities and narrower CI's. Secondly, we note that the MSE's and MSPE's are very similar among all the five methods in all 8 simulation settings (figures not shown). However, the two-step random-weighting approach, especially weighting schemes (2.4) and (2.5) – denoted RW1 and RW2, outperforms the LASSO residual bootstrap (denoted RB) in all other performance measures.

Figure 2.1 displays the sampling distribution of total variation distance  $\{TV^{(t)}\}_{1 \leq t \leq T}$  between the random-weighting distribution and a Bayesian posterior (averaged across all  $\beta$ 's), among the  $T = 500$  simulated data sets in the 8 simulation settings for the 4 methods: RW1, RW2, RW3 and RB. Generally, larger sample size  $n$  leads to smaller total variations. Moreover, in all simulation settings, RW1 and RW2 have smaller total variations than that of RB, which illustrates the viability of the two-step random-weighting samples to approximate posterior inference. RW3 has larger total variations especially in Settings 5 and 6, where the strong irrepresentable condition (2.11) is violated. This illustrates the need for restrictive regularity assumption for weighting scheme (2.6) that we highlighted in part (c) of Theorem 2.2.

In Figure 2.2, we show the sampling distributions of  $\{\hat{p}_1^{(t)}\}_{1 \leq t \leq T}$  and  $\{\hat{p}_7^{(t)}\}_{1 \leq t \leq T}$  among the  $T = 500$  simulated data sets in the 8 simulation settings for all the five methods. Recall that the first variable corresponds to  $\beta_{0,1} = 1$  and the seventh variable corresponds to  $\beta_{0,7} = 0$ . Sampling distribution of conditional (on data) probabilities of selecting other relevant predictors is similar to that of the first variable, and sampling distribution of conditional probabilities of selecting other irrelevant predictors is similar to that of the seventh variable. In all 8 simulation settings, all methods almost always select the first variable, except for RW3 in Simulation Settings 5 and 6, due to the violation of condition (2.11). However, similar to MCMC, the two-step random-weighting schemes (especially RW1) have lower conditional probabilities of selecting the seventh variable (which is an irrelevant predictor) than the LASSO RB. This illustrates that the two-step random-weighting approach is more capable of discarding irrelevant variables as compared to LASSO residual bootstrap. Only in Simulation Settings 5 and 6 do we see similarly high conditional probabilities of selecting

the seventh variable among RW1, RW2, RW3 and RB, due to violation of condition (2.11).

Empirical coverage and average width of the two-sided 90% CI's for relevant predictors (i.e.  $\beta_{0,j} \neq 0$ ) paint a similar story. For illustration, the empirical coverage  $\hat{q}_j$  and average width  $\hat{l}_j$  (in parentheses) of the two-sided 90% CI for the first 10 variables, i.e. for  $j = 1, \dots, 10$ , in Simulation Setting 8, are tabulated in Table 2.2. Generally, average widths of CI's are similar among all five methods in all but two simulation settings, where RW3 has much wider 90% CI's in Simulation Settings 5 and 6. Interestingly, empirical coverage for MCMC and random-weighting samples is similar and close to 90%, but the LASSO residual bootstrap samples always have the lowest empirical coverage, especially in Simulation Settings 7 and 8, where their empirical coverage is only around 30% - 40%.

#### 2.4.2. Simulation: Part II

On a separate calculation, we use Simulation Setting 2 (see Table 2.1) to illustrate that there are computational advantages in using  $\lambda_n$  chosen via cross-validation on the unweighted LASSO+LS procedure (Liu & Yu, 2013), instead of cross-validation on the standard LASSO method, for obtaining the two-step random-weighting samples. For brevity, we shall refer to the former as the two-step cross validation, and the latter as the one-step cross validation.

Specifically, for each of the  $T = 500$  simulated data sets under Simulation Setting 2, we repeat the two-step random-weighting calculations outlined in Algorithm 2, but with  $\lambda_n$  chosen via cross-validation on the standard LASSO method. This is in fact the same regularization parameter  $\lambda_n^{RB}$  that we used to generate the residual bootstrap samples.

We find from the simulation results that the two-step cross-validation leads to larger  $\lambda_n$  as compared to the one-step cross-validation. This ties back to the conflicting demands of the standard LASSO method on  $\lambda_n$ : smaller  $\lambda_n$  allows more variables into the model to reduce estimation MSE; and larger  $\lambda_n$  enables more regularization to discard irrelevant variables. On the other hand, using a two-step LASSO+LS procedure frees up these conflicting constraints on  $\lambda_n$ .

For these two sets of random-weighting samples, we repeat the same calculations of

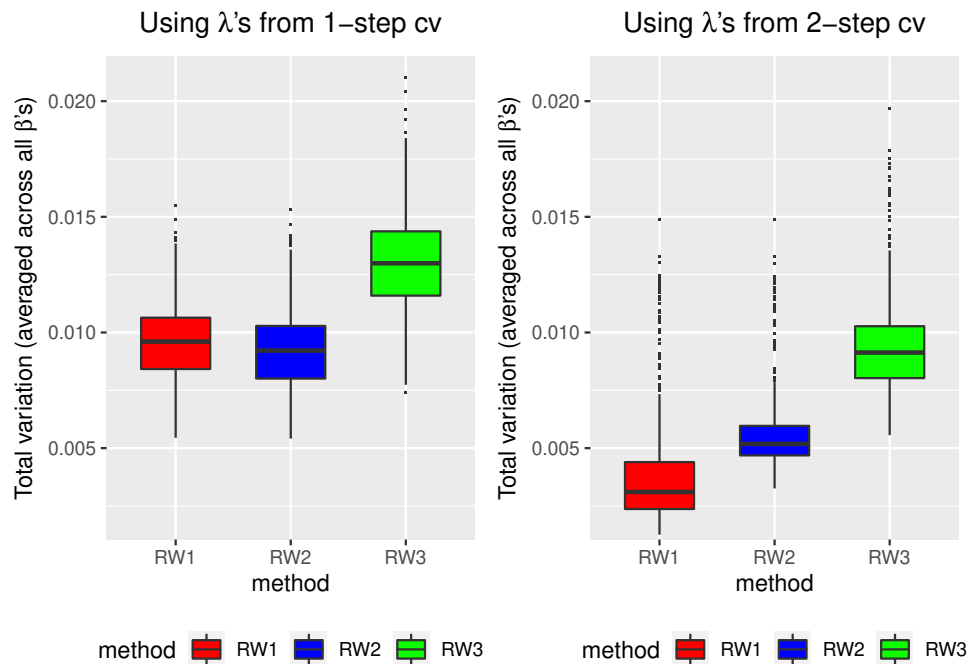


Figure 2.3: Simulation Part II: Sampling distribution of total variation distance between random-weighting distribution and a Bayesian posterior (averaged across all  $\beta$ 's) among  $T = 500$  simulated data sets in Simulation Setting 2 between ecdf of MCMC samples and ecdf of the two-step random-weighting samples, computed with  $\lambda_n$  obtained via 1-step cross validation or 2-step cross validation, using weighting schemes (2.4) (2.5) and (2.6) (denoted RW1, RW2 and RW3 respectively).

performance measures as we did in Part I of our simulation studies. We found out that MSE's, MSPE's and empirical coverage of the two-sided 90% CI are very similar between these two sets of random-weighting samples. However, from Figure 2.3, we see that larger regularization  $\lambda_n$  based on the two-step cross validation leads to lower total variation distance between random-weighting distribution and a Bayesian posterior. Meanwhile, in Figure 2.4, the random-weighting samples computed with the larger  $\lambda_n$  have much lower conditional probabilities of selecting irrelevant variables (variables 7 – 10), whilst almost always selecting relevant predictors (variables 1 – 6). This also helps to illustrate the fact that the two-step random-weighting approach is able to utilize more regularization to discard irrelevant predictors while maintaining estimation accuracy.

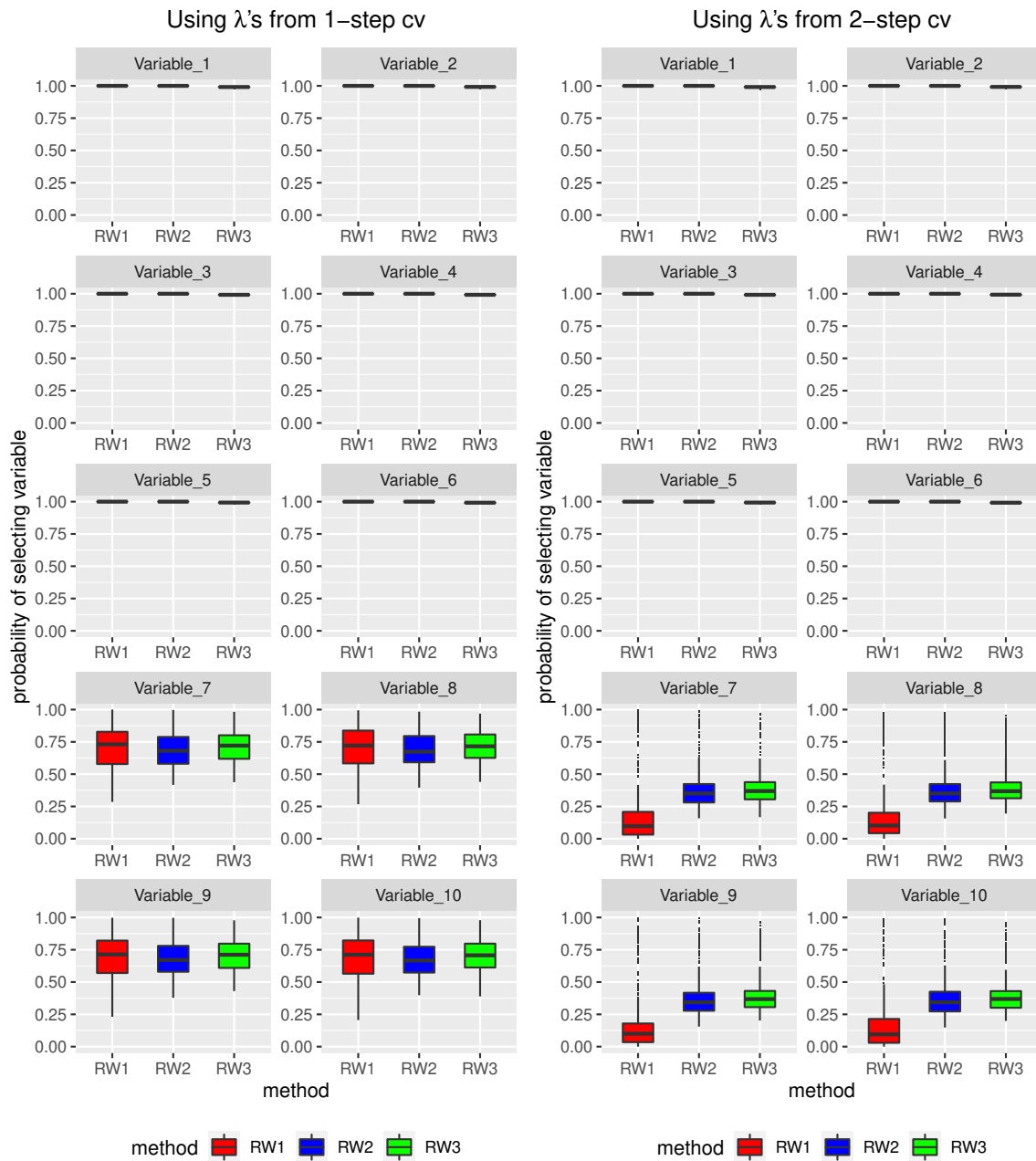


Figure 2.4: Simulation Part II: Sampling distribution of conditional (on data) probabilities of selecting  $\beta$ 's among  $T = 500$  simulated data sets in Simulation Setting 2 by the two-step random-weighting approach, computed with  $\lambda_n$  obtained via 1-step cross validation or 2-step cross validation, using weighting schemes (2.4) (2.5) and (2.6) (denoted RW1, RW2 and RW3 respectively).

### 2.4.3. Benchmark data example

To further illustrate the two-step random-weighting methodology, we apply it to the often-analyzed Boston Housing data set, which is available in the R package MASS (Venables & Ripley, 2002). Data from  $n = 506$  housing prices in the suburbs of Boston are available, with response the median value of owner-occupied homes in \$1000's, and with 13 variables ( $p = 13$ ) listed in Table 2.3.

Table 2.3: Variables in Boston Housing Data Set

Abbreviation	Variable
crim	per capita crime rate by town
zn	proportion of residential land zoned for lots over 25,000 sq.ft.
indus	proportion of non-retail business acres per town
chas	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
nox	nitrogen oxides concentration (parts per 10 million)
rm	average number of rooms per dwelling
age	proportion of owner-occupied units built prior to 1940
dis	weighted mean of distances to five Boston employment centers
rad	index of accessibility to radial highways
tax	full-value property-tax rate per \$10,000
ptratio	pupil-teacher ratio by town
Black	proportion Black residents by town
lstat	lower status of the population (percent)

Again, we apply Bayesian LASSO, the random-weighting approach for all three weighting schemes (2.4), (2.5) and (2.6) according to Algorithm 2, as well as the parametric residual bootstrap method (Knight & Fu, 2000) with  $B = 1000$ . We use the same prior specifications as well as RJMCMC implementation for Bayesian LASSO as we did in our simulation studies. For the random-weighting approach, random weights are drawn from a standard exponential distribution, and the regularization parameter is chosen with cross-validation using Liu and Yu (2013)'s unweighted LASSO+LS procedure (i.e. 2-step cross-validation). Meanwhile, for residual bootstrap, its regularization parameter is chosen via cross-validation using standard LASSO.

Figure 2.5 shows the marginal posterior distributions of  $\beta$ 's sampled from MCMC as well as the marginal conditional (on data) distributions of  $\beta$ 's obtained from the random-

weighting methods and the parametric residual bootstrap. For most of the coefficients, there is very good agreement among the methods. One notable feature is that the parametric residual bootstrap approach induces the least sparsity among all five methods for variables `indus` and `age`. In addition, Bayesian LASSO appears to introduce slightly more sparsity than the random-weighting schemes for the variable `age`. Besides that, random-weighting with different penalty weights (2.6) appears to produce lower outliers for variables `crim`, `indus` and `ptratio`.

## 2.5. Discussion

The findings above extend what is known about asymptotic conditional sampling distribution of random-weighting solutions in LASSO regression, and thereby contribute to our understanding of uncertainty quantification in penalized estimation settings. Because random weighting is readily deployed in contemporary applications involving large-scale optimization, further work is warranted that sheds more light on the random-weighting approach and its links with bootstrap and Bayesian approaches.

### Connection to Bayes

In fixed dimensional ( $p_n = p$ ) setting where  $\beta_0$  is not sparse (i.e.  $q = p$ ), Theorems 2.3 and 2.4 describe the first order behavior of the conditional distribution of the one-step random-weighting samples  $\hat{\beta}_n^w$ . Under typical parametric Bayesian inference for  $\beta$  in the linear model (2.1), for any prior measure of  $\beta$  that is absolutely continuous in a neighborhood of  $\beta_0$  with a continuous positive density at  $\beta_0$ , the Bernstein-von Mises Theorem (e.g., Theorem 10.1 of van der Vaart (1998)) ensures that for every Borel set  $A \subset \Theta \subset \mathbb{R}^p$ ,

$$P \left[ \sqrt{n} \left( \beta - \hat{\beta}_n^{\text{MLE}} \right) \in A \mid \mathcal{F}_n \right] \rightarrow P [Z \in A]$$

along almost every sample path, where  $Z \sim N(\mathbf{0}, \sigma_\epsilon^2 C^{-1})$ . Hence, based on Theorem 2.4 (with centering on  $\hat{\beta}_n^{\text{MLE}} = \hat{\beta}_n^{\text{OLS}}$ ), for any  $\lambda_n = o(\sqrt{n})$ , by drawing random weights from



$F_W$  with unitary mean and variance ( $\mu_W = \sigma_W^2 = 1$ ), the conditional distribution of the one-step random-weighting samples  $\hat{\beta}_n^w$  converges to the same limit as in the Bernstein-von Mises Theorem, i.e., the conditional distribution of  $\hat{\beta}_n^w$  is the same – at least up to the first order – as the posterior distribution of  $\beta$  under the regime of Bayesian inference.

Theorem 2.4 (with centering on  $\hat{\beta}_n^{\text{MLE}}$ ) highlights an important implication for the choice of  $F_W$  in deploying the random-weighting approach to approximate posterior inference. Specifically, non-unitary mean or variance of the random weights would cause the random-weighting samples to converge to a conditional normal distribution with an asymptotic variance that is different from the one guaranteed by the Bernstein-von-Mises Theorem.

M. A. Newton and Raftery (1994)'s first-order approximation theory for the random-weighting method relies on some classical regularity assumptions that do not hold in the LASSO setting studied here (2.2). The present work therefore extends the range of cases in which random-weighting operates successfully in large samples to achieve approximate Bayesian inference.

Comparison of random weighting and posterior distribution is less straightforward in cases where  $\beta_0$  is sparse. Castillo et al. (2015) used a mixture of point masses at zero and continuous distributions as a sparse prior in their full Bayesian procedures for high-dimensional sparse linear regression. For this sparse prior, they showed that the resulting posterior distribution is not approximated by a non-singular normal, but by a random mixture of different dimensional normal distributions. Whilst we do not have an explicit result on the distributional approximation for  $\hat{\beta}_n^w$  in growing- $p_n$  setting (e.g., Theorem 6 of Castillo et al. (2015)), our Theorem 2.5 ensures that the conditional distribution of  $\hat{\beta}_n^w$  does amass around the true support of  $\beta$ , and on the true support, the random-weighting samples attain asymptotic Gaussian distributional behavior. Theorem 3.4 is therefore comparable to Corollary 2 in Castillo et al. (2015), although different techniques are deployed; for instance we consider almost sure weak conditional convergence, whereas Castillo et al. (2015) considers sample average total-variation distance convergence, and we have no explicit prior structure. Yet the basic message of both is that the mass of the posterior

distribution, on the one hand, and the random-weighting distribution, on the other, are similarly concentrating on the correct model subset according to the same Gaussian law. We also acknowledge the fact that these Bayesian models could handle high-dimensional problem where  $p_n$  grows nearly exponential with sample size  $n$  by using sparse-inducing priors on  $\beta$ . On the other hand, our results require  $p_n$  to grow at a polynomial rate of  $o(\sqrt{n})$ .

### **Perturbation bootstrap (in general)**

Whilst the random-weighting approach has a Bayesian justification, its resemblance to existing bootstrap algorithms, especially the perturbation bootstrap, warrants a comparison with non-Bayesian bootstrap literature. The (naive) perturbation bootstrap was introduced by Jin, Ying, and Wei (2001) as a method to estimate sampling distributions of estimators related to  $U$ -process-structured objective functions. S. Chatterjee and Bose (2005) established first-order distributional consistency of a generalized perturbation bootstrap technique in M-estimation where they allowed both  $n \rightarrow \infty$  and  $p_n \rightarrow \infty$ . That paper also pointed out that for broader classes of models, the generalized bootstrap method is not second-order accurate without appropriate bias-correction and studentization. In particular, the work in (naive) perturbation bootstrap resembles the Bayesian NPL objective function (Fong et al., 2019). Subsequently, Minnier, Tian, and Cai (2011) proved the first-order distributional consistency of the perturbation bootstrap for Zou (2006)'s Adaptive LASSO (ALasso) and Fan and Li (2001)'s smoothly clipped absolute deviation (SCAD) under fixed- $p$  setting in order to construct accurate confidence regions for ALasso and SCAD estimators. Again, their work has the flavor of Bayesian Loss-NPL (Fong et al., 2019) where the loss function is either ALasso or SCAD. More recently, Das et al. (2019) extended the work of Minnier et al. (2011) by introducing a suitably Studentized version of modified perturbation bootstrap ALasso estimator that achieves second-order correctness in distributional consistency even when  $p_n \rightarrow \infty$ .

## Bootstrapping for LASSO

Various bootstrap techniques have been considered to construct confidence regions for standard LASSO estimators in (2.2) under different model settings, including fixed or random design, as well as homoscedastic or heteroscedastic errors  $\epsilon$ . Knight and Fu (2000) first considered the residual bootstrap under fixed design and homoscedastic error. A. Chatterjee and Lahiri (2010) presented a rigorous proof for the heuristic discussion of Knight and Fu (2000)'s Section 4 to show that the LASSO residual bootstrap samples fail to be distributionally consistent unless  $\beta_0$  is not sparse, for which Knight and Fu (2000) invoked the Skorokhod's argument. Subsequently, A. Chatterjee and Lahiri (2011a) rectified the shortcoming by proposing a modified residual bootstrap method by thresholding the Lasso estimator. Meanwhile, Camponovo (2015) proposed a modified paired-bootstrap technique and established its distributional consistency to approximate the distribution of Lasso estimators in linear models with random design and heteroscedastic errors. Recently, Das and Lahiri (2019) considered the perturbation bootstrap method for Lasso estimators under both fixed and random designs with heteroscedastic errors. Since centering on the thresholded Lasso estimator (c.f. A. Chatterjee & Lahiri, 2011a) resulted in distributional inconsistency of the naive perturbation bootstrap, Das and Lahiri (2019) proceeded with a suitably Studentized version of modified perturbation bootstrap (c.f. Das et al. (2019)) to rectify the shortcoming.

## Comparison and contribution of our paper

Interestingly, the setup of naive perturbation bootstrap in Das and Lahiri (2019) mimics the proposed random-weighting approach (2.3) in LASSO regression with weighting scheme (2.4), but there remain some differences in our approach. Das and Lahiri (2019) also considered heteroscedastic error term  $\epsilon$ , which we do not consider in this paper. Meanwhile, the weighting schemes considered in this paper are slightly more flexible, since we also consider the cases where independent random weights are also assigned on the LASSO penalty term in weighting schemes (2.5) and (2.6). The random weights in Das and Lahiri

(2019)'s perturbation bootstrap are restricted to independent draws from distribution with  $\sigma_W^2 = \mu_W^2$ , whereas we consider any positive random weights with finite fourth moment. Furthermore, our extended random-weighting framework in Section 2.3.2 attains conditional sparse normality property under growing  $p_n$  setting, whereas Das and Lahiri (2019)'s (modified) perturbation bootstrap method achieves distributional consistency under fixed dimensional ( $p_n = p$ ) setting.

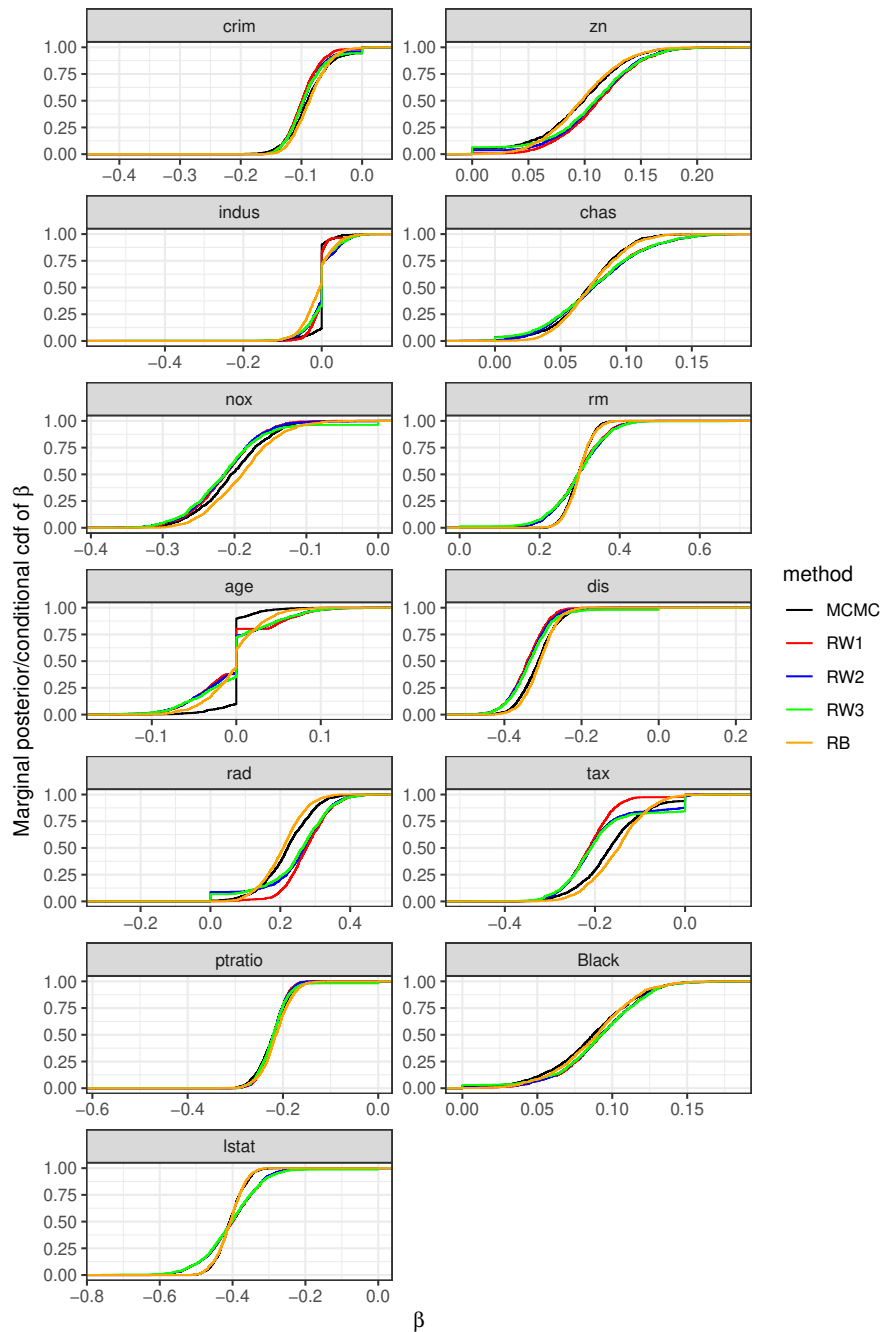


Figure 2.5: Boston Housing data example: Marginal posterior/conditional distribution plots for  $\beta = (\beta_1, \dots, \beta_{13})'$  sampled from the 5 methods – MCMC via Bayesian LASSO, the two-step random-weighting approach using weighting schemes (2.4) (2.5) and (2.6) (denoted RW1, RW2 and RW3 respectively), as well as the parametric residual bootstrap (denoted RB).

## Chapter 3

# Technical Details for Chapter 2

We present the proofs for all the theorems, proposition and corollaries in Chapter 2. Many subsequent proofs rely on this following result.

**Lemma 3.1.** *Let  $U_1, U_2, \dots$  be any i.i.d. random variables with  $\mathbb{E}(U_i) = 0$  and  $\mathbb{E}[(U_i)^2] = \sigma^2 < \infty$ . Then for any bounded sequence of real numbers  $\{k_i\}$  and for any  $\frac{1}{2} < c < 1$ ,*

$$\frac{1}{n^c} \sum_{i=1}^n k_i U_i \xrightarrow{a.s.} 0.$$

*Proof.* Since  $\{k_i\}$  are bounded,  $\exists M > 0$  such that  $|k_i| \leq M \forall i$ . Then

$$\sum_{n=1}^{\infty} \text{Var} \left( \frac{k_n U_n}{n^c} \right) = \sigma^2 \sum_{n=1}^{\infty} \frac{k_n^2}{n^{2c}} \leq \sigma^2 M^2 \sum_{n=1}^{\infty} \frac{1}{n^{2c}} < \infty.$$

By Theorem 2.5.3 of Durrett (2010), with probability one,

$$\sum_{n=1}^{\infty} \frac{k_n U_n}{n^c} < \infty.$$

Finally, apply Kronecker's Lemma to obtain the desired result.

□

**Lemma 3.2.** *Assume assumptions (2.8) and (2.9). Then,*

$$\left\| \left( C_{n(11)}^w \right)^{-1} \right\|_2 = O_p(1).$$

*Proof.* Due to assumptions (2.8) and (2.9) and that  $q$  is fixed,  $C_{n(11)}$  is invertible for all  $n$ .

We also verify the invertibility of  $C_{n(11)}^w$  by recognizing that

$$C_{n(11)}^w = \frac{1}{n} X'_{(1)} D_n X_{(1)} = \frac{1}{n} \left( D_n^{\frac{1}{2}} X_{(1)} \right)' \left( D_n^{\frac{1}{2}} X_{(1)} \right)$$

where  $D_n^{1/2} = \text{diag}(\sqrt{W_1}, \dots, \sqrt{W_n})$ , which is a full-rank square matrix. Thus,

$$\text{rank} \left( C_{n(11)}^w \right) = \text{rank} \left( D_n^{\frac{1}{2}} X_{(1)} \right) = \text{rank} \left( X_{(1)} \right) = q,$$

i.e.  $C_{n(11)}^w$  is full-rank and is invertible for every  $n$ . Next,

$$C_{n(11)}^w = C_{n(11)} + \frac{1}{n} X'_{(1)} (D_n - \mu_W I_n) X_{(1)}$$

where the Strong Law of Large Numbers ensures that

$$\frac{1}{n} X'_{(1)} (D_n - \mu_W I_n) X_{(1)} \xrightarrow{\text{a.s.}} \mathbf{0}$$

due to assumption (2.8). Since  $C_{n(11)}$  is invertible for all  $n$ , we have

$$\left\| \left( C_{n(11)}^w \right)^{-1} \right\|_2 = \left\| \left( C_{n(11)} + o(1) \right)^{-1} \right\|_2 = \mathcal{O}(1) \text{ a.s.}$$

□

In fact, if we assume  $C_{n(11)} \rightarrow C_{11}$  for some nonsingular matrix  $C_{11}$  in Lemma 3.2, then by the Strong Law of Large Numbers and Continuous Mapping Theorem,

$$\left( C_{n(11)}^w \right)^{-1} \xrightarrow{\text{a.s.}} \frac{1}{\mu_W} C_{11}^{-1}.$$

**Lemma 3.3.** *Assume assumptions (2.8) and (2.9). For any  $\frac{1}{2} < c_1 < 1$ , if  $\exists 0 \leq c_3 < 2c_1 - 1$  for which  $p_n = \mathcal{O}(n^{c_3})$ , then*

$$\left\| n^{1-c_1} \tilde{C}_n^w \right\|_2 = o_p(1).$$

*Proof.* Let

$$H = X_{(1)} C_{n(11)}^{-1} C_{n(12)} - X_{(2)}.$$

Then

$$n^{1-c_1} \tilde{C}_n^w = \frac{1}{n^{c_1}} H' (\mu_W I_n - D_n) X_{(1)} \left( C_{n(11)}^w \right)^{-1}.$$

Due to assumptions (2.8) and (2.9) and that  $q$  is fixed, every element of the matrix  $H$  is bounded. Let  $h_{ij}$  and  $x_{ij}$  be the  $(i, j)^{th}$  element of  $H$  and  $X_{(1)}$  respectively. For  $0 \leq c_3 < 2c_1 - 1$ , by Lemma 3.1,

$$\frac{1}{n^{c_1 - \frac{c_3}{2}}} \sum_{i=1}^n h_{k,i} x_{i,l} (W_i - \mu_W) \xrightarrow{\text{a.s.}} 0$$

for every  $k = 1, \dots, p_n - q$  and  $l = 1, \dots, q$ . Thus,

$$\begin{aligned} & \left\| \frac{1}{n^{c_1}} H' (\mu_W I_n - D_n) X_{(1)} \right\|_2^2 \\ & \leq \left\| \frac{1}{n^{c_1}} H' (\mu_W I_n - D_n) X_{(1)} \right\|_F^2 \\ & = \sum_{k=1}^{p_n - q} \sum_{l=1}^q \left[ \frac{1}{n^{\frac{c_3}{2}}} \times \frac{1}{n^{c_1 - \frac{c_3}{2}}} \sum_{i=1}^n h_{k,i} x_{i,l} (\mu_W - W_i) \right]^2 \\ & = \mathcal{O}(p_n) \times o(n^{-c_3}) = o(1) \quad \text{a.s..} \end{aligned}$$

Finally, by Lemma 3.2,

$$\left\| n^{1-c_1} \tilde{C}_n^w \right\|_2 \leq \left\| \frac{1}{n^{c_1}} H' (\mu_W I_n - D_n) X_{(1)} \right\|_2 \left\| \left( C_{n(11)}^w \right)^{-1} \right\|_2 = o_p(1).$$

□



**Lemma 3.4.** *Suppose that  $p_n = p$  is fixed. Assume (2.8) and (2.10). Then, as  $n \rightarrow \infty$ ,*

$$\frac{\mu_W}{n} X' D_n X \xrightarrow{a.s.} \mu_W C.$$

*Proof.* Due to assumption (2.8), the Strong Law of Large Numbers gives

$$\frac{1}{n} X' (D_n - \mu_W I_n) X = \frac{1}{n} \sum_{i=1}^n (W_i - \mu_W) \mathbf{x}_i \mathbf{x}_i' \xrightarrow{a.s.} \mathbf{0},$$

where  $\mathbf{x}_i$  is the  $i^{\text{th}}$  row of  $X$ . Then, due to assumption (2.10),

$$\frac{1}{n} X' D_n X = \frac{1}{n} X' (D_n - \mu_W I_n) X + \frac{\mu_W}{n} X' X \xrightarrow{a.s.} \mathbf{0} + \mu_W C = \mu_W C.$$

□

An immediate consequence of Lemma 3.4 is that when  $p$  is fixed,

$$C_{n(ij)}^w \xrightarrow{a.s.} \mu_W C_{ij} \quad \forall i, j = 1, 2.$$

We remind readers that in this paper, we consider a common probability space  $P = P_D \times P_W$ , which correspond to the two sources of randomness  $(\epsilon, \mathbf{W})$ . Note that the product probability space highlights the fact that the random weights  $\mathbf{W}$  are drawn independently from the data  $D$ . The rest of the proofs deals with convergence of conditional probabilities/distributions (given data, i.e. given  $\mathcal{F}_n$ ) for expressions containing  $\epsilon$ , where the convergence takes place almost surely under  $P_D$  (i.e. for almost every data set). See Mason and Newton (1992) for relevant background.

**Lemma 3.5.** *Assume (2.7). Then*

$$\frac{\epsilon' D_n \epsilon}{n} \xrightarrow{c.p.} \mu_W \sigma_\epsilon^2 \quad a.s. P_D.$$

*Proof.* Clearly,

$$\frac{1}{n} \sum_{i=1}^n \epsilon_i^2 \rightarrow \sigma_\epsilon^2 \quad a.s. P_D.$$

Due to assumption (2.7),

$$\frac{1}{n} \sum_{i=1}^n \epsilon_i^4 = \mathcal{O}(1) \quad a.s. P_D,$$

which leads to

$$\frac{1}{n^2} \sum_{i=1}^n \mathbb{E}(\epsilon_i^4 W_i^2 | \mathcal{F}_n) = \frac{1}{n^2} \sum_{i=1}^n \epsilon_i^4 \mathbb{E}(W_i^2) = \frac{\sigma_W^2 + \mu_W^2}{n} \left( \frac{1}{n} \sum_{i=1}^n \epsilon_i^4 \right) \rightarrow 0 \quad a.s. P_D.$$

Hence, by the Weak Law of Large Numbers (e.g., Theorem 1.14(ii) of Shao (2003)),

$$\frac{1}{n} \epsilon' (D_n - \mu_W I_n) \epsilon = \frac{1}{n} \sum_{i=1}^n \epsilon_i^2 (W_i - \mu_W) \xrightarrow{c.p.} 0 \quad a.s. P_D,$$

and thus,

$$\frac{\epsilon' D_n \epsilon}{n} = \frac{1}{n} \sum_{i=1}^n \epsilon_i^2 (W_i - \mu_W) + \frac{\mu_W}{n} \sum_{i=1}^n \epsilon_i^2 \xrightarrow{c.p.} 0 + \mu_W \sigma_\epsilon^2 = \mu_W \sigma_\epsilon^2 \quad a.s. P_D.$$

□

**Lemma 3.6.** Assume (2.7), (2.8) and (2.9). Then for any  $c > 0$ ,

$$\frac{1}{n^c} \mathbf{Z}_{n(1)}^w = o_p(1) \quad a.s. P_D.$$

*Proof.* Let  $x_{ij}$  be the  $(i, j)^{th}$  element of  $X_{(1)}$ . Then, we can rewrite

$$\begin{aligned} \left( \frac{1}{n^c} \left\| \mathbf{Z}_{n(1)}^w \right\|_2 \right)^2 &= \frac{1}{n^{2c}} \sum_{j=1}^q \left( \frac{1}{\sqrt{n}} \sum_{i=1}^n \epsilon_i x_{ji} (W_i - \mu_W) + \frac{\mu_W}{\sqrt{n}} \sum_{i=1}^n \epsilon_i x_{ji} \right)^2 \\ &= \sum_{j=1}^q \left( \frac{1}{n^{\frac{1}{2}+c}} \sum_{i=1}^n \epsilon_i x_{ji} (W_i - \mu_W) + \frac{\mu_W}{n^{\frac{1}{2}+c}} \sum_{i=1}^n \epsilon_i x_{ji} \right)^2, \end{aligned}$$

where we note that

$$\mathbb{E} \left( \sum_{i=1}^n \epsilon_i x_{ji} W_i \middle| \mathcal{F}_n \right) = \sum_{i=1}^n \epsilon_i x_{ji} \mathbb{E}(W_i) = \mu_W \sum_{i=1}^n \epsilon_i x_{ji},$$

and

$$\text{Var} \left( \sum_{i=1}^n \epsilon_i x_{ji} W_i \middle| \mathcal{F}_n \right) = \sum_{i=1}^n \epsilon_i^2 x_{ji}^2 \text{Var}(W_i) = \sigma_W^2 \sum_{i=1}^n \epsilon_i^2 x_{ji}^2.$$

Now, due to assumption (2.8),

$$\frac{1}{n} \sum_{i=1}^n \epsilon_i^2 x_{ji}^2 = \mathcal{O}(1) \text{ a.s. } P_D \implies \sum_{i=1}^n \epsilon_i^2 x_{ji}^2 = \mathcal{O}(n) \text{ a.s. } P_D,$$

and coupled with assumption (2.7),

$$\frac{1}{n} \sum_{i=1}^n \epsilon_i^4 x_{ji}^4 = \mathcal{O}(1) \text{ a.s. } P_D \implies \sum_{i=1}^n \epsilon_i^4 x_{ji}^4 = \mathcal{O}(n) \text{ a.s. } P_D.$$

Thus, by using assumptions (2.7) and (2.8) and that  $F_W$  has finite fourth moment, the Liapounov's sufficient condition is satisfied

$$\begin{aligned} & \left[ \sum_{i=1}^n \epsilon_i^2 x_{ji}^2 \text{Var}(W_i) \right]^{-2} \left[ \sum_{i=1}^n \epsilon_i^4 x_{ji}^4 \mathbb{E}(W_i - \mu_W)^4 \right] \\ &= \mathcal{O}(n^{-2}) \times \mathcal{O}(n) = \mathcal{O}(n^{-1}) \text{ a.s. } P_D, \end{aligned}$$

in order to deploy the Lindeberg's Central Limit Theorem

$$\frac{\sum_{i=1}^n \epsilon_i x_{ji} (W_i - \mu_W)}{\sqrt{\sigma_W^2 \sum_{i=1}^n \epsilon_i^2 x_{ji}^2}} \xrightarrow{\text{c.d.}} N(0, 1) \text{ a.s. } P_D.$$

Subsequently, for all  $j = 1, \dots, q$ ,

$$\frac{1}{\sqrt{n}} \sum_{i=1}^n \epsilon_i x_{ji} (W_i - \mu_W)$$

$$\begin{aligned}
&= \sqrt{\frac{\sigma_W^2}{n} \sum_{i=1}^n \epsilon_i^2 x_{ji}^2} \times \frac{\sum_{i=1}^n \epsilon_i x_{ji} (W_i - \mu_W)}{\sqrt{\sigma_W^2 \sum_{i=1}^n \epsilon_i^2 x_{ji}^2}} \\
&= \mathcal{O}_p(1) \quad a.s. P_D,
\end{aligned}$$

and hence,

$$\frac{1}{n^{\frac{1}{2}+c}} \sum_{i=1}^n \epsilon_i x_{ji} (W_i - \mu_W) = o_p(1) \quad a.s. P_D.$$

Finally, by assumption (2.8) and Lemma 3.1,

$$\frac{\mu_W}{n^{\frac{1}{2}+c}} \sum_{i=1}^n \epsilon_i x_{ji} \rightarrow 0 \quad a.s. P_D$$

for all  $j = 1, \dots, q$ . Since  $q$  is fixed,

$$\left( \frac{1}{n^c} \left\| \mathbf{Z}_{n(1)}^w \right\|_2 \right)^2 = o_p(1) \quad a.s. P_D,$$

and the result follows. □

If we assume that  $C_{n(11)} \rightarrow C_{11}$  for some nonsingular matrix  $C_{11}$  in Lemma 3.6, notations could be simplified in the preceding proof by using Cramer-Wold device. We point out to readers that the  $C_{n(11)} \rightarrow C_{11}$  assumption is required in Theorem 2.5 but not in Theorem 2.2. The following proof contains some interim results that will be utilized in the proof of Theorem 2.5.

Specifically, let  $\mathbf{x}_{i(1)}$  be the  $i^{th}$  row of  $X_{(1)}$ . Then, for every  $\mathbf{z} \in \mathbb{R}^q$ ,

$$\begin{aligned}
&\mathbf{z}' \left[ \frac{1}{\sqrt{n}} X'_{(1)} (D_n - \mu_W I_n) \boldsymbol{\epsilon} \right] \\
&= \frac{1}{\sqrt{n}} \sum_{i=1}^n \epsilon_i (W_i - \mu_W) \mathbf{z}' \mathbf{x}_{i(1)} \\
&= \sqrt{\frac{\sigma_W^2}{n} \sum_{i=1}^n \epsilon_i^2 (\mathbf{z}' \mathbf{x}_{i(1)})^2} \times \frac{\sum_{i=1}^n \epsilon_i (W_i - \mu_W) \mathbf{z}' \mathbf{x}_{i(1)}}{\sqrt{\sigma_W^2 \sum_{i=1}^n \epsilon_i^2 (\mathbf{z}' \mathbf{x}_{i(1)})^2}},
\end{aligned}$$

where we note that

$$\mathbb{E} \left( \sum_{i=1}^n \epsilon_i W_i (\mathbf{z}' \mathbf{x}_{i(1)}) \middle| \mathcal{F}_n \right) = \sum_{i=1}^n \epsilon_i (\mathbf{z}' \mathbf{x}_{i(1)}) \mathbb{E}(W_i) = \mu_W \sum_{i=1}^n \epsilon_i (\mathbf{z}' \mathbf{x}_{i(1)}),$$

and

$$\text{Var} \left( \sum_{i=1}^n \epsilon_i W_i (\mathbf{z}' \mathbf{x}_{i(1)}) \middle| \mathcal{F}_n \right) = \sum_{i=1}^n \epsilon_i^2 (\mathbf{z}' \mathbf{x}_{i(1)})^2 \text{Var}(W_i) = \sigma_W^2 \sum_{i=1}^n \epsilon_i^2 (\mathbf{z}' \mathbf{x}_{i(1)})^2.$$

Now,

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n \epsilon_i^2 (\mathbf{z}' \mathbf{x}_{i(1)})^2 &= \mathbf{z}' \left( \frac{1}{n} \sum_{i=1}^n \epsilon_i^2 \mathbf{x}_{i(1)} \mathbf{x}'_{i(1)} \right) \mathbf{z} \\ &= \mathbf{z}' \left( \sigma_\epsilon^2 C_{n(11)} + \frac{1}{n} \sum_{i=1}^n (\epsilon_i^2 - \sigma_\epsilon^2) \mathbf{x}_{i(1)} \mathbf{x}'_{i(1)} \right) \mathbf{z} \\ &\rightarrow \mathbf{z}' (\sigma_\epsilon^2 C_{11}) \mathbf{z} \quad a.s. P_D \end{aligned}$$

due to assumption (2.8) and the Strong Law of Large Numbers. Thus,

$$\sum_{i=1}^n \epsilon_i^2 (\mathbf{z}' \mathbf{x}_{i(1)})^2 = \mathcal{O}(n) \quad a.s. P_D.$$

In addition, by assumptions (2.7) and (2.8),

$$\frac{1}{n} \sum_{i=1}^n \epsilon_i^4 (\mathbf{z}' \mathbf{x}_{i(1)})^4 \leq (qM_1 \|\mathbf{z}\|_2)^4 \left( \frac{1}{n} \sum_{i=1}^n \epsilon_i^4 \right) = \mathcal{O}(1) \quad a.s. P_D,$$

which implies

$$\sum_{i=1}^n \epsilon_i^4 (\mathbf{z}' \mathbf{x}_{i(1)})^4 = \mathcal{O}(n) \quad a.s. P_D.$$

Therefore, by using assumptions (2.7) and (2.8) and that  $F_W$  has finite fourth moment, we

could verify the Liapounov's sufficient condition

$$\begin{aligned} & \left[ \sum_{i=1}^n \epsilon_i^2 (\mathbf{z}' \mathbf{x}_{i(1)})^2 \text{Var}(W_i) \right]^{-2} \left[ \sum_{i=1}^n \epsilon_i^4 (\mathbf{z}' \mathbf{x}_{i(1)})^4 \mathbb{E}(W_i - \mu_W)^4 \right] \\ &= \mathcal{O}(n^{-2}) \times \mathcal{O}(n) = \mathcal{O}(n^{-1}) \quad a.s. P_D, \end{aligned}$$

in order to deploy the Lindeberg's Central Limit Theorem

$$\frac{\sum_{i=1}^n \epsilon_i (W_i - \mu_W) \mathbf{z}' \mathbf{x}_{i(1)}}{\sqrt{\sigma_W^2 \sum_{i=1}^n \epsilon_i^2 (\mathbf{z}' \mathbf{x}_{i(1)})^2}} \xrightarrow{\text{c.d.}} N(0, 1) \quad a.s. P_D.$$

Then, by Slutsky's Theorem, for every  $\mathbf{z} \in \mathbb{R}^q$ ,

$$\mathbf{z}' \left[ \frac{1}{\sqrt{n}} X'_{(1)} (D_n - \mu_W I_n) \boldsymbol{\epsilon} \right] \xrightarrow{\text{c.d.}} N(0, \mathbf{z}' (\sigma_W^2 \sigma_\epsilon^2 C_{11}) \mathbf{z}).$$

and by Cramer-Wold device,

$$\frac{1}{\sqrt{n}} X'_{(1)} (D_n - \mu_W I_n) \boldsymbol{\epsilon} \xrightarrow{\text{c.d.}} N_q(\mathbf{0}, \sigma_W^2 \sigma_\epsilon^2 C_{11}),$$

Since assumption (2.8) and Lemma 3.1 ensure that for any  $c > 0$ ,

$$\frac{1}{n^{\frac{1}{2}+c}} X'_{(1)} \boldsymbol{\epsilon} \rightarrow \mathbf{0} \quad a.s. P_D,$$

we finally have

$$\frac{1}{n^c} \mathbf{Z}_{n(1)}^w = \frac{1}{n^c} \left[ \frac{1}{\sqrt{n}} X'_{(1)} (D_n - \mu_W I_n) \boldsymbol{\epsilon} \right] + \frac{\mu_W}{n^{\frac{1}{2}+c}} X'_{(1)} \boldsymbol{\epsilon} = o_p(1) \quad a.s. P_D.$$

**Lemma 3.7.** Assume (2.7), (2.8) and (2.9).

(a) If there exists  $\frac{1}{2} < c_1 < c_2 < 1.5 - c_1$  and  $0 \leq c_3 < 2(c_2 - c_1)$  for which  $p_n = \mathcal{O}(n^{c_3})$ , then

$$\frac{1}{n^{c_2 - \frac{1}{2}}} \left\| \mathbf{Z}_{n(3)}^w \right\|_2 = o_p(1) \quad a.s. P_D.$$

(b) If there exists  $\frac{1}{2} < c_1 < c_2 < 1.5 - c_1$  and  $0 \leq c_3 < \frac{2}{3}(c_2 - c_1)$  for which  $p_n = \mathcal{O}(n^{c_3})$ , then

$$\frac{p_n - q}{n^{c_2 - \frac{1}{2}}} \left\| \mathbf{Z}_{n(3)}^w \right\|_2 = o_p(1) \quad a.s. P_D.$$

*Proof.* Let

$$H = X_{(1)} C_{n(11)}^{-1} C_{n(12)} - X_{(2)}.$$

Then

$$\mathbf{Z}_{n(3)}^w = \frac{1}{\sqrt{n}} H' D_n \epsilon.$$

Due to assumptions (2.8) and (2.9) and that  $q$  is fixed, every element of the matrix  $H$  is bounded. Let  $h_{ij}$  be the  $(i, j)^{th}$  element of  $H$ . Then, for all  $j = 1, \dots, p_n - q$ ,

$$\frac{1}{n} \sum_{i=1}^n h_{ji}^2 \epsilon_i^2 = O(1) \quad a.s. P_D \implies \sum_{i=1}^n h_{ji}^2 \epsilon_i^2 = O(n) \quad a.s. P_D,$$

and

$$\frac{1}{n} \sum_{i=1}^n h_{ji}^4 \epsilon_i^4 = O(1) \quad a.s. P_D \implies \sum_{i=1}^n h_{ji}^4 \epsilon_i^4 = O(n) \quad a.s. P_D$$

due to assumption (2.7). Next, we note that

$$\mathbb{E} \left( \sum_{i=1}^n h_{ji} \epsilon_i W_i \middle| \mathcal{F}_n \right) = \sum_{i=1}^n h_{ji} \epsilon_i \mathbb{E}(W_i) = \mu_W \sum_{i=1}^n h_{ji} \epsilon_i,$$

and

$$Var \left( \sum_{i=1}^n h_{ji} \epsilon_i W_i \middle| \mathcal{F}_n \right) = \sum_{i=1}^n h_{ji}^2 \epsilon_i^2 Var(W_i) = \sigma_W^2 \sum_{i=1}^n h_{ji}^2 \epsilon_i^2.$$

By using assumptions (2.7) and (2.8) and that  $F_W$  has finite fourth moment, we could verify the Liapounov's sufficient condition

$$\begin{aligned} & \left[ \sum_{i=1}^n h_{ji}^2 \epsilon_i^2 Var(W_i) \right]^{-2} \left[ \sum_{i=1}^n h_{ji}^4 \epsilon_i^4 \mathbb{E}(W_i - \mu_W)^4 \right] \\ &= \mathcal{O}(n^{-2}) \times \mathcal{O}(n) = \mathcal{O}(n^{-1}) \quad a.s. P_D, \end{aligned}$$

in order to deploy the Lindeberg's Central Limit Theorem

$$\frac{\sum_{i=1}^n h_{ji}\epsilon_i(W_i - \mu_W)}{\sqrt{\sigma_W^2 \sum_{i=1}^n h_{ji}^2 \epsilon_i^2}} \xrightarrow{\text{c.d.}} N(0, 1) \quad \text{a.s. } P_D.$$

Thus, for all  $j = 1, \dots, p_n - q$ ,

$$\begin{aligned} & \frac{1}{\sqrt{n}} \sum_{i=1}^n h_{ji}\epsilon_i(W_i - \mu_W) \\ &= \sqrt{\frac{\sigma_W^2}{n} \sum_{i=1}^n h_{ji}^2 \epsilon_i^2} \times \frac{\sum_{i=1}^n h_{ji}\epsilon_i(W_i - \mu_W)}{\sqrt{\sigma_W^2 \sum_{i=1}^n h_{ji}^2 \epsilon_i^2}} \\ &= O_p(1) \quad \text{a.s. } P_D, \end{aligned}$$

which leads to

$$\frac{1}{n^{c_1}} \sum_{i=1}^n h_{ji}\epsilon_i(W_i - \mu_W) = o_p(1) \quad \text{a.s. } P_D,$$

whereas Lemma 3.1 ensures that

$$\frac{1}{n^{c_1}} \sum_{i=1}^n h_{ji}\epsilon_i \rightarrow 0 \quad \text{a.s. } P_D.$$

Therefore, for part (a) of Lemma 3.7,

$$\begin{aligned} & \left( \frac{1}{n^{c_2 - \frac{1}{2}}} \left\| \mathbf{Z}_{n(3)}^w \right\|_2 \right)^2 \\ &= \frac{1}{n^{2c_2 - 1}} \left\| \mathbf{Z}_{n(3)}^w \right\|_2^2 \\ &= \frac{1}{n^{2c_2 - 1}} \sum_{j=1}^{p_n - q} \left( \frac{1}{\sqrt{n}} \sum_{i=1}^n h_{ji}\epsilon_i(W_i - \mu_W) + \frac{1}{\sqrt{n}} \sum_{i=1}^n h_{ji}\epsilon_i \right)^2 \\ &= \frac{n^{2c_1 - 1}}{n^{2c_2 - 1}} \sum_{j=1}^{p_n - q} \left( \frac{1}{n^{c_1}} \sum_{i=1}^n h_{ji}\epsilon_i(W_i - \mu_W) + \frac{1}{n^{c_1}} \sum_{i=1}^n h_{ji}\epsilon_i \right)^2 \\ &= \mathcal{O} \left( n^{2(c_1 - c_2)} \right) \times o_p(n^{c_3}) \quad \text{a.s. } P_D \\ &= o_p(1) \quad \text{a.s. } P_D \end{aligned}$$



since  $c_3 < 2(c_2 - c_1)$ .

For part (b) of Lemma 3.7,

$$\begin{aligned} & \left( \frac{p_n - q}{n^{c_2 - \frac{1}{2}}} \left\| \mathbf{Z}_{n(3)}^w \right\|_2 \right)^2 \\ &= \mathcal{O} \left( n^{2(c_1 - c_2 + c_3)} \right) \times o_p(n^{c_3}) \quad a.s. P_D \\ &= o_p(1) \quad a.s. P_D \end{aligned}$$

since  $c_3 < \frac{2}{3}(c_2 - c_1)$ . □

**Lemma 3.8.** *Assume (2.8) and that  $p_n = p$  is fixed. Then*

$$\frac{1}{n} X' D_n \boldsymbol{\epsilon} \xrightarrow{c.p.} \mathbf{0} \quad a.s. P_D.$$

*Proof.* Let  $\mathbf{x}_i$  and  $x_{ij}$  be the  $i^{th}$  row and  $(i, j)^{th}$  element of  $X$  respectively. Due to assumption (2.8),

$$\frac{1}{n} X' \boldsymbol{\epsilon} \rightarrow \mathbf{0} \quad a.s. P_D,$$

and for all  $j = 1, \dots, p$ ,

$$\begin{aligned} & \frac{1}{n^2} \sum_{i=1}^n \mathbb{E} \left( x_{ji}^2 \epsilon_i^2 W_i^2 \mid \mathcal{F}_n \right) \\ &= \frac{1}{n^2} \sum_{i=1}^n x_{ji}^2 \epsilon_i^2 \mathbb{E}(W_i^2) \\ &\leq \frac{M_1^2 (\sigma_W^2 + \mu_W^2)}{n} \left( \frac{1}{n} \sum_{i=1}^n \epsilon_i^2 \right) \\ &\rightarrow 0 \quad a.s. P_D. \end{aligned}$$

Hence, by the Weak Law of Large Numbers (e.g., Theorem 1.14(ii) of Shao (2003)),

$$\frac{1}{n} X' (D_n - \mu_W I_n) \boldsymbol{\epsilon} = \frac{1}{n} \sum_{i=1}^n \epsilon_i (W_i - \mu_W) \mathbf{x}_i \xrightarrow{c.p.} \mathbf{0} \quad a.s. P_D.$$

Finally,

$$\frac{X'D_n\epsilon}{n} = \frac{1}{n}X'(D_n - \mu_W I_n)\epsilon + \frac{\mu_W}{n}X'\epsilon \xrightarrow{\text{c.p.}} \mathbf{0} \quad \text{a.s. } P_D.$$

□

**Lemma 3.9.** *Suppose that  $p_n = p$  is fixed. Assume (2.7), (2.8), (2.10), and*

$$\frac{1}{\sqrt{n}}X'e_n \rightarrow \mathbf{0} \quad \text{a.s. } P_D,$$

where  $e_n$  is the residual of the strongly consistent estimator  $\widehat{\beta}_n^{\text{SC}}$  of the linear model (2.1). Then,

$$\frac{1}{\sqrt{n}}X'D_n e_n \xrightarrow{\text{c.d.}} N_p(\mathbf{0}, \sigma_W^2 \sigma_\epsilon^2 C) \quad \text{a.s. } P_D.$$

*Proof.* Due to assumption (2.10),

$$\frac{\sigma_\epsilon^2}{n}X'X \rightarrow \sigma_\epsilon^2 C.$$

Since  $\widehat{\beta}_n^{\text{SC}}$  is a strongly consistent estimator of  $\beta$  in (2.1), we have

$$\left(\widehat{\beta}_n^{\text{SC}} - \beta_0\right) \rightarrow \mathbf{0} \quad \text{a.s. } P_D.$$

Let  $\mathbf{x}_i$  be the  $i^{\text{th}}$  row of  $X$ , and let  $e_i$  be the  $i^{\text{th}}$  element of  $e_n$ . Due to assumption (2.8) and Lemma 3.1 and the fact that  $\widehat{\beta}_n^{\text{SC}}$  is strongly consistent,

$$\begin{aligned} & \frac{1}{n} \sum_{i=1}^n (e_i^2 - \sigma_\epsilon^2) \mathbf{x}_i \mathbf{x}_i' \\ &= \frac{1}{n} \sum_{i=1}^n \left( \left[ \mathbf{x}_i' \left( \beta_0 - \widehat{\beta}_n^{\text{SC}} \right) + \epsilon_i \right]^2 - \sigma_\epsilon^2 \right) \mathbf{x}_i \mathbf{x}_i' \\ &= \frac{1}{n} \sum_{i=1}^n (\epsilon_i^2 - \sigma_\epsilon^2) \mathbf{x}_i \mathbf{x}_i' \\ & \quad + \frac{2}{n} \sum_{i=1}^n \epsilon_i \left[ \mathbf{x}_i' \left( \beta_0 - \widehat{\beta}_n^{\text{SC}} \right) \right] \mathbf{x}_i \mathbf{x}_i' \end{aligned}$$

$$\begin{aligned}
& + \frac{1}{n} \sum_{i=1}^n \left[ \mathbf{x}'_i (\boldsymbol{\beta}_0 - \widehat{\boldsymbol{\beta}}_n^{\text{SC}}) \right]^2 \mathbf{x}_i \mathbf{x}'_i \\
& \rightarrow \mathbf{0} \quad \text{a.s. } P_D,
\end{aligned}$$

which leads to

$$\frac{1}{n} \sum_{i=1}^n e_i^2 \mathbf{x}_i \mathbf{x}'_i = \frac{1}{n} \sum_{i=1}^n (e_i^2 - \sigma_\epsilon^2) \mathbf{x}_i \mathbf{x}'_i + \frac{\sigma_\epsilon^2}{n} X'X \rightarrow \sigma_\epsilon^2 C \quad \text{a.s. } P_D. \quad (3.1)$$

Now for every  $\mathbf{z} \in \mathbb{R}^p$ , consider

$$\begin{aligned}
& \mathbf{z}' \left[ \frac{1}{\sqrt{n}} X' (D_n - \mu_W I_n) \mathbf{e}_n \right] \\
& = \frac{1}{\sqrt{n}} \sum_{i=1}^n e_i (W_i - \mu_W) (\mathbf{z}' \mathbf{x}_i) \\
& = \sqrt{\frac{\sigma_W^2}{n} \sum_{i=1}^n e_i^2 (\mathbf{z}' \mathbf{x}_i)^2} \times \frac{\sum_{i=1}^n e_i (W_i - \mu_W) (\mathbf{z}' \mathbf{x}_i)}{\sqrt{\sigma_W^2 \sum_{i=1}^n e_i^2 (\mathbf{z}' \mathbf{x}_i)^2}}.
\end{aligned}$$

We verify that

$$\mathbb{E} \left\{ \sum_{i=1}^n e_i W_i (\mathbf{z}' \mathbf{x}_i) \middle| \mathcal{F}_n \right\} = \mu_W \sum_{i=1}^n e_i (\mathbf{z}' \mathbf{x}_i),$$

and

$$\text{Var} \left( \sum_{i=1}^n e_i W_i (\mathbf{z}' \mathbf{x}_i) \middle| \mathcal{F}_n \right) = \sigma_W^2 \sum_{i=1}^n e_i^2 (\mathbf{z}' \mathbf{x}_i)^2.$$

From (3.1), we have

$$\frac{1}{n} \sum_{i=1}^n e_i^2 (\mathbf{z}' \mathbf{x}_i)^2 = \mathbf{z}' \left( \frac{1}{n} \sum_{i=1}^n e_i^2 \mathbf{x}_i \mathbf{x}'_i \right) \mathbf{z} \rightarrow \mathbf{z}' (\sigma_\epsilon^2 C) \mathbf{z} \quad \text{a.s. } P_D,$$

and thus

$$\sum_{i=1}^n e_i^2 (\mathbf{z}' \mathbf{x}_i)^2 = \mathcal{O}(n) \quad \text{a.s. } P_D.$$

Due to assumptions (2.7) and (2.8) and the fact that  $\widehat{\beta}_n^{\text{SC}}$  is strongly consistent,

$$\begin{aligned}
& \frac{1}{n} \sum_{i=1}^n e_i^4(\mathbf{z}'\mathbf{x}_i)^4 \\
& \leq (pM_1\|\mathbf{z}\|_2)^4 \times \left( \frac{1}{n} \sum_{i=1}^n e_i^4 \right) \\
& = (pM_1\|\mathbf{z}\|_2)^4 \times \left( \frac{1}{n} \sum_{i=1}^n [\epsilon_i - \mathbf{x}'_i (\widehat{\beta}_n^{\text{SC}} - \beta_0)]^4 \right) \\
& \leq (pM_1\|\mathbf{z}\|_2)^4 \times \left[ \frac{1}{n} \sum_{i=1}^n (|\epsilon_i| + pM_1 \|\widehat{\beta}_n^{\text{SC}} - \beta_0\|_2)^4 \right] \\
& = \mathcal{O}(1) \quad a.s. P_D,
\end{aligned}$$

and thus

$$\sum_{i=1}^n e_i^4(\mathbf{z}'\mathbf{x}_i)^4 = \mathcal{O}(n) \quad a.s. P_D.$$

Since the i.i.d. random weights are drawn from  $F_W$  which has finite fourth moment, the Liapounov's sufficient condition is satisfied

$$\begin{aligned}
& \left[ \sum_{i=1}^n e_i^2(\mathbf{z}'\mathbf{x}_i)^2 \text{Var}(W_i) \right]^{-2} \left[ \sum_{i=1}^n e_i^4(\mathbf{z}'\mathbf{x}_i)^4 \mathbb{E}(W_i - \mu_W)^4 \right] \\
& = \mathcal{O}(n^{-2}) \times \mathcal{O}(n) \\
& = \mathcal{O}(n^{-1}) \quad a.s. P_D
\end{aligned}$$

in order to deploy the Lindeberg's Central Limit Theorem

$$\frac{\sum_{i=1}^n e_i(W_i - \mu_W)(\mathbf{z}'\mathbf{x}_i)}{\sqrt{\sigma_W^2 \sum_{i=1}^n e_i^2(\mathbf{z}'\mathbf{x}_i)^2}} \xrightarrow{\text{c.d.}} N(0, 1) \quad a.s. P_D.$$

By Slutsky's Theorem, for every  $\mathbf{z} \in \mathbb{R}^p$ ,

$$\mathbf{z}' \left[ \frac{1}{\sqrt{n}} X'(D_n - \mu_W I_n) \mathbf{e}_n \right] \xrightarrow{\text{c.d.}} N(0, \mathbf{z}' (\sigma_W^2 \sigma_\epsilon^2 C) \mathbf{z}) \quad a.s. P_D,$$

and by Cramer-Wold device,

$$\frac{1}{\sqrt{n}}X'(D_n - \mu_W I_n)\mathbf{e}_n \xrightarrow{\text{c.d.}} N_p(\mathbf{0}, \sigma_W^2 \sigma_\epsilon^2 C) \quad \text{a.s. } P_D.$$

Finally,

$$\frac{1}{\sqrt{n}}X'D_n\mathbf{e}_n \xrightarrow{\text{c.d.}} N_p(\mathbf{0}, \sigma_W^2 \sigma_\epsilon^2 C) \quad \text{a.s. } P_D$$

since by assumption (2.12),

$$\frac{\mu_W}{\sqrt{n}}X'\mathbf{e}_n \rightarrow \mathbf{0} \quad \text{a.s. } P_D.$$

□

We are now ready to prove the main results presented in the main text. The proof of Proposition 2.1 is similar to that of Proposition 1 of Zhao and Yu (2006).

*Proof of Proposition 2.1.* First, we note that since  $\text{rank}(X) = p_n$ , where  $p_n \leq n$ , the solution to (2.3) is unique by Osborne et al. (2000) and R. J. Tibshirani (2013). We begin with weighting scheme (2.6). Results for the other two simpler weighting schemes could then be easily inferred.

$$\begin{aligned} \widehat{\beta}_n^w &= \arg \min_{\beta} \left\{ \frac{1}{n}(Y - X\beta)'D_n(Y - X\beta) + \frac{\lambda_n}{n} \sum_{j=1}^{p_n} W_{0,j}|\beta_j| \right\} \\ &= \arg \min_{\beta} \left\{ \frac{1}{n}[\epsilon - X(\beta - \beta_0)]'D_n[\epsilon - X(\beta - \beta_0)] \right. \\ &\quad \left. + \frac{\lambda_n}{n} \sum_{j=1}^{p_n} W_{0,j}|\beta_{0,j} + \beta_j - \beta_{0,j}| \right\}. \end{aligned}$$

Therefore,

$$\begin{aligned} &(\widehat{\beta}_n^w - \beta_0) \\ &= \arg \min_{\mathbf{u}_n} \left\{ \frac{1}{n}(\epsilon - X\mathbf{u}_n)'D_n(\epsilon - X\mathbf{u}_n) + \frac{\lambda_n}{n} \sum_{j=1}^{p_n} W_{0,j}|\beta_{0,j} + u_{n,j}| \right\} \end{aligned}$$

$$= \arg \min_{\mathbf{u}_n} \left\{ \mathbf{u}'_n \left( \frac{X' D_n X}{n} \right) \mathbf{u}_n - 2 \mathbf{u}'_n \left( \frac{X' D_n \boldsymbol{\epsilon}}{n} \right) + \frac{\boldsymbol{\epsilon}' D_n \boldsymbol{\epsilon}}{n} + \frac{\lambda_n}{n} \sum_{j=1}^{p_n} W_{0,j} |\beta_{0,j} + u_{n,j}| \right\}.$$

The term  $(\boldsymbol{\epsilon}' D_n \boldsymbol{\epsilon})/n$  could be dropped since for every  $n$ , it does not contain  $\mathbf{u}_n$  and Lemma 3.5 ensures that it converges in conditional probability to a finite limit. Differentiating the first two terms with respect to  $\mathbf{u}_n$  yields

$$\frac{1}{n} \{2X' D_n X \mathbf{u}_n - 2X' D_n \boldsymbol{\epsilon}\} = \frac{1}{n} \{2\sqrt{n} [C_n^w (\sqrt{n} \mathbf{u}_n) - \mathbf{Z}_n^w]\}.$$

For  $j = 1, \dots, p_n$ , considering sub-differentials of the penalty term with respect to  $u_{n,j}$  yields

$$\begin{aligned} & \begin{cases} \frac{\lambda_n}{n} W_{0,j} \times \text{sgn}(\beta_{0,j} + u_{n,j}) & \text{for } \beta_{0,j} + u_{n,j} \neq 0 \\ \frac{\lambda_n}{n} W_{0,j} \times [-1, 1] & \text{for } \beta_{0,j} + u_{n,j} = 0 \end{cases} \\ &= \begin{cases} \frac{\lambda_n}{n} W_{0,j} \times \text{sgn}(\widehat{\beta}_{n,j}^w) & \text{for } \widehat{\beta}_{n,j}^w \neq 0 \\ \frac{\lambda_n}{n} W_{0,j} \times [-1, 1] & \text{for } \widehat{\beta}_{n,j}^w = 0 \end{cases} \end{aligned}$$

Note that  $\widehat{\boldsymbol{\beta}}_n^w = \widehat{\mathbf{u}}_n + \boldsymbol{\beta}_0$ , which can be partitioned into

$$\widehat{\boldsymbol{\beta}}_n^w = \begin{bmatrix} \widehat{\boldsymbol{\beta}}_{n(1^*)}^w \\ \widehat{\boldsymbol{\beta}}_{n(2^*)}^w \end{bmatrix},$$

where  $\widehat{\boldsymbol{\beta}}_{n(1^*)}^w$  consists of non-zero elements of  $\widehat{\boldsymbol{\beta}}_n^w$ , and  $\widehat{\boldsymbol{\beta}}_{n(2^*)}^w = \mathbf{0}$ . The asterisk here is to distinguish the partition of random-weighting samples  $\widehat{\boldsymbol{\beta}}_n^w$  from the true partition of  $\boldsymbol{\beta}_0$ . It follows that

$$2\sqrt{n} [C_n^w (\sqrt{n} \widehat{\mathbf{u}}_n) - \mathbf{Z}_n^w]$$

$$= 2\sqrt{n} \left\{ \begin{bmatrix} C_{n(11^*)}^w & C_{n(12^*)}^w \\ C_{n(21^*)}^w & C_{n(22^*)}^w \end{bmatrix} \times \sqrt{n} \begin{bmatrix} \hat{\mathbf{u}}_{n(1^*)} \\ \hat{\mathbf{u}}_{n(2^*)} \end{bmatrix} - \begin{bmatrix} \mathbf{Z}_{n(1^*)}^w \\ \mathbf{Z}_{n(2^*)}^w \end{bmatrix} \right\}.$$

Note that  $\hat{\mathbf{u}}_{n(2^*)}$  does not necessarily equal to  $\mathbf{0}$  unless the partition of the random-weighting samples  $\hat{\beta}_n^w$  coincides with the true partition of  $\beta_0$ . As a consequence of the Karush-Kuhn-Tucker (KKT) conditions, we have

$$C_{n(11^*)}^w [\sqrt{n}\hat{\mathbf{u}}_{n(1^*)}] + C_{n(12^*)}^w [\sqrt{n}\hat{\mathbf{u}}_{n(2^*)}] - \mathbf{Z}_{n(1^*)}^w = -\frac{\lambda_n}{2\sqrt{n}} \mathbf{W}_{0(1)} \circ \text{sgn}(\hat{\beta}_{n(1^*)}^w) \quad (3.2)$$

and

$$\left| C_{n(21^*)}^w [\sqrt{n}\hat{\mathbf{u}}_{n(1^*)}] + C_{n(22^*)}^w [\sqrt{n}\hat{\mathbf{u}}_{n(2^*)}] - \mathbf{Z}_{n(2^*)}^w \right| \leq \frac{\lambda_n}{2\sqrt{n}} \mathbf{W}_{0(2)} \quad (3.3)$$

element-wise. Meanwhile, we also note that

$$\begin{aligned} \{|\hat{\mathbf{u}}_{n(1)}| < |\beta_{0(1)}|\} &= \{\hat{\mathbf{u}}_{n(1)} < |\beta_{0(1)}|\} \cap \{\hat{\mathbf{u}}_{n(1)} > -|\beta_{0(1)}|\} \\ &= \{\hat{\beta}_{n(1)}^w < \beta_{0(1)} + |\beta_{0(1)}|\} \cap \{\hat{\beta}_{n(1)}^w > \beta_{0(1)} - |\beta_{0(1)}|\}, \end{aligned}$$

where all inequalities hold element-wise. Thus,  $\hat{\beta}_{n(1)}^w < 0$  element-wise if  $\beta_{0(1)} < 0$  element-wise, and vice versa. In other words,

$$\{\text{sgn}(\hat{\beta}_{n(1)}^w) = \text{sgn}(\beta_{0(1)})\} \supseteq \{|\hat{\mathbf{u}}_{n(1)}| < |\beta_{0(1)}| \text{ element-wise}\}. \quad (3.4)$$

Therefore, by (3.2), (3.3), (3.4), and uniqueness of solution for the random-weighting setup (2.3), if there exists  $\hat{\mathbf{u}}_n$  such that the following equation and inequalities hold:

$$C_{n(11)}^w [\sqrt{n}\hat{\mathbf{u}}_{n(1)}] - \mathbf{Z}_{n(1)}^w = -\frac{\lambda_n}{2\sqrt{n}} \mathbf{W}_{0(1)} \circ \text{sgn}(\beta_{0(1)}) \quad (3.5)$$

$$-\frac{\lambda_n}{2\sqrt{n}} \mathbf{W}_{0(2)} \leq C_{n(21)}^w [\sqrt{n}\hat{\mathbf{u}}_{n(1)}] - \mathbf{Z}_{n(2)}^w \leq \frac{\lambda_n}{2\sqrt{n}} \mathbf{W}_{0(2)} \text{ element-wise} \quad (3.6)$$

$$|\hat{\mathbf{u}}_{n(1)}| < |\beta_{0(1)}| \text{ element-wise}, \quad (3.7)$$

then we have  $\text{sgn}(\widehat{\beta}_{n(1)}^w) = \text{sgn}[\beta_{0(1)}]$  and  $\widehat{\mathbf{u}}_{n(2)} = \widehat{\beta}_{n(2)}^w = \beta_{0(2)} = \mathbf{0}$ , ie.

$$\widehat{\beta}_n^w \stackrel{s}{=} \beta_0,$$

and

$$\begin{aligned} & P\left(\widehat{\beta}_n^w \stackrel{s}{=} \beta_0 \middle| \mathcal{F}_n\right) \\ & \geq P\left(\left\{\left|C_{n(21)}^w[\sqrt{n}\widehat{\mathbf{u}}_{n(1)}] - \mathbf{Z}_{n(2)}^w\right| \leq \frac{\lambda_n}{2\sqrt{n}}\mathbf{W}_{0(2)} \text{ element-wise}\right\} \right. \\ & \quad \left. \cap \left\{C_{n(11)}^w[\sqrt{n}\widehat{\mathbf{u}}_{n(1)}] - \mathbf{Z}_{n(1)}^w = -\frac{\lambda_n}{2\sqrt{n}}\mathbf{W}_{0(1)} \circ \text{sgn}[\beta_{0(1)}]\right\} \right. \\ & \quad \left. \cap \left\{|\widehat{\mathbf{u}}_{n(1)}| < |\beta_{0(1)}| \text{ element-wise}\right\} \middle| \mathcal{F}_n\right). \end{aligned}$$

Now we proceed to simplify these equation and inequalities (3.5), (3.6) and (3.7). Equation (3.5) can be re-written as

$$\sqrt{n}\widehat{\mathbf{u}}_{n(1)} = \left(C_{n(11)}^w\right)^{-1} \left[\mathbf{Z}_{n(1)}^w - \frac{\lambda_n}{2\sqrt{n}}\mathbf{W}_{0(1)} \circ \text{sgn}[\beta_{0(1)}]\right]. \quad (3.8)$$

Substituting inequality (3.7) into equation (3.8) above leads to  $A_n^w$ . Replace the expression

$$\mathbf{W}_{0(1)} \circ \text{sgn}[\beta_{0(1)}]$$

in equation (3.8) with  $W_0 \text{sgn}[\beta_{0(1)}]$  and  $\text{sgn}[\beta_{0(1)}]$  for weighting schemes (2.5) and (2.4) respectively to obtain  $A_n^w$ .

Next, substituting equation (3.8) into inequality (3.6) and simple arithmetic yield

$$\begin{aligned} \widetilde{B}_n^w & \equiv \left\{ \left| \widetilde{C}_n^w \mathbf{Z}_{n(1)}^w + \mathbf{Z}_{n(3)}^w - \frac{\lambda_n}{2\sqrt{n}} C_{n(21)}^w \left(C_{n(11)}^w\right)^{-1} \mathbf{W}_{0(1)} \circ \text{sgn}[\beta_{0(1)}] \right| \right. \\ & \quad \left. - \frac{\lambda_n}{2\sqrt{n}} \left| C_{n(21)} C_{n(11)}^{-1} \mathbf{W}_{0(1)} \circ \text{sgn}[\beta_{0(1)}] \right| \right. \\ & \quad \left. \leq \frac{\lambda_n}{2\sqrt{n}} \left( \mathbf{W}_{0(2)} - \left| C_{n(21)} C_{n(11)}^{-1} \mathbf{W}_{0(1)} \circ \text{sgn}[\beta_{0(1)}] \right| \right) \text{ element-wise} \right\} \end{aligned}$$



for weighting scheme (2.6). Now, observe that  $B_n^w \subseteq \tilde{B}_n^w$ , since (LHS of  $B_n^w$ )  $\geq$  (LHS of  $\tilde{B}_n^w$ ) element-wise. Thus,

$$P\left(\hat{\beta}_n^w \stackrel{s}{=} \beta_0 \mid \mathcal{F}_n\right) \geq P\left(A_n^w \cap \tilde{B}_n^w \mid \mathcal{F}_n\right) \geq P\left(A_n^w \cap B_n^w \mid \mathcal{F}_n\right).$$

For weighting scheme (2.5),

$$\begin{aligned} \tilde{B}_n^w &\equiv \left\{ \left| \tilde{C}_n^w \mathbf{Z}_{n(1)}^w + \mathbf{Z}_{n(3)}^w - \frac{\lambda_n W_0}{2\sqrt{n}} C_{n(21)}^w \left(C_{n(11)}^w\right)^{-1} \text{sgn}[\beta_{0(1)}] \right| \right. \\ &\quad \left. - \frac{\lambda_n W_0}{2\sqrt{n}} \left| C_{n(21)} C_{n(11)}^{-1} \text{sgn}[\beta_{0(1)}] \right| \right. \\ &\quad \left. \leq \frac{\lambda_n W_0}{2\sqrt{n}} \left( \mathbf{1}_{p_n - q} - \left| C_{n(21)} C_{n(11)}^{-1} \text{sgn}[\beta_{0(1)}] \right| \right) \text{ element-wise} \right\}. \end{aligned} \quad (3.9)$$

Now, observe that  $B_n^w \subseteq \tilde{B}_n^w$ , since (LHS of  $B_n^w$ )  $\geq$  (LHS of  $\tilde{B}_n^w$ ) element-wise, whereas (RHS of  $B_n^w$ )  $\leq$  (RHS of  $\tilde{B}_n^w$ ) element-wise due to the Irrepresentable condition (2.11).

Therefore,

$$P\left(\hat{\beta}_n^w \stackrel{s}{=} \beta_0 \mid \mathcal{F}_n\right) \geq P\left(A_n^w \cap \tilde{B}_n^w \mid \mathcal{F}_n\right) \geq P\left(A_n^w \cap B_n^w \mid \mathcal{F}_n\right).$$

For weighting scheme (2.4), substitute  $W_0 = 1$  in (3.9) and the result follows.  $\square$

*Proof of Theorem 2.2.* From Proposition 2.1,

$$\begin{aligned} P\left(\hat{\beta}_n^w(\lambda_n) \stackrel{s}{=} \beta_0 \mid \mathcal{F}_n\right) &\geq P\left(A_n^w \cap B_n^w \mid \mathcal{F}_n\right) \\ &= 1 - P\left[\left(A_n^w \cap B_n^w\right)^c \mid \mathcal{F}_n\right] \\ &= 1 - P\left[\left(A_n^w\right)^c \cup \left(B_n^w\right)^c \mid \mathcal{F}_n\right] \\ &\geq 1 - \left\{ P\left[\left(A_n^w\right)^c \mid \mathcal{F}_n\right] + P\left[\left(B_n^w\right)^c \mid \mathcal{F}_n\right] \right\}. \end{aligned}$$

We now investigate the conditional probabilities  $P\left[\left(A_n^w\right)^c \mid \mathcal{F}_n\right]$  and  $P\left[\left(B_n^w\right)^c \mid \mathcal{F}_n\right]$  separately. All three weighting schemes (2.4), (2.5) and (2.6) share very similar  $P\left[\left(A_n^w\right)^c \mid \mathcal{F}_n\right]$ . We start off with the most general version (2.6) of the weighting schemes. Results for the other

two simpler weighting schemes could then be easily inferred. For ease of notation, let

$$\mathbf{z}_n = [z_{n,1}, \dots, z_{n,q}]' := \left( C_{n(11)}^w \right)^{-1} \left( \mathbf{z}_{n(1)}^w - \frac{\lambda_n}{2\sqrt{n}} \mathbf{W}_{0(1)} \circ \text{sgn} [\boldsymbol{\beta}_{0(1)}] \right).$$

Note that

$$\frac{\lambda_n}{2n} \mathbf{W}_{0(1)} \circ \text{sgn} [\boldsymbol{\beta}_{0(1)}] \xrightarrow{p} \mathbf{0}.$$

Hence, by Lemmas 3.2 and 3.6,

$$\begin{aligned} P[(A_n^w)^c | \mathcal{F}_n] &= P \left( \bigcup_{j=1}^q \{ |z_{n,j}| > \sqrt{n} |\beta_{0,j}| \} \middle| \mathcal{F}_n \right) \\ &\leq \sum_{j=1}^q P \left( \frac{1}{\sqrt{n}} |z_{n,j}| > |\beta_{0,j}| \middle| \mathcal{F}_n \right) \\ &\rightarrow 0 \quad a.s. P_D, \end{aligned}$$

because for all  $j = 1, \dots, q$ , we have  $|\beta_{0,j}| > 0$  but

$$\frac{1}{\sqrt{n}} |z_{n,j}| = o_p(1) \quad a.s. P_D.$$

For weighting schemes (2.5) and (2.4), replace the expression

$$\mathbf{W}_{0(1)} \circ \text{sgn} [\boldsymbol{\beta}_{0(1)}]$$

with  $W_0 \text{sgn} [\boldsymbol{\beta}_{0(1)}]$  and  $\text{sgn} [\boldsymbol{\beta}_{0(1)}]$  respectively to obtain the same result

$$P[(A_n^w)^c | \mathcal{F}_n] \rightarrow 0 \quad a.s. P_D.$$

We now turn our attention to  $P[(B_n^w)^c | \mathcal{F}_n]$ , where weighting scheme (2.6) is markedly different – and derived separately – from weighting schemes (2.4) and (2.5). We first consider weighting scheme (2.5), and then infer the result for weighting scheme (2.4) as a

special case. For ease of notation, define

$$\begin{aligned}\zeta_n &= [\zeta_{n,1}, \dots, \zeta_{n,p_n-q}]' := \mathbf{Z}_{n(3)}^w, \\ \nu_n &= [\nu_{n,1}, \dots, \nu_{n,p_n-q}]' := \tilde{C}_n^w \left( \mathbf{Z}_{n(1)}^w - \frac{\lambda_n W_0}{2\sqrt{n}} \text{sgn} [\boldsymbol{\beta}_{0(1)}] \right).\end{aligned}$$

Then, for any  $\xi > 0$ ,

$$\begin{aligned}& P[(B_n^w)^c | \mathcal{F}_n] \\ &= P\left(\bigcup_{j=1}^{p_n-q} \left\{ |\zeta_{n,j} + \nu_{n,j}| > \frac{\lambda_n}{2\sqrt{n}} \eta_j \right\} \middle| \mathcal{F}_n\right) \\ &\leq P\left(\bigcup_{j=1}^{p_n-q} \left\{ |\zeta_{n,j}| + |\nu_{n,j}| > \frac{\lambda_n}{2\sqrt{n}} \eta_j \right\} \middle| \mathcal{F}_n\right) \\ &\leq P\left(\bigcup_{j=1}^{p_n-q} \left[ \left\{ |\zeta_{n,j}| + |\nu_{n,j}| > \frac{\lambda_n}{2\sqrt{n}} \eta_j \right\} \cap \{ |\nu_{n,j}| \leq \xi \} \right] \middle| \mathcal{F}_n\right) \\ &\quad + P\left(\bigcup_{j=1}^{p_n-q} \left[ \left\{ |\zeta_{n,j}| + |\nu_{n,j}| > \frac{\lambda_n}{2\sqrt{n}} \eta_j \right\} \cap \{ |\nu_{n,j}| > \xi \} \right] \middle| \mathcal{F}_n\right) \\ &\leq P\left(\bigcup_{j=1}^{p_n-q} \left\{ |\zeta_{n,j}| > \frac{\lambda_n}{2\sqrt{n}} \eta_j - \xi \right\} \middle| \mathcal{F}_n\right) + P\left(\bigcup_{j=1}^{p_n-q} \{ |\nu_{n,j}| > \xi \} \middle| \mathcal{F}_n\right) \\ &\leq P\left(\bigcup_{j=1}^{p_n-q} \left\{ |\zeta_{n,j}| > \frac{\lambda_n W_0}{2\sqrt{n}} \eta_j - \xi \right\} \middle| \mathcal{F}_n\right) + P(\|\nu_n\|_2 > \xi | \mathcal{F}_n).\end{aligned}$$

Since

$$\frac{\lambda_n W_0}{n^{1.5-c_1}} \text{sgn} [\boldsymbol{\beta}_{0(1)}] = o_p(1),$$

we have, by Lemmas 3.3 and 3.6,

$$\|\nu_n\|_2 \leq \left\| n^{1-c_1} \tilde{C}_n^w \right\|_2 \left\| \frac{1}{n^{1-c_1}} \mathbf{Z}_{n(1)}^w - \frac{\lambda_n W_0}{2n^{1.5-c_1}} \text{sgn} [\boldsymbol{\beta}_{0(1)}] \right\|_2 = o_p(1) \quad a.s. P_D,$$

and thus,

$$P(\|\nu_n\|_2 > \xi | \mathcal{F}_n) = o(1) \quad a.s. P_D.$$

Now, let

$$\eta_* = \min_{1 \leq j \leq p_n - q} \eta_j,$$

and note that  $0 < \eta_* \leq 1$  from assumption (2.11). Then,

$$\begin{aligned} & P \left( \bigcup_{j=1}^{p_n - q} \left\{ |\zeta_{n,j}| > \frac{\lambda_n W_0}{2\sqrt{n}} \eta_j - \xi \right\} \middle| \mathcal{F}_n \right) \\ & \leq P \left( \bigcup_{j=1}^{p_n - q} \left\{ |\zeta_{n,j}| > \frac{\lambda_n W_0}{2\sqrt{n}} \eta_* - \xi \right\} \middle| \mathcal{F}_n \right) \\ & = P \left( \max_{1 \leq j \leq p_n - q} |\zeta_{n,j}| > \frac{\lambda_n W_0}{2\sqrt{n}} \eta_* - \xi \middle| \mathcal{F}_n \right) \\ & \leq P \left( \|\zeta_n\|_2 > \frac{\lambda_n W_0}{2\sqrt{n}} \eta_* - \xi \middle| \mathcal{F}_n \right) \\ & = P \left( \frac{1}{n^{c_2 - \frac{1}{2}}} (\|\zeta_n\|_2 + \xi) > \frac{\lambda_n W_0}{2n^{c_2}} \eta_* \middle| \mathcal{F}_n \right) \\ & = o(1) \quad a.s. P_D, \end{aligned}$$

because

$$\frac{\lambda_n W_0}{2n^{c_2}} \eta_* = \mathcal{O}_p(1)$$

whereas part (a) of Lemma 3.7 ensures that

$$\frac{1}{n^{c_2 - \frac{1}{2}}} (\|\zeta_n\|_2 + \xi) = o_p(1) \quad a.s. P_D.$$

Thus, for weighting scheme (2.5), we have just shown that

$$P [(B_n^w)^c | \mathcal{F}_n] = o(1) \quad a.s. P_D.$$

For weighting scheme (2.4), take  $W_0 = 1$  and repeat the preceding steps to obtain the same result.

Now, for weighting scheme (2.6), define

$$\begin{aligned}\boldsymbol{\nu}_n &= [\nu_{n,1}, \dots, \nu_{n,p_n-q}]' := \tilde{C}_n^w \left( \mathbf{Z}_{n(1)}^w - \frac{\lambda_n}{2\sqrt{n}} \mathbf{W}_{0(1)} \circ \text{sgn} [\boldsymbol{\beta}_{0(1)}] \right), \\ \boldsymbol{\gamma}_n &= [\gamma_{n,1}, \dots, \gamma_{n,p_n-q}]' := C_{n(21)} C_{n(11)}^{-1} \mathbf{W}_{0(1)} \circ \text{sgn} [\boldsymbol{\beta}_{0(1)}].\end{aligned}$$

and for any  $\xi > 0$ ,

$$\begin{aligned}& P \left[ (B_n^w)^c \mid \mathcal{F}_n \right] \\ &= P \left( \bigcup_{j=1}^{p_n-q} \left\{ |\zeta_{n,j} + \nu_{n,j}| > \frac{\lambda_n}{2\sqrt{n}} (W_{0(2),j} - |\gamma_{n,j}|) \right\} \mid \mathcal{F}_n \right) \\ &\leq P \left( \bigcup_{j=1}^{p_n-q} \left\{ |\zeta_{n,j}| > \frac{\lambda_n}{2\sqrt{n}} (W_{0(2),j} - |\gamma_{n,j}|) - \xi \right\} \mid \mathcal{F}_n \right) + P \left( \|\boldsymbol{\nu}_n\|_2 > \xi \mid \mathcal{F}_n \right).\end{aligned}$$

Again,

$$\frac{\lambda_n}{n^{1.5-c_1}} \mathbf{W}_{0(1)} \circ \text{sgn} [\boldsymbol{\beta}_{0(1)}] = o_p(1),$$

so, by Lemmas 3.3 and 3.6,

$$P \left( \|\boldsymbol{\nu}_n\|_2 > \xi \mid \mathcal{F}_n \right) = o(1) \quad a.s. P_D.$$

Notice how the penalty weights  $\mathbf{W}_{0(1)}$  and  $\mathbf{W}_{0(2)}$  upend the strong irrerepresentable condition (2.11). Specifically,

$$P \left( W_{0(2),j} - |\gamma_{n,j}| < 0 \right) > 0,$$

which then renders the probability bound to be unhelpful. Instead, notice that from the strong irrerepresentable condition (2.11),

$$\gamma_{n,j} \leq (1 - \eta_*) \times \max_{1 \leq j \leq q} W_{0(1),j}$$

for all  $j = 1, \dots, q$ . We focus on the more restrictive case where

$$\eta_* = 1 \iff \boldsymbol{\eta} = \mathbf{1}_{p_n - q},$$

which leads to a more meaningful probability bound. Then,  $\gamma_{n,j} = 0$  for all  $j = 1, \dots, q$ , and

$$\begin{aligned} & P \left( \bigcup_{j=1}^{p_n - q} \left\{ |\zeta_{n,j}| > \frac{\lambda_n}{2\sqrt{n}} W_{0(2),j} - \xi \right\} \middle| \mathcal{F}_n \right) \\ & \leq P \left( \bigcup_{j=1}^{p_n - q} \left\{ |\zeta_{n,j}| > \frac{\lambda_n}{2\sqrt{n}} \left( \min_{1 \leq j \leq p_n - q} W_{0(2),j} \right) - \xi \right\} \middle| \mathcal{F}_n \right) \\ & \leq P \left( \left\| \boldsymbol{\zeta}_n \right\|_2 > \frac{\lambda_n}{2\sqrt{n}} \left( \min_{1 \leq j \leq p_n - q} W_{0(2),j} \right) - \xi \middle| \mathcal{F}_n \right) \\ & = P \left( \frac{1}{n^{c_2 - \frac{1}{2}}} \left( \left\| \boldsymbol{\zeta}_n \right\|_2 + \xi \right) > \frac{\lambda_n}{2n^{c_2}} \left( \min_{1 \leq j \leq p_n - q} W_{0(2),j} \right) \middle| \mathcal{F}_n \right) \end{aligned}$$

For the case of exponential random weights

$$F_W(w) = 1 - e^{-\theta_w w}$$

for some  $\theta_w > 0$ , we immediately have

$$\left( \min_{1 \leq j \leq p_n - q} W_{0(2),j} \right) \sim \text{Exp}((p_n - q)\theta_w).$$

Then, by part (b) of Lemma 3.7,

$$\begin{aligned} & P \left( \frac{1}{n^{c_2 - \frac{1}{2}}} \left( \left\| \boldsymbol{\zeta}_n \right\|_2 + \xi \right) > \frac{\lambda_n}{2n^{c_2}} \left( \min_{1 \leq j \leq p_n - q} W_{0(2),j} \right) \middle| \mathcal{F}_n \right) \\ & = P \left( W < \theta_w \frac{2n^{c_2}}{\lambda_n} \frac{p_n - q}{n^{c_2 - \frac{1}{2}}} \left( \left\| \boldsymbol{\zeta}_n \right\|_2 + \xi \right) \middle| \mathcal{F}_n \right) \text{ where } W \sim \text{Exp}(1) \\ & = o(1) \quad \text{a.s. } P_D, \end{aligned}$$

and we have just shown that

$$P[(B_n^w)^c | \mathcal{F}_n] = o(1) \quad a.s. P_D$$

for weighting scheme (2.6).

Finally,

$$\begin{aligned} & P\left(\widehat{\beta}_n^w(\lambda_n) \stackrel{s}{=} \beta_0 | \mathcal{F}_n\right) \\ & \geq 1 - \left\{ P[(A_n^w)^c | \mathcal{F}_n] + P[(B_n^w)^c | \mathcal{F}_n] \right\} \\ & = 1 - o(1) \quad a.s. P_D \end{aligned}$$

for all three weighting schemes (2.4), (2.5) and (2.6). □

*Proof of Theorem 2.3.* From the proof of Proposition 2.1,

$$\begin{aligned} & (\widehat{\beta}_n^w - \beta_0) \\ & = \arg \min_{\mathbf{u}} \left\{ \mathbf{u}' \left( \frac{X' D_n X}{n} \right) \mathbf{u} - 2\mathbf{u}' \left( \frac{X' D_n \boldsymbol{\epsilon}}{n} \right) + \frac{\boldsymbol{\epsilon}' D_n \boldsymbol{\epsilon}}{n} \right. \\ & \quad \left. + \frac{\lambda_n}{n} \sum_{j=1}^p W_{0,j} |\beta_{0,j} + u_{n,j}| \right\} \\ & := \arg \min_{\mathbf{u}} g_n(\mathbf{u}). \end{aligned}$$

By Lemmas 3.4, 3.5 and 3.8, for  $\frac{\lambda_n}{n} \rightarrow \lambda_0 \in [0, \infty)$ , Slutsky Theorem gives

$$g_n(\mathbf{u}) \xrightarrow{\text{c.d.}} g(\mathbf{u}) + \mu_W \sigma_\epsilon^2 \quad a.s. P_D.$$

Note that for weighting schemes (2.5) and (2.6),  $g(\mathbf{u})$  is a random function as it contains random weights. Since  $g_n(\mathbf{u})$  is convex and  $g(\mathbf{u})$  has a unique minimum, it follows from

Geyer (1996) that

$$\arg \min_{\mathbf{u}} g_n(\mathbf{u}) \xrightarrow{\text{c.d.}} \arg \min_{\mathbf{u}} \{g(\mathbf{u}) + \mu_W \sigma_\epsilon^2\} = \arg \min_{\mathbf{u}} g(\mathbf{u}) \quad a.s. P_D.$$

For weighting schemes (2.4),  $g(\mathbf{u})$  is not a random function. Instead, we note that since  $g_n(\mathbf{u})$  is convex, it follows from pointwise convergence of conditional probability that

$$\widehat{\beta}_n^w - \beta_0 = \mathcal{O}_p(1).$$

For any compact set  $K$ , by applying the Convexity Lemma (Pollard, 1991),

$$\sup_{\mathbf{u} \in K} |g_n(\mathbf{u}) - g(\mathbf{u}) - \mu_W \sigma_\epsilon^2| \xrightarrow{\text{c.P.}} 0 \quad a.s. P_D.$$

Therefore,

$$\left(\widehat{\beta}_n^w - \beta_0\right) = \arg \min_{\mathbf{u}} g_n(\mathbf{u}) \xrightarrow{\text{c.P.}} \arg \min_{\mathbf{u}} g(\mathbf{u}) \quad a.s. P_D.$$

Finally, for all three weighting schemes, if  $\lambda_0 = 0$ ,  $\arg \min_{\mathbf{u}} g(\mathbf{u}) = \mathbf{0}$ , i.e.

$$\widehat{\beta}_n^w \xrightarrow{\text{c.P.}} \beta_0 \quad a.s. P_D.$$

□

*Proof of Theorem 2.4.* Let  $e_n$  be the residual that corresponds to the strongly consistent estimator  $\widehat{\beta}_n^{\text{SC}}$  of the linear regression model (2.1), and define

$$Q_n(\mathbf{z}) := \left\| D_n^{\frac{1}{2}}(\mathbf{y} - X\mathbf{z}) \right\|_2^2 + \lambda_n \sum_{j=1}^p W_{0,j} |z_j|,$$

which leads to

$$Q_n \left( \widehat{\beta}_n^{\text{SC}} + \frac{1}{\sqrt{n}} \mathbf{u} \right)$$



$$\begin{aligned}
&= \left\| D_n^{\frac{1}{2}} \left[ Y - X \left( \widehat{\beta}_n^{\text{SC}} + \frac{1}{\sqrt{n}} \mathbf{u} \right) \right] \right\|_2^2 + \lambda_n \sum_{j=1}^p W_{0,j} \left| \widehat{\beta}_{n,j}^{\text{SC}} + \frac{1}{\sqrt{n}} u_j \right| \\
&= \left\| D_n^{\frac{1}{2}} \left( \mathbf{e}_n - \frac{1}{\sqrt{n}} X \mathbf{u} \right) \right\|_2^2 + \lambda_n \sum_{j=1}^p W_{0,j} \left| \widehat{\beta}_{n,j}^{\text{SC}} + \frac{1}{\sqrt{n}} u_j \right|,
\end{aligned}$$

and

$$\begin{aligned}
Q_n \left( \widehat{\beta}_n^{\text{SC}} \right) &= \left\| D_n^{\frac{1}{2}} \left( Y - X \widehat{\beta}_n^{\text{SC}} \right) \right\|_2^2 + \lambda_n \sum_{j=1}^p W_{0,j} \left| \widehat{\beta}_{n,j}^{\text{SC}} \right| \\
&= \left\| D_n^{\frac{1}{2}} \mathbf{e}_n \right\|_2^2 + \lambda_n \sum_{j=1}^p W_{0,j} \left| \widehat{\beta}_{n,j}^{\text{SC}} \right|.
\end{aligned}$$

Now, define

$$V_n(\mathbf{u}) := Q_n \left( \widehat{\beta}_n^{\text{SC}} + \frac{1}{\sqrt{n}} \mathbf{u} \right) - Q_n \left( \widehat{\beta}_n^{\text{SC}} \right),$$

and note that

$$\arg \min_{\mathbf{u}} V_n(\mathbf{u}) = \arg \min_{\mathbf{u}} Q_n \left( \widehat{\beta}_n^{\text{SC}} + \frac{1}{\sqrt{n}} \mathbf{u} \right) = \sqrt{n} \left( \widehat{\beta}_n^w - \widehat{\beta}_n^{\text{SC}} \right).$$

Notice that  $V_n(\mathbf{u})$  can be simplified into

$$\begin{aligned}
&\mathbf{u}' \left( \frac{X' D_n X}{n} \right) \mathbf{u} - 2 \mathbf{u}' \left( \frac{X' D_n \mathbf{e}_n}{\sqrt{n}} \right) \\
&+ \frac{\lambda_n}{\sqrt{n}} \sum_{j=1}^p W_{0,j} \left( \left| \sqrt{n} \widehat{\beta}_{n,j}^{\text{SC}} + u_j \right| - \left| \sqrt{n} \widehat{\beta}_{n,j}^{\text{SC}} \right| \right),
\end{aligned}$$

where its penalty term can be expanded into

$$\begin{aligned}
&\frac{\lambda_n}{\sqrt{n}} \sum_{j=1}^p W_{0,j} \left( \left| \sqrt{n} \widehat{\beta}_{n,j}^{\text{SC}} + u_j \right| - \left| \sqrt{n} \widehat{\beta}_{n,j}^{\text{SC}} \right| \right) \\
&= \frac{\lambda_n}{\sqrt{n}} \sum_{j=1}^p W_{0,j} \left\{ \left| \sqrt{n} \left[ \beta_{0,j} + \left( \widehat{\beta}_{n,j}^{\text{SC}} - \beta_{0,j} \right) \right] + \mu_j \right| \right. \\
&\quad \left. - \left| \sqrt{n} \left[ \beta_{0,j} + \left( \widehat{\beta}_{n,j}^{\text{SC}} - \beta_{0,j} \right) \right] \right| \right\}
\end{aligned}$$

$$:= \frac{\lambda_n}{\sqrt{n}} \sum_{j=1}^p W_{0,j} p_n(u_j).$$

For  $\beta_{0,j} \neq 0$ ,

$$\left( \widehat{\beta}_{n,j}^{\text{SC}} - \beta_{0,j} \right) \rightarrow 0 \quad a.s. \ P_D,$$

and hence  $\sqrt{n}\beta_{0,j}$  dominates  $u_j$  for large  $n$ . Thus, it is easy to verify that  $p_n(u_j)$  converges to  $u_j \text{sgn}(\beta_{0,j})$  for all  $j \in \{j : \beta_{0,j} \neq 0\}$ . Thus, by Lemmas 3.4 and 3.9, if  $q = p$ , Slutsky Theorem ensures that

$$V_n(\mathbf{u}) \xrightarrow{\text{c.d.}} V(\mathbf{u}) := \mu_W \mathbf{u}' C \mathbf{u} - 2\mathbf{u}' \Psi + \lambda_0 \sum_{j=1}^p W_j [u_j \text{sgn}(\beta_{0,j})] \quad a.s. \ P_D,$$

where  $\Psi$  has a  $N(\mathbf{0}, \sigma_W^2 \sigma_\epsilon^2 C)$  distribution, and

- (i)  $W_j = 1$  for all  $j$  under weighting scheme (2.4),
- (ii)  $W_j = W_0$  for all  $j$ ,  $W_0 \sim F_W$  and  $W_0 \perp \Psi$  under weighting scheme (2.5),
- (iii)  $W_j \stackrel{iid}{\sim} F_W$  and  $W_j \perp \Psi$  for all  $j$  under weighting scheme (2.6).

Since  $V_n(\mathbf{u})$  is convex and  $V(\mathbf{u})$  has a unique minimum, it follows from Geyer (1996) that

$$\sqrt{n} \left( \widehat{\beta}_n^w - \widehat{\beta}_n^{\text{SC}} \right) = \arg \min_{\mathbf{u}} V_n(\mathbf{u}) \xrightarrow{\text{c.d.}} \arg \min_{\mathbf{u}} V(\mathbf{u}) \quad a.s. \ P_D$$

when  $q = p$ . In particular, if  $\lambda_0 = 0$ ,

$$\arg \min_{\mathbf{u}} V(\mathbf{u}) = \frac{1}{\mu_W} C^{-1} \Psi \sim N \left( \mathbf{0}, \frac{\sigma_W^2 \sigma_\epsilon^2}{\mu_W^2} C^{-1} \right).$$

However, if  $0 < q < p$ , then for  $j \in \{j : \beta_{0,j} = 0\}$ ,  $p_n(u_j)$  is back to

$$\left| \sqrt{n} \widehat{\beta}_{n,j}^{\text{SC}} + \mu_j \right| - \left| \sqrt{n} \widehat{\beta}_{n,j}^{\text{SC}} \right|,$$

which depends on the sample path of realized data. This necessitates the Skorokhod argument, thus leading to the penalty term in (2.13).  $\square$

We need the following lemma to prove Theorem 2.5:

**Lemma 3.10.** *Consider Liu and Yu (2013)'s unweighted two-step LASSO+LS estimator  $\widehat{\beta}_n^{LAS+LS}$ , with its corresponding set of selected variables denoted as  $\widehat{S}_n$ . Adopt assumptions (2.8), (2.9) and (2.11). If there exists  $\frac{1}{2} < c_1 < c_2 < 1$  and  $0 \leq c_3 < 2(c_2 - c_1)$  for which  $\lambda_n = \mathcal{O}(n^{c_2})$  and  $p_n = \mathcal{O}(n^{c_3})$ , then as  $n \rightarrow \infty$ ,*

$$P\left(\widehat{S}_n = S_0 \mid \mathcal{F}_n\right) \rightarrow 1 \quad a.s. P_D.$$

*Proof.* The first step (i.e. the variable selection step) of obtaining  $\widehat{\beta}_n^{LAS+LS}$  is effectively the standard LASSO procedure. Thus, by assumption (2.11), from the proof of Proposition 1 of Zhao and Yu (2006), we obtain

$$\left\{\widehat{S}_n = S_0\right\} \supseteq \{A_n \cap B_n\}$$

and thus

$$P\left(\widehat{S}_n = S_0 \mid \mathcal{F}_n\right) \geq P\left(A_n \cap B_n \mid \mathcal{F}_n\right),$$

where

$$A_n \equiv \left\{ \left| C_{n(11)}^{-1} \frac{X'_{(1)} \epsilon}{\sqrt{n}} \right| \leq \sqrt{n} \left( |\beta_{0(1)}| - \frac{\lambda_n}{2n} \left| C_{n(11)}^{-1} \text{sgn}(\beta_{0(1)}) \right| \right) \text{ element-wise} \right\}$$

$$B_n \equiv \left\{ \left| \frac{1}{\sqrt{n}} \left[ C_{n(21)} C_{n(11)}^{-1} X'_{(1)} - X'_{(2)} \right] \epsilon \right| \leq \frac{\lambda_n}{2\sqrt{n}} \eta \text{ element-wise} \right\}.$$

Next, we want to show that

$$P\left(A_n^c \mid \mathcal{F}_n\right) \rightarrow 0 \quad a.s. P_D \quad \text{and} \quad P\left(B_n^c \mid \mathcal{F}_n\right) \rightarrow 0 \quad a.s. P_D$$

such that

$$P\left(\widehat{S}_n = S_0 \mid \mathcal{F}_n\right) \geq 1 - [P(A_n^c \mid \mathcal{F}_n) + P(B_n^c \mid \mathcal{F}_n)] \rightarrow 1 \quad a.s. P_D.$$

First, by assumptions (2.8) and (2.9),  $C_{n(11)}^{-1} = \mathcal{O}(1)$  for all  $n$ , whereas

$$\frac{\lambda_n}{2n} C_{n(11)}^{-1} \text{sgn}(\beta_{0(1)}) \rightarrow \mathbf{0}.$$

By Lemma 3.1, for any  $\frac{1}{2} < c' < 1$ ,

$$\frac{1}{n^{c'}} X'_{(1)} \boldsymbol{\epsilon} \rightarrow \mathbf{0} \quad a.s. P_D \quad \implies \quad \frac{1}{n^{c' - \frac{1}{2}}} \left( C_{n(11)}^{-1} \frac{X'_{(1)} \boldsymbol{\epsilon}}{\sqrt{n}} \right) \rightarrow \mathbf{0} \quad a.s. P_D.$$

For ease of notation, let

$$\mathbf{z} = [z_1, \dots, z_q]' := C_{n(11)}^{-1} \frac{X'_{(1)} \boldsymbol{\epsilon}}{\sqrt{n}}.$$

Then, for any  $\frac{1}{2} < c' < 1$ ,

$$\begin{aligned} P(A_n^c \mid \mathcal{F}_n) &\leq \sum_{j=1}^q P\left(|z_j| > \sqrt{n} [|\beta_{0,j}| + o(1)] \mid \mathcal{F}_n\right) \\ &= \sum_{j=1}^q P\left(\frac{|z_j|}{n^{c' - \frac{1}{2}}} > n^{1-c'} [|\beta_{0,j}| + o(1)] \mid \mathcal{F}_n\right) \\ &\rightarrow 0 \quad a.s. P_D. \end{aligned}$$

Next, using the same notations that we introduced in the proofs of Lemma 3.7 and Theorem 2.2, let

$$H = X_{(1)} C_{n(11)}^{-1} C_{n(12)} - X_{(2)},$$

and let

$$\eta_* = \min_{1 \leq j \leq p_n - q} \boldsymbol{\eta},$$

where assumption (2.11) ensures that  $0 < \eta_* \leq 1$ . Again, due to assumptions (2.8) and (2.9)

and that  $q$  is fixed, every element in the matrix  $H$  is bounded. Let  $h_{ij}$  be the  $(i, j)^{th}$  element of  $H$ . Again, by Lemma 3.1, for all  $j = 1, \dots, p_n - q$ ,

$$\frac{1}{n^{c_1}} \sum_{i=1}^n h_{ji} \epsilon_i \rightarrow 0 \quad a.s. P_D$$

for  $\frac{1}{2} < c_1 < 1$ . Consequently, we have

$$\begin{aligned} P(B_n^c | \mathcal{F}_n) &= P\left(\bigcup_{j=1}^{p_n-q} \left\{ \left| \frac{1}{\sqrt{n}} \sum_{i=1}^n h_{ji} \epsilon_i \right| > \frac{\lambda_n}{2\sqrt{n}} \eta_j \right\} \middle| \mathcal{F}_n\right) \\ &\leq P\left(\max_{1 \leq j \leq p_n-q} \left| \frac{1}{\sqrt{n}} \sum_{i=1}^n h_{ji} \epsilon_i \right| > \frac{\lambda_n}{2\sqrt{n}} \eta_* \middle| \mathcal{F}_n\right) \\ &\leq P\left(\left\| \frac{1}{\sqrt{n}} H' \epsilon \right\|_2 > \frac{\lambda_n}{2\sqrt{n}} \eta_* \middle| \mathcal{F}_n\right) \\ &= P\left(\frac{1}{n^{c_2 - \frac{1}{2}}} \left\| \frac{1}{\sqrt{n}} H' \epsilon \right\|_2 > \frac{\lambda_n}{2n^{c_2}} \eta_* \middle| \mathcal{F}_n\right), \end{aligned}$$

where

$$\begin{aligned} \left(\frac{1}{n^{c_2 - \frac{1}{2}}} \left\| \frac{1}{\sqrt{n}} H' \epsilon \right\|_2\right)^2 &= \frac{1}{n^{2c_2 - 1}} \sum_{j=1}^{p_n-q} \left(\frac{1}{\sqrt{n}} \sum_{i=1}^n h_{ji} \epsilon_i\right)^2 \\ &= \frac{n^{2c_1 - 1}}{n^{2c_2 - 1}} \sum_{j=1}^{p_n-q} \left(\frac{1}{n^{c_1}} \sum_{i=1}^n h_{ji} \epsilon_i\right)^2 \\ &= \mathcal{O}\left(\frac{1}{n^{2(c_2 - c_1)}}\right) \times o(n^{c_3}) \quad a.s. P_D \\ &= o(1) \quad a.s. P_D \end{aligned}$$

because  $c_3 < 2(c_2 - c_1)$  and  $\frac{1}{2} < c_1 < c_2 < 1$ , whereas

$$\frac{\lambda_n}{2n^{c_2}} \eta_* = \mathcal{O}(1).$$

Hence  $P(B_n^c | \mathcal{F}_n) \rightarrow 0$  almost surely under  $P_D$  and the result follows.  $\square$

Note that the constraints on  $c_1$ ,  $c_2$  and  $c_3$  in Lemma 3.10 cover the more restrictive

constraints found in Theorem 2.2. Therefore, the result in Lemma 3.10 still holds under the assumptions of Theorem 2.2.

A slightly different layout of the proof for Lemma 3.10 would be as follows: using the results in Proposition 1 of Zhao and Yu (2006), on the probability space  $P_D$ ,

$$P_D(\widehat{S}_n = S_0) \geq P_D(A_n \cap B_n).$$

Using the same techniques in the preceding proof, we show that

$$\lim_{n \rightarrow \infty} A_n^c = \emptyset \quad a.s. \ P_D \implies P_D\left(\lim_{n \rightarrow \infty} A_n^c\right) = 0 \implies P_D(A_n^c \text{ i.o.}) = 0,$$

and

$$\lim_{n \rightarrow \infty} B_n^c = \emptyset \quad a.s. \ P_D \implies P_D\left(\lim_{n \rightarrow \infty} B_n^c\right) = 0 \implies P_D(B_n^c \text{ i.o.}) = 0,$$

where *i.o.* stands for “infinitely often”. Then,

$$\begin{aligned} P_D((A_n \cap B_n)^c \text{ i.o.}) &\leq P_D(A_n^c \text{ i.o.}) + P_D(B_n^c \text{ i.o.}) = 0 \\ \implies P_D(\{A_n \cap B_n\} \text{ i.o.}) &= 1 \\ \implies P_D(\{\widehat{S}_n = S_0\} \text{ i.o.}) &\geq P_D(\{A_n \cap B_n\} \text{ i.o.}) = 1 \\ \implies P_D\left(\lim_{n \rightarrow \infty} \widehat{S}_n = S_0\right) &= 1, \end{aligned}$$

and thus, on the probability space  $P = P_D \times P_W$ ,

$$P(\widehat{S}_n = S_0 | \mathcal{F}_n) \rightarrow 1 \quad a.s. \ P_D.$$

We have

$$\lim_{n \rightarrow \infty} A_n^c = \emptyset \quad a.s. \ P_D$$

because for any  $\frac{1}{2} < c' < 1$ ,

$$\frac{1}{n^{c'-\frac{1}{2}}} \left( C_{n(11)}^{-1} \frac{X'_{(1)} \boldsymbol{\epsilon}}{\sqrt{n}} \right) \rightarrow \mathbf{0} \quad a.s. P_D$$

whereas

$$n^{1-c'} \left( |\boldsymbol{\beta}_{0(1)}| - \frac{\lambda_n}{2n} \left| C_{n(11)}^{-1} \text{sgn}(\boldsymbol{\beta}_{0(1)}) \right| \right) = \mathcal{O}(n^{1-c'}).$$

Meanwhile, we establish

$$\lim_{n \rightarrow \infty} B_n^c = \emptyset \quad a.s. P_D$$

because

$$B_n^c \subseteq \left\{ \frac{1}{n^{c_2-\frac{1}{2}}} \left\| \frac{1}{\sqrt{n}} H' \boldsymbol{\epsilon} \right\|_2 > \frac{\lambda_n}{2n^{c_2}} \eta_* \right\},$$

where

$$\frac{1}{n^{c_2-\frac{1}{2}}} \left\| \frac{1}{\sqrt{n}} H' \boldsymbol{\epsilon} \right\|_2 = o(1) \quad a.s. P_D \quad \text{but} \quad \frac{\lambda_n}{2n^{c_2}} \eta_* = \mathcal{O}(1).$$

The following version of Sherman–Morrison–Woodbury matrix-inversion identity (e.g., Equation (26) of Henderson and Searle (1981)) will come in handy later: For any square matrices  $A$  and  $B$  of conformal sizes where  $A$  is invertible, we have

$$(A + B)^{-1} = A^{-1} - A^{-1} B A^{-1} (I + B A^{-1})^{-1}. \quad (3.10)$$

*Proof of Theorem 2.5.* Since the first-step is in fact equivalent to the one-step procedure, Theorem 2.2 immediately gives us

$$P(\widehat{S}_n^w = S_0 | \mathcal{F}_n) \geq P(\widehat{\boldsymbol{\beta}}_n^w \stackrel{s}{=} \boldsymbol{\beta}_0 | \mathcal{F}_n) \rightarrow 1 \quad a.s. P_D,$$

while Lemma 3.10 immediately gives us

$$P(\widehat{S}_n = S_0 | \mathcal{F}_n) \rightarrow 1 \quad a.s. P_D.$$

Conditional on  $\{\widehat{S}_n^w = S_0\}$  and  $\{\widehat{S}_n = S_0\}$ , since  $Y = X_{(1)}\beta_{0(1)} + \epsilon$ ,

$$\begin{aligned} & \widehat{\beta}_{n(1)}^w - \widehat{\beta}_{n(1)}^{LAS+LS} \\ &= \left(X'_{(1)}D_nX_{(1)}\right)^{-1} X'_{(1)}D_nY - \left(X'_{(1)}X_{(1)}\right)^{-1} X'_{(1)}Y \\ &= \left(X'_{(1)}D_nX_{(1)}\right)^{-1} X'_{(1)}D_n\epsilon - \left(X'_{(1)}X_{(1)}\right)^{-1} X'_{(1)}\epsilon \\ &= \left(C_{n(11)}^w\right)^{-1} \frac{X'_{(1)}(D_n - I_n)\epsilon}{n} - \left[C_{n(11)}^{-1} - \left(C_{n(11)}^w\right)^{-1}\right] \frac{X'_{(1)}\epsilon}{n}, \end{aligned}$$

which leads to

$$\begin{aligned} & \sqrt{n} \left(\widehat{\beta}_{n(1)}^w - \widehat{\beta}_{n(1)}^{LAS+LS}\right) \\ &= \left(C_{n(11)}^w\right)^{-1} \frac{X'_{(1)}(D_n - I_n)\epsilon}{\sqrt{n}} - \left[C_{n(11)}^{-1} - \left(C_{n(11)}^w\right)^{-1}\right] \frac{X'_{(1)}\epsilon}{\sqrt{n}}. \end{aligned}$$

Based on the (alternative) proof of Lemma 3.2, we have seen that

$$\left(C_{n(11)}^w\right)^{-1} \xrightarrow{\text{a.s.}} C_{11}^{-1},$$

and from the (alternative) proof of Lemma 3.6, we could deploy Slutsky's Theorem to obtain

$$\left(C_{n(11)}^w\right)^{-1} \frac{X'_{(1)}(D_n - I_n)\epsilon}{\sqrt{n}} \xrightarrow{\text{c.d.}} N_q(\mathbf{0}, \sigma_W^2 \sigma_\epsilon^2 C_{11}^{-1}) \quad \text{a.s. } P_D.$$

Meanwhile, we deploy the matrix inversion identity (3.10) by taking  $A = C_{n(11)}$  and

$$B = \frac{1}{n} X'_{(1)}(D_n - I_n)X_{(1)}$$

to obtain

$$\begin{aligned} \left(C_{n(11)}^w\right)^{-1} &= \left[C_{n(11)} + \frac{1}{n} X'_{(1)}(D_n - I_n)X_{(1)}\right]^{-1} \\ &= A^{-1} - A^{-1}BA^{-1} (I_q + BA^{-1})^{-1}. \end{aligned}$$



Then,

$$\begin{aligned}
& \left[ C_{n(11)}^{-1} - \left( C_{n(11)}^w \right)^{-1} \right] \frac{X'_{(1)} \boldsymbol{\epsilon}}{\sqrt{n}} \\
&= C_{n(11)}^{-1} \left[ \frac{X'_{(1)} (D_n - I_n) X_{(1)}}{n} \right] C_{n(11)}^{-1} \left[ I_q + \left( \frac{X'_{(1)} (D_n - I_n) X_{(1)}}{n} \right) C_{n(11)}^{-1} \right]^{-1} \frac{X'_{(1)} \boldsymbol{\epsilon}}{\sqrt{n}} \\
&= C_{n(11)}^{-1} \left[ \frac{X'_{(1)} (D_n - I_n) X_{(1)}}{n^{1-c}} \right] C_{n(11)}^{-1} \left[ I_q + \left( \frac{X'_{(1)} (D_n - I_n) X_{(1)}}{n} \right) C_{n(11)}^{-1} \right]^{-1} \frac{X'_{(1)} \boldsymbol{\epsilon}}{n^{\frac{1}{2}+c}},
\end{aligned}$$

where Lemma 3.1 and assumption (2.8) ensure that for any  $0 < c < \frac{1}{2}$ ,

$$\frac{1}{n^{1-c}} X'_{(1)} (D_n - I_n) X_{(1)} \xrightarrow{\text{a.s.}} \mathbf{0}$$

and

$$\frac{X'_{(1)} \boldsymbol{\epsilon}}{n^{\frac{1}{2}+c}} \rightarrow \mathbf{0} \quad \text{a.s. } P_D.$$

Since  $C_{n(11)}$  is invertible for all  $n$ , we have

$$C_{n(11)}^{-1} \rightarrow C_{11}^{-1},$$

and

$$\begin{aligned}
\left[ I_q + \left( \frac{X'_{(1)} (D_n - I_n) X_{(1)}}{n} \right) C_{n(11)}^{-1} \right]^{-1} &= C_{n(11)} \left( C_{n(11)}^w \right)^{-1} \\
&\xrightarrow{\text{a.s.}} C_{11} C_{11}^{-1} \\
&= I_q.
\end{aligned}$$

Hence,

$$\left[ C_{n(11)}^{-1} - \left( C_{n(11)}^w \right)^{-1} \right] \frac{X'_{(1)} \boldsymbol{\epsilon}}{\sqrt{n}} \xrightarrow{\text{c.P.}} \mathbf{0} \quad \text{a.s. } P_D.$$

Consequently, conditional on  $\{\widehat{S}_n^w = S_0\}$  and  $\{\widehat{S}_n = S_0\}$ , Slutsky's Theorem ensures that

$$\sqrt{n} \left( \widehat{\boldsymbol{\beta}}_{n(1)}^w - \widehat{\boldsymbol{\beta}}_{n(1)}^{LAS+LS} \right) \xrightarrow{\text{c.d.}} N_q \left( \mathbf{0}, \sigma_W^2 \sigma_\epsilon^2 C_{11}^{-1} \right) \quad \text{a.s. } P_D.$$

Finally, for any  $t \in \mathbb{R}$ ,

$$\begin{aligned}
& P\left(\sqrt{n}\left(\widehat{\beta}_{n(1)}^w - \widehat{\beta}_{n(1)}^{LAS+LS}\right) \leq t \mid \mathcal{F}_n\right) \\
&= P\left(\sqrt{n}\left(\widehat{\beta}_{n(1)}^w - \widehat{\beta}_{n(1)}^{LAS+LS}\right) \leq t, \left\{\widehat{S}_n^w = S_0, \widehat{S}_n = S_0\right\} \mid \mathcal{F}_n\right) \\
&\quad + P\left(\sqrt{n}\left(\widehat{\beta}_{n(1)}^w - \widehat{\beta}_{n(1)}^{LAS+LS}\right) \leq t, \left\{\widehat{S}_n^w = S_0, \widehat{S}_n = S_0\right\}^c \mid \mathcal{F}_n\right) \\
&\leq P\left(\sqrt{n}\left(\widehat{\beta}_{n(1)}^w - \widehat{\beta}_{n(1)}^{LAS+LS}\right) \leq t, \left\{\widehat{S}_n^w = S_0, \widehat{S}_n = S_0\right\} \mid \mathcal{F}_n\right) \\
&\quad + P\left(\left\{\widehat{S}_n^w \neq S_0\right\} \cup \left\{\widehat{S}_n \neq S_0\right\} \mid \mathcal{F}_n\right) \\
&\leq P\left(\sqrt{n}\left(\widehat{\beta}_{n(1)}^w - \widehat{\beta}_{n(1)}^{LAS+LS}\right) \leq t, \left\{\widehat{S}_n^w = S_0, \widehat{S}_n = S_0\right\} \mid \mathcal{F}_n\right) \\
&\quad + P\left(\widehat{S}_n^w \neq S_0 \mid \mathcal{F}_n\right) + P\left(\widehat{S}_n \neq S_0 \mid \mathcal{F}_n\right)
\end{aligned}$$

where

$$P\left(\widehat{S}_n^w \neq S_0 \mid \mathcal{F}_n\right) \rightarrow 0 \text{ a.s. } P_D \quad \text{and} \quad P\left(\widehat{S}_n \neq S_0 \mid \mathcal{F}_n\right) \rightarrow 0 \text{ a.s. } P_D,$$

and

$$P\left(\sqrt{n}\left(\widehat{\beta}_{n(1)}^w - \widehat{\beta}_{n(1)}^{LAS+LS}\right) \leq t, \left\{\widehat{S}_n^w = S_0, \widehat{S}_n = S_0\right\} \mid \mathcal{F}_n\right) \rightarrow P(Z \leq t)$$

almost surely under  $P_D$  for  $Z \sim N_q(\mathbf{0}, \sigma_W^2 \sigma_\epsilon^2 C_{11}^{-1})$ .  $\square$

*Proof of Theorem 2.6.* Since  $Y = X_{(1)}\beta_{0(1)} + \epsilon$ , by conditioning on  $\left\{\widehat{S}_n^w = S_0\right\}$ , we have

$\widehat{\beta}_{n(2)}^w = \beta_{0(2)} = \mathbf{0}$ , and

$$\begin{aligned}
\widehat{\beta}_{n(1)}^w - \beta_{0(1)} &= \left(X'_{(1)} D_n X_{(1)}\right)^{-1} X'_{(1)} D_n Y - \beta_{0(1)} \\
&= \left(X'_{(1)} D_n X_{(1)}\right)^{-1} X'_{(1)} D_n \epsilon \\
&= \left(C_{n(11)}^w\right)^{-1} \frac{X'_{(1)} D_n \epsilon}{n} \\
&\xrightarrow{\text{c.p.}} \mathbf{0} \text{ a.s. } P_D
\end{aligned}$$

by Lemmas 3.4 and 3.6. Finally, for any  $\xi > 0$ ,

$$\begin{aligned}
& P\left(\left\|\widehat{\beta}_n^w - \beta_0\right\|_2 > \xi \mid \mathcal{F}_n\right) \\
&= P\left(\left\|\widehat{\beta}_n^w - \beta_0\right\|_2 > \xi, \widehat{S}_n^w = S_0 \mid \mathcal{F}_n\right) + P\left(\left\|\widehat{\beta}_n^w - \beta_0\right\|_2 > \xi, \widehat{S}_n^w \neq S_0 \mid \mathcal{F}_n\right) \\
&\leq P\left(\left\|\widehat{\beta}_n^w - \beta_0\right\|_2 > \xi, \widehat{S}_n^w = S_0 \mid \mathcal{F}_n\right) + P\left(\widehat{S}_n^w \neq S_0 \mid \mathcal{F}_n\right) \\
&\rightarrow 0 \quad a.s. P_D.
\end{aligned}$$

□

**Remark 3.1.** Consider Theorem 2.4 with centering on  $\beta_0$

$$\sqrt{n}\left(\widehat{\beta}_n^w - \beta_0\right).$$

Using the same technique in the proof of Theorem 2.4, we work with

$$V_n(\mathbf{u}) := Q_n\left(\beta_0 + \frac{1}{\sqrt{n}}\mathbf{u}\right) - Q_n(\beta_0)$$

which can be simplified into

$$\mathbf{u}'\left(\frac{X'D_nX}{n}\right)\mathbf{u} - 2\mathbf{u}'\left(\frac{X'D_n\epsilon}{\sqrt{n}}\right) + \frac{\lambda_n}{\sqrt{n}}\sum_{j=1}^p W_{0,j}\left(\left|\sqrt{n}\beta_{0,j} + u_j\right| - \left|\sqrt{n}\beta_{0,j}\right|\right).$$

Again, assumption 2.10 ensures convergence of the first term, whereas argument for the penalty term in the proof of Theorem 2.4 still applies to the third term. However, the second term has

$$\frac{X'D_n\epsilon}{\sqrt{n}} = \frac{1}{\sqrt{n}}X'(D_n - \mu_W I_n)\epsilon + \frac{1}{\sqrt{n}}X'\epsilon,$$

where

$$\frac{1}{\sqrt{n}}X'(D_n - \mu_W I_n)\epsilon = \mathcal{O}_p(1) \quad a.s. P_D,$$

but  $(X'\epsilon)/(\sqrt{n})$  is asymptotically normal under  $P_D$  (Knight & Fu, 2000). Thus, conditional

on  $\mathcal{F}_n$ ,  $(X'D_n\epsilon)/(\sqrt{n})$  depends on the sample path of realized data  $\{y_1, y_2, \dots\}$ , thus causing  $\sqrt{n}(\hat{\beta}_n^w - \beta_0)$  to be unable to achieve convergence in conditional distribution almost surely under  $P_D$ .

**Lemma 3.11 (Rate of Convergence).** *Adopt all assumptions in Theorem 2.5. If there exists  $0 < c_4 < \frac{1}{2}$  such that*

$$0 \leq c_3 < \min\{2(c_2 - c_1), 2c_1 - 1\} - c_4$$

*under weighting schemes (2.4) and (2.5), or*

$$0 \leq c_3 < \min\left\{\frac{2}{3}(c_2 - c_1) - \frac{c_4}{3}, 2c_1 - 1 - c_4\right\}$$

*under weighting schemes (2.6), then*

$$P\left(\hat{S}_n^w \neq S_0 \mid \mathcal{F}_n\right) = o(n^{-c_4}) \quad a.s. P_D.$$

*Proof of Lemma 3.11.* The result is immediate by extracting the additional  $n^{-c_4}$  factor from the proofs of Lemmas 3.3 and 3.7 as well as Theorem 2.2. In particular, from Lemma 3.6, it is clear that the rate of convergence of  $P[(A_n^w)^c \mid \mathcal{F}_n]$  is faster than  $P[(B_n^w)^c \mid \mathcal{F}_n]$ , whereas the conditions in Lemma 3.11 ensure that  $P[(B_n^w)^c \mid \mathcal{F}_n] = o(n^{-c_4})$ . Finally,

$$P\left(\hat{S}_n^w \neq S_0 \mid \mathcal{F}_n\right) \leq P[(A_n^w)^c \mid \mathcal{F}_n] + P[(B_n^w)^c \mid \mathcal{F}_n] = o(n^{-c_4}) \quad a.s. P_D.$$

□

## Chapter 4

# Random Weighting in Discrete Mixture Models

### 4.1. Framework

#### 4.1.1. Bayesian NPL: parameters and loss functions

Regardless of idiosyncrasies in the application domain, suppose that data available for analysis amount to a sample of points  $\{y_1, y_2, \dots, y_n\}$  in a subset of  $d$ -dimensional Euclidean space:  $\Omega \subseteq \mathbb{R}^d$ . Our calculations presume the existence of a distribution  $F_*$  on  $\Omega$  from which the  $y_i$ 's are regarded as the realization of a random sample. Rather than further assume that  $F_*$  is constrained to some statistical model, we use modeling considerations somewhat more loosely to guide inference computations, as, for example, in Bayesian Nonparametric Learning (NPL) (e.g., Fong et al., 2019). That is, we require a parameter space  $\Theta \subseteq \mathbb{R}^p$  and loss function  $\tilde{l}(t, y)$  mapping  $\Theta \times \Omega$  into  $\mathbb{R}$ , and we use this loss function to associate with any distribution  $F$  on  $\Omega$  the parameter

$$\theta := \arg \min_{t \in \Theta} \mathcal{L}(t, F) := \arg \min_{t \in \Theta} \int_{\Omega} \tilde{l}(t, y) dF(y). \quad (4.1)$$

The choice of  $\tilde{l}(t, y)$  establishes the parameter  $\theta$  as a functional of the underlying distribution, rather than as an index for a parametric model, which is its role in conventional Bayesian analysis. It is well known, for example, that setting  $\tilde{l}(t, y) = \|y - t\|_2^2$  returns the mean. More generally, setting  $\tilde{l}(t, y)$  to be the negative loglikelihood corresponding to some parametric model  $F_\theta \in \mathcal{F}_\Theta$  leads to  $\theta_* := \arg \min_{t \in \Theta} \mathcal{L}(t, F_*)$ ; this minimizes the Kullback-Leibler divergence  $KL(f_* \| f_\theta)$ , where  $f_*$  and  $f_\theta$  are the respective densities of  $F_*$  and  $F_\theta$ . Notably in this case, we may have a well-defined parameter without assuming that the parametric model has captured (i.e., contained) the data-generating distribution  $F_*$ . Contemporary, high-dimensional examples further warrant inclusion of regularization terms in the loss function:

$$\tilde{l}(t, y) = l(t, y) + \lambda l_0(t)$$

for some tuning parameter  $\lambda > 0$  and penalty function  $l_0(t)$ , which then extends (4.1),

$$\mathcal{L}(t, F) = \int_{\Omega} [l(t, y) + \lambda l_0(t)] dF(y) = \int_{\Omega} l(t, y) dF(y) + \lambda l_0(t).$$

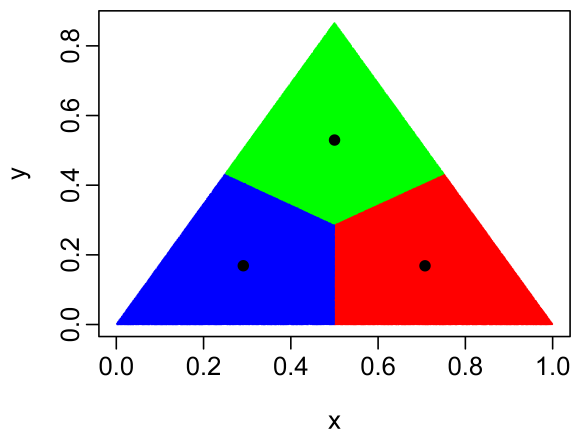


Figure 4.1: K-means clustering for data points which are uniformly distributed on an equilateral triangle with vertices  $\{(0, 0), (1, 0), (0.5, \sqrt{3}/2)\}$ . The black dots represent the centroids obtained by the K-means algorithm specified with  $K = 3$  and the data points are colored according to their respective clusters.

The use of models and loss-functions to guide inference has been exceedingly effective (e.g., Hastie, Tibshirani, & Friedman, 2009), in part because statistical estimation is enabled by plugging the empirical distribution  $F_n$  into (4.1) and leveraging sophisticated optimization tools to solve the minimization problem. Our specific interest here is clustering, which may align with the present framework through, for example, the expected loss:

$$\mathcal{L}(A_K, F) = \int_{\Omega} \min_{a \in A_K} \|y - a\|_2^2 dF(y),$$

where  $A_K = \{a_1, \dots, a_K\}$  contains  $K$  distinct points on  $\mathbb{R}^d$ . Then the  $K$ -means algorithm aims to minimize  $\mathcal{L}(A_K, F_n)$  (Hartigan & Wong, 1979; Pollard, 1981). Figure 4.1 illustrates the functional parameter in a toy example with  $K = 3$ . We note that the functional parameter exists even if the underlying population  $F_*$  is not induced by some *true* clustering mechanism.

In practice, the number of clusters may not be known in advance. Hence, we are interested in mixture-model-related loss functions  $\tilde{l}(t, y)$  that allow the number of clusters to be a data-driven variable (which could also account for cluster-size information), instead of being treated as a fixed parameter  $K$  that has to be pre-specified by the analyst. Furthermore, we are also interested in quantifying uncertainty in the mixture-model functional parameters due to our own uncertainty about the data-generating mechanism  $F_*$ .

#### 4.1.2. Bayesian NPL: posterior sampling and random weighting

The idea to use a Dirichlet process (DP) to express uncertainty in  $F_*$  has been studied extensively (e.g., Müller et al., 2015), and so too have been techniques that allow approximate DP calculations. We are guided here by the Bayesian NPL approach explained in Fong et al. (2019), and the particular Bayesian-bootstrap approximation that follows when the DP prior mass converges to zero, *a posteriori*. Then it is computationally elementary to sample the distribution  $F$  from its posterior given  $\{y_1, y_2, \dots, y_n\}$ ; the computational challenge is in optimizing the expected loss under that  $F$ , which then happens repeatedly, perhaps in parallel, over many posterior draws of  $F$  to produce a posterior sample of functionally-induced

parameters. Specifically, a draw  $F$  from the approximate DP posterior is a distribution supported on the unique sample points, with probability masses that themselves have a finite Dirichlet distribution. This is conveniently achieved with mutually independent standard Exponentially distributed weights  $\mathbf{W} = (W_1, W_2, \dots, W_n)$ . Then the expected loss  $\mathcal{L}(t, F)$  associated with such a posterior-sampled  $F$  is proportional to

$$\mathcal{L}_\lambda(t, \mathbf{W}) := \sum_{i=1}^n W_i l(t, y_i) + \lambda l_0(t). \quad (4.2)$$

There are various ways to handle the regularization weight  $\lambda$ ; for simplicity here we ignore posterior variation this penalty, but other approaches have merit (e.g., Ng & Newton, 2020). We present a detailed derivation in the supplementary material Chapter 5.4 on how we arrive at (4.2) from  $\mathcal{L}(t, F_w)$ , where  $F_w$  represents the Bayesian bootstrap approximation of the DP posterior sampling of  $F_*$ .

In summary, the random-weighting approach amounts to repeated assignment of random weights  $\mathbf{W} = (W_1, W_2, \dots, W_n)$  and minimization in  $t$  of (4.2) to obtain a sample of the functional parameter values,  $\theta$ . Utility of the approach depends in part on the suitability of loss functions  $l(t, y_i)$  and  $l_0(t)$ . The finite mixture case was examined in Fong et al. (2019), who adopted the negative loglikelihood of a finite Gaussian mixture model as the loss function. To eliminate the need to choose the number of clusters and to improve other features, here we examine Bayesian NPL for loss functions developed from certain parameter-limiting calculations within a class of nonparametric models.

### 4.1.3. Working model and small-variance asymptotics

In the search for a suitable loss function  $\tilde{l}(t, y)$ , we first consider the Dirichlet Process Mixture (DPM) model via the Chinese Restaurant Process (CRP) specification (Blackwell



& MacQueen, 1973) as our *working model*:

$$\begin{aligned}
y_i | (z_i = k, \mu_k, \Sigma) &\sim N_d(\mu_k, \Sigma) \\
\mu_k | (\Sigma, \mathbf{z}, \kappa) &\sim N_d(\mu_0, h(\Sigma)) \\
\Sigma &\sim p(\Sigma) \\
(\mathbf{z}, \kappa) &\sim CRP(\alpha_0),
\end{aligned} \tag{4.3}$$

where  $\alpha_0 > 0$  and  $\mu_0 \in \mathbb{R}^d$ . As an example,  $p(\Sigma)$  could be an inverse-Wishart density with  $\nu_0$  degrees of freedom and a symmetric positive-definite scale matrix  $\psi_0$ , whereas  $h(\Sigma) = \Sigma/\xi_0$  for some  $\xi_0 > 0$ . Our working model has a common covariance structure  $\Sigma$  across all mixture components (unless deliberately stated otherwise). Note that the number of clusters is denoted with  $\kappa$  in (4.3) to highlight the fact that it is a random variable to distinguish it from the user-specified  $K$  in the finite-mixture and K-means settings. Both  $\kappa$  and cluster assignments  $\mathbf{z} = \{z_1, \dots, z_n\}$  characterize the partitioning of a DPM.

The notion of *working model* serves as a reminder to the readers that we do not assume that (4.3) is the true sampling distribution  $F_*$ , and (4.3) is also not involved in the Bayesian NPL setup as explained in the preceding subsection. In fact, the joint density of (4.3), which is given by

$$\begin{aligned}
&p(\mathbf{Y}, \mathbf{z}, \kappa, \{\mu_k\}_{k=1}^\kappa, \Sigma) \\
&:= p(\mathbf{Y} | \mathbf{z}, \kappa, \{\mu_k\}_{k=1}^\kappa, \Sigma) \times p(\{\mu_k\}_{k=1}^\kappa | \Sigma, \mathbf{z}, \kappa) \times p(\Sigma) \times p(\mathbf{z}, \kappa) \\
&= (2\pi)^{-\frac{nd}{2}} |\Sigma|^{-\frac{n}{2}} \exp \left\{ -\frac{1}{2} \sum_{k=1}^{\kappa} \sum_{i:z_i=k} (y_i - \mu_k)' \Sigma^{-1} (y_i - \mu_k) \right\} \\
&\times (2\pi)^{-\frac{d\kappa}{2}} |h(\Sigma)|^{-\frac{\kappa}{2}} \exp \left\{ -\frac{1}{2} \sum_{k=1}^{\kappa} (\mu_k - \mu_0)' [h(\Sigma)]^{-1} (\mu_k - \mu_0) \right\} \\
&\times p(\Sigma) \times \alpha_0^{\kappa-1} \frac{\Gamma(\alpha_0 + 1)}{\Gamma(\alpha_0 + n)} \prod_{k=1}^{\kappa} \Gamma(n_k),
\end{aligned} \tag{4.4}$$

merely serves as a guidepost for us in specifying the loss functions  $l(t, y_i)$  and  $l_0(t)$  in (4.2).

We further point out that using Bayesian nonparametric approach as a guiding model is a novel idea in Bayesian NPL literature.

Many maximum-a-posteriori (MAP) procedures for (4.4), such as Dahl (2009) and Raykov, Boukouvalas, and Little (2016), may appear promising at first glance, but they rely on the exchangeability property (Pitman, 1995) of the DPM that breaks down once we introduce random-weights into these loss functions. In fact, optimization of (4.4) by itself may already be daunting; some model-guided simplification to (4.4) is preferred to ensure a simpler, practical working model that could be computed more efficiently.

Consequently, we turn our attention to Kulis and Jordan (2012) who approached (4.3) from the perspective of small-variance-asymptotics (SVA) that resulted in the DP-means objective function:

$$\mathcal{L}_\lambda^{\text{DPmeans}}(\boldsymbol{\mu}, \mathbf{z}, \kappa) := \sum_{k=1}^{\kappa} \sum_{i: z_i=k} \|y_i - \mu_k\|_2^2 + \lambda \kappa. \quad (4.5)$$

In particular, Broderick, Kulis, and Jordan (2013) constructed (4.5) with the SVA setup outlined in Remark 4.1. Under their SVA regime, a diminishing  $\sigma^2$  (indicating a decreasing variance of mixture components which results in more clusters) faces an opposing force of diminishing  $\alpha_0$  (i.e. lower intensity of the CRP to create new clusters), such that a balance is achieved via the regularization parameter  $\lambda$ .

**Remark 4.1** (DP-means as SVA of the DPM (Broderick et al., 2013)). *Consider the DPM model (4.3), where  $\Sigma = \sigma^2 I_d$  and  $h(\Sigma) = c^2 I_d$  for some finite  $c$ . If  $\sigma^2 \rightarrow 0$  and  $\alpha_0 \rightarrow 0$  such that they are modulated with  $\alpha_0 = \exp\{-\lambda/(2\sigma^2)\}$  for some  $\lambda > 0$ , then the negative log of (4.4), multiplied by  $\sigma^2$ , converges to the DP-means objective function (4.5).*

The DP-means setup (4.5) is simple and could be easily adopted for our random-weighting approach, by specifying a (cluster-specific) squared-error loss in (4.2) with a penalty on the number of clusters. However, the DP-means still has some modeling limitations, especially its inability to replicate the **rich-gets-richer** (*rgr*) property of the DPM (e.g., Raykov et al., 2016).

## 4.2. Methodology

### 4.2.1. DP-rich: Alternate asymptotics for the DPM

While the small-variance limit in (4.5) nicely reveals within-cluster sum-of squares and cluster-number features, it has an unintended negative consequence. Namely, it eliminates from the objective function any mechanism to measure the cluster sizes. For the sake of comparison, consider another extreme, where  $\sigma^2 \rightarrow \infty$  instead of shrinking to zero. This would indicate that the data points arise from very “noisy” Normal components/clusters, and data clustering will be completely dictated by the Chinese Restaurant process (CRP), without regard to the distance that points are from centroids. We find it helpful to modulate other working model parameters, and to leave  $\sigma^2$  alone to represent some intrinsic sampling variation. We assume  $\Sigma = \sigma^2 I_d$  and  $h(\Sigma) = \frac{\sigma^2}{\xi_0} I_d$  in (4.3), where  $\sigma^2 = \lambda_2$  for some tuning parameter  $\lambda_2 > 0$  to be calibrated by the analyst.

In addition, notice that  $\alpha_0$  is the CRP intensity parameter while  $\xi_0$  acts as the scaling factor between the variance of mixture components and the prior variance of  $\mu_k$ . Contrary to the SVA setup in Remark 4.1, we further argue that increasing  $\alpha_0 \rightarrow \infty$  and reducing  $\xi_0 \rightarrow 0$  must go hand-in-hand from an Empirical-Bayes perspective: if the variance of mixture components stays rather “constant” (i.e.,  $\sigma^2 = \lambda_2$ ), then larger number of clusters signifies wider data coverage in the Euclidean space. In this case, new centroids must have arisen farther away from  $\mu_0$  in order to establish these new “colonies” or clusters. Hence,  $\alpha_0 \rightarrow \infty$  (indicating higher intensity to create new clusters under the CRP prior) and  $\xi_0 \rightarrow 0$  (suggesting a noisier prior for  $\mu_k$ ) must happen concurrently. Finally, these limiting behaviors of  $\alpha_0$  and  $\xi_0$  are modulated together with  $\lambda_2$  via the relationship

$$\lambda_1 = \lambda_2 \log \left[ \left( \frac{2\pi\lambda_2}{\xi_0} \right)^{\frac{d}{2}} \frac{1}{\alpha_0} \right], \quad (4.6)$$

where  $\lambda_1 > 0$  is another tuning parameter to be calibrated by the analyst (note that this modulating relationship between  $\lambda_1$  and  $\lambda_2$  in (4.6) holds with the limiting behavior of

$\xi_0$  and  $\alpha_0$ ; in practice, we only require the regularization parameters  $\lambda_1, \lambda_2 > 0$ ). These considerations lead to our first main result in Theorem 4.1 – new asymptotics for the DPM which we coin as the **DP-rich** objective function

$$\mathcal{L}_{(\lambda_1, \lambda_2)}^{\text{DP-rich}}(\boldsymbol{\mu}, \boldsymbol{z}, \kappa) := \sum_{k=1}^{\kappa} \sum_{i: z_i=k} \|y_i - \mu_k\|_2^2 + \lambda_1 \cdot \kappa - \lambda_2 \sum_{k=1}^{\kappa} \log [\Gamma(n_k)]. \quad (4.7)$$

**Theorem 4.1 (DP-rich as alternative asymptotics for the DPM).** *Consider the DPM model (4.3), where  $\Sigma = \sigma^2 I_d$  and  $h(\Sigma) = \frac{\sigma^2}{\xi_0} I_d$ . If  $\sigma^2 = \lambda_2$  for some  $\lambda_2 > 0$ ,  $\alpha_0 \rightarrow \infty$  and  $\xi_0 \rightarrow 0$  such that they are modulated via (4.6) for some  $\lambda_1 > 0$ , then the negative log of (4.4), multiplied by  $\sigma^2$ , converges to the **DP-rich** objective function (4.7).*

**Proof of Theorem 4.1.** Given  $\Sigma = \sigma^2 I_d$  and  $h(\Sigma) = \frac{\sigma^2}{\xi_0} I_d$  and  $\sigma^2 = \lambda_2$  for some  $\lambda_2 > 0$ , we have

$$\begin{aligned} -\sigma^2 \log p(\mathbf{Y}, \boldsymbol{z}, \kappa, \boldsymbol{\mu}) &= \frac{1}{2} \left[ \sum_{k=1}^{\kappa} \sum_{i: z_i=k} \|y_i - \mu_k\|_2^2 + \xi_0 \sum_{k=1}^{\kappa} \|\mu_k - \mu_0\|_2^2 \right] \\ &\quad + \kappa \cdot \lambda_2 \cdot \log \left[ \left( \frac{2\pi\lambda_2}{\xi_0} \right)^{d/2} \cdot \frac{1}{\alpha_0} \right] - \lambda_2 \sum_{k=1}^{\kappa} \log [\Gamma(n_k)] \\ &\quad + \frac{nd}{2} \lambda_2 \log(2\pi\lambda_2) - \lambda_2 \log \left[ \frac{\Gamma(\alpha_0 + 1)}{\alpha_0 \Gamma(\alpha_0 + n)} \right]. \end{aligned} \quad (4.8)$$

Notice that we have treated  $\Sigma$  as deterministic in this case and so we dropped the term  $p(\Sigma)$  in (4.4). Next, the third line of (4.8) does not contain  $(\boldsymbol{\mu}, \kappa, \boldsymbol{z})$  and could be dropped. Finally, push  $\alpha_0 \rightarrow \infty$  and  $\xi_0 \rightarrow 0$  such that (4.6) is satisfied, and scale the entire equation by 2 to arrive at (4.7). In particular, we also verify that for any finite  $n \geq 1$ , as  $\alpha_0 \rightarrow \infty$ ,  $\frac{\Gamma(\alpha_0 + 1)}{\alpha_0 \Gamma(\alpha_0 + n)} = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_0 + n)} \rightarrow 1$ .  $\square$

Note that we pick the name “**DP-rich**” to highlight the fact that we are able to retain the rich-gets-richer (*rgr*) property by following an alternative asymptotic argument that differs from its counterpart in Remark 4.1. Clearly, setting  $\lambda_2 = 0$  in (4.7), i.e. switching off the *rgr* regularization in the DP-rich setup, returns the DP-means objective function.

In addition, notice that, from (4.7),  $\lambda_1$  allows direct calibration by the analyst to tune the number of clusters  $\kappa$  obtained by the DP-rich algorithm, whereas  $\lambda_2$  controls the magnitude of the algorithm's *rgr* effect brought about by the term  $\log[\Gamma(n_k)]$ .

#### 4.2.2. DP-rich: optimization and illustration

Similar to the DP-means algorithm, the objective function in (4.7) can be optimized using a block coordinate descent-type algorithm (Tseng, 2001) that alternates between cluster reassignments and centroid updates until the algorithm converges when the cluster assignment for all observations no longer changes.

First, consider the cluster re-assignment step. To reassign the  $i^{th}$  data point, we first hold all the cluster parameters and cluster labels of all other observations constant. Then we reassign this  $i^{th}$  observation to (either an existing or a new) cluster that contributes the least to the increment of the objective, i.e. minimizing the cost to pay for assigning this observation. Specifically, an observation  $y_i$  is either assigned to an existing cluster  $\mathcal{C}_k$  for  $k \in \{1, \dots, \kappa\}$  or allocated into a new cluster  $\mathcal{C}_{\kappa+1}$ , by comparing its "cost" of joining an existing cluster  $\mathcal{C}_k$

$$d_{ik} = \|y_i - \mu_k\|_2^2 - \lambda_2 \log(n_{k,-i}) \quad (4.9)$$

for  $k = 1, \dots, \kappa$ , as well as its "cost" to create a new cluster  $\mathcal{C}_{\kappa+1}$

$$d_{i,\kappa+1} = \lambda_1. \quad (4.10)$$

The term  $n_{k,-i}$  in (4.9) denotes the number of observations in cluster  $\mathcal{C}_k$  excluding the current  $i^{th}$  observation, i.e. if  $i \in \mathcal{C}_{k'}$ , then  $n_{k',-i} = n_{k'} - 1$  and  $n_{k,-i} = n_k$  for  $k \in \{1, \dots, \kappa\} \setminus \{k'\}$ . From (4.9), it is evident that the allocation of an observation  $y_i$  into an existing cluster  $\mathcal{C}_k$  is affected by two opposing forces, namely the squared Euclidean distance from the cluster centroid  $\mu_k$ , which is discounted by  $\log(n_{k,-i})$  with a factor of  $\lambda_2$ . The term  $\log(n_{k,-i})$  can be viewed as the "gravitational mass" of the cluster  $\mathcal{C}_k$  that "pulls" or "attracts" the data

point  $y_i$ .

---

**Algorithm 3** DP-rich
 

---

**Require:** data  $\{y_1, \dots, y_n\}$ , regularization parameters  $\lambda_1$  and  $\lambda_2$

- 1: Initialize by assigning all observations into a single cluster, and initialize  $\mu_1$  as the grand centroid.
- 2: **while** not all  $z_i^{old} = z_i$  **do**
- 3:    $z_i^{old} \leftarrow z_i$  for all  $i$ .
- 4:   **for** each data point  $y_i$  **do**
- 5:     Compute  $d_{ik}$  with (4.9) for  $k = 1, \dots, \kappa$ .
- 6:     If  $\min_{1 \leq k \leq \kappa} d_{ik} > \lambda_1$ , set  $\kappa = \kappa + 1$ ,  $z_i = \kappa$  and  $\mu_\kappa = y_i$ . Otherwise, set  $z_i = \arg \min_{1 \leq k \leq \kappa} d_{ik}$ .
- 7:     Drop empty clusters if they exist.
- 8:   **end for**
- 9:   For each cluster  $k$ , update its cluster centroid  $\mu_k$  as the average of observations allocated to the cluster.
- 10: **end while**

**Ensure:** Number of clusters  $\kappa$ , cluster centroids  $\{\mu_k\}_{1 \leq k \leq \kappa}$ , and cluster assignments  $\{z_i\}_{1 \leq i \leq n}$ .

---

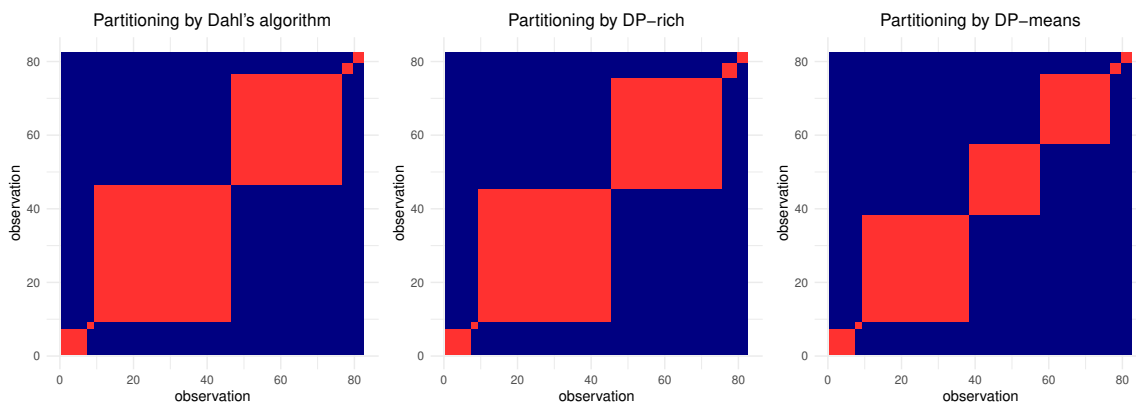


Figure 4.2: Cluster partitions obtained by Dahl (2009)'s algorithm, DP-rich and DP-means approaches for the 1-dimensional Galaxy data set (Roeder, 1990), where red color indicates that the pair of observations is clustered together and navy-blue color otherwise. The observations in the data set are arranged in ascending order.

After re-assigning all the observations, we move on to the centroid updates. Conditional on the existing partition  $(\kappa, \mathbf{z})$ , the centroid  $\mu_k$  is updated as the average of  $\{y_i : i \in \mathcal{C}_k\}$  for  $k = 1, \dots, \kappa$ . Algorithm 3 outlines this DP-rich procedure in detail, while Lemma 4.2

ensures local convergence of Algorithm 3. We refer readers to the supplementary material Chapter 5.1 for other implementation details of the algorithm.

In particular, we would like to point out that, while the K-means procedure is influenced by the choices of initial cluster centroids (Arthur & Vassilvitskii, 2007), the DP-rich procedure depends on the order in which data points are processed (i.e., the order in which  $y_i : i \in \{1, \dots, n\}$  is processed in the for-loop (lines 4–8) of Algorithm 3). This feature is also shared by the DP-means algorithm (Kulis & Jordan, 2012), because both DP-rich and DP-means algorithms involve inserting new cluster(s) and/or deleting empty cluster(s) during their cluster reassignment steps. To mitigate the problem of sub-optimal local solution, we follow Kulis and Jordan (2012)’s suggestion to repeat the algorithm several times in which we process the data points with different randomly-permuted order, and pick the set of solutions with the smallest objective.

**Lemma 4.2 (Local Convergence of DP-rich).** *Algorithm 3 monotonically decreases the DP-rich objective function (4.7) until local convergence is achieved.*

**Proof of Lemma 4.2.** The proof follows a similar argument as the proof for Kulis and Jordan (2012)’s Theorem 3.1, except that the reassignment step now depends on a squared Euclidean distance discounted by  $\lambda_2 \log(n_{k,-i})$ .  $\square$

As an example, we use the galaxy benchmark data set (Roeder, 1990) to illustrate that the DP-rich approach has an advantage over the DP-means approach (Kulis & Jordan, 2012) in capturing the *rich-gets-richer* (*rgr*) property brought about by the DPM. Briefly, this benchmark data set contains physical information on velocities for 82 galaxies drawn from six well-separated conic sections of the Corona Borealis region (i.e.,  $n = 82$ ,  $d = 1$  and  $K_{true} = 6$ ). We compare these two methods with Dahl (2009)’s algorithm which is guaranteed to find the MAP clustering for 1-dimensional data if the underlying sampling distribution is (4.3) with known mixture-component variance  $\sigma_y^2$  and centroids’ prior variance  $h(\sigma^2) = \sigma_\mu^2$ . Specifically, we specify the priors  $\mu_0$ ,  $\sigma_y^2$  and  $\sigma_\mu^2$  via Empirical Bayes, i.e., these priors are estimated using cluster parameters obtained from a K-means implementation with  $K = 6$ .

We also fix  $\alpha_0 = 1.3$  such that the CRP prior mean of  $\kappa$  is approximately 6. For DP-rich, we specify  $\lambda_2$  to be the estimated  $\sigma_y^2$ . For meaningful comparison, we fix  $\lambda_1 = 5$  for both DP-rich and DP-means. We repeat both DP-rich and DP-means algorithms 20 times and we pick the solutions with the lowest objectives. Figure 4.2 illustrates the partitions obtained by these three methods. We see that at  $\lambda_1 = 5$ , DP-rich obtains 6 clusters for the data points whereas DP-means has 7 clusters. The presence of the *rgr* regularization in DP-rich attracts the data points (that would otherwise fall into two separate clusters under DP-means) into one combined cluster. From Figure 4.2, it is evident that the partition obtained by DP-rich is more “similar” to that of Dahl (2009)’s algorithm. Using the partition obtained by Dahl (2009)’s algorithm as benchmark, the Normalized Mutual Information (NMI) (Vinh, Epps, & Bailey, 2010) for DP-rich is 0.916, whereas the NMI for DP-means in this case is 0.700.

### 4.2.3. Main model: Random-Weighting Scaled DP-rich

Now that we have a suitable loss function  $\tilde{l}(t, y) = l(t, y) + \lambda \cdot l_\lambda(t)$  in the form of DP-rich (4.7), we introduce the objective function  $\mathcal{L}_\lambda(t, \mathbf{W})$  in (4.2) for our main random-weighting countable-mixture model, which we coin as the **random-weighting scaled DP-rich (RW SDP-rich)** approach:

$$\begin{aligned} & \mathcal{L}_{(\lambda_1, \lambda_2)}^{\text{rwSDP-rich}}(\mathbf{z}, \kappa, \boldsymbol{\mu}, \Sigma) \\ & := \frac{1}{2} \left[ \sum_{k=1}^{\kappa} \sum_{i: z_i=k} W_i (y_i - \mu_k)' \Sigma^{-1} (y_i - \mu_k) + \xi_0 \sum_{k=1}^{\kappa} (\mu_k - \mu_0)' \Sigma^{-1} (\mu_k - \mu_0) + \text{Tr}(\psi_0 \Sigma^{-1}) \right] \\ & + \left( \sum_{i=1}^n W_i + \nu_0 - d - 1 \right) \log |\Sigma^{1/2}| + \lambda_1 \cdot \kappa - \lambda_2 \sum_{k=1}^{\kappa} \log [\Gamma(n_k)], \end{aligned} \quad (4.11)$$

where the prior components  $\mu_0 \in \mathbb{R}^d$  and  $\xi_0 > 0$ ,  $\nu_0 > d + 1$  and the symmetric positive definite matrix  $\psi_0$  are specified in (4.3),  $W_i \stackrel{iid}{\sim} \text{Exp}(1)$ , and  $\lambda_1, \lambda_2 > 0$  are the tuning/regularization parameters to be supplied by the analyst. Similar to the RW DP-means, the couplet  $(\kappa, \mathbf{z})$  characterize the partition obtained by the RW SDP-rich model.

Specifically, we adopt the random-weighting framework on an extended version of



the DP-rich model to arrive at (4.11). Besides retaining the tuning parameters  $\lambda_1$  and  $\lambda_2$  (that allow direct calibration of  $\kappa$  and the magnitude of the *rgr* effect respectively), we incorporate a common covariance term  $\Sigma$  into the objective function (to be optimized with other parameters) that enables the RW SDP-rich approach to capture potential non-spherical nature (correlation and different scaling among features or dimensions) of the data. In fact, the RW SDP-rich objective function (4.11) is obtained by *modifying* (4.4); see supplementary material Chapter 5.2 for more details about the modification.

#### 4.2.4. RW SDP-rich: optimization

We repeatedly assign i.i.d. standard Exponential weights  $\{W_i\}_{1 \leq i \leq n}$  and optimize (4.11) for  $B$  times to obtain  $B$  random-weighting samples. For any given set of the i.i.d.  $(W_1, \dots, W_n)$ , the objective function in (4.11) can be optimized using an algorithm that is similar to Algorithm 3. In particular, the “cost” of the  $i^{\text{th}}$  data point joining an existing cluster  $\mathcal{C}_k$  is updated as

$$d_{ik}^w = \frac{1}{2} W_i (y_i - \mu_k^w)' \Sigma_w^{-1} (y_i - \mu_k^w) - \lambda_2 \log(n_{k,-i}) \quad (4.12)$$

for  $k = 1, \dots, \kappa$ , while the “cost” to create a new cluster  $\mathcal{C}_{\kappa+1}$  is given by

$$d_{i,\kappa+1}^w = \frac{1}{2} \frac{\xi_0 W_i}{\xi_0 + W_i} (y_i - \mu_0)' \Sigma_w^{-1} (y_i - \mu_0) + \lambda_1. \quad (4.13)$$

For cluster-parameter updates, conditional on the existing partition  $(\kappa, \mathbf{z})$ , the cluster-specific centroids are updated as

$$\mu_k^w = \frac{\sum_{i:z_i=k} W_i y_i + \xi_0 \mu_0}{\sum_{i:z_i=k} W_i + \xi_0} \quad (4.14)$$

for  $k = 1, \dots, \kappa$ , and the common (across all  $\kappa$  clusters) covariance term is updated as

$$\Sigma_w = \frac{\sum_{k=1}^{\kappa} \sum_{i:z_i=k} W_i (y_i - \mu_k) (y_i - \mu_k)' + \xi_0 \sum_{k=1}^{\kappa} (\mu_k - \mu_0) (\mu_k - \mu_0)' + \psi_0}{(\sum_{i=1}^n W_i + \nu_0) - d - 1}. \quad (4.15)$$

These parameter updates (4.14) and (4.15) enable incorporation of prior information in (4.3), which will be superseded by data information as sample size increases.

---

**Algorithm 4** Random-weighting Scaled DP-rich (RW SDP-rich)

---

**Require:** data  $\{y_1, \dots, y_n\}$ , regularization parameters  $\lambda_1$  and  $\lambda_2$ , prior terms  $\{\mu_0, \xi_0, \nu_0, \psi_0\}$ , and number of posterior draws  $B$

- 1: **for**  $b = 1, \dots, B$  **do**
- 2:     Draw  $W_i \stackrel{iid}{\sim} Exp(1) \forall i = 1, \dots, n$ .
- 3:     Initialize by assigning all observations into a single cluster. In addition, initialize  $\Sigma_b^w = \psi_0 / (\nu_0 - d - 1)$ .
- 4:     **while true do**
- 5:          $z_{i,b}^{w,old} \leftarrow z_{i,b}^w$  for all  $i$ .
- 6:         **for** each data point  $y_i$  **do**
- 7:             Compute  $d_{ik}^w$  with (4.12) for  $k = 1, \dots, \kappa_b^w$ , and compute  $d_{i,\kappa_b^w+1}^w$  with (4.13).
- 8:             If  $\min_{1 \leq k \leq \kappa_b^w} d_{ik}^w > d_{i,\kappa_b^w+1}^w$ , set  $\kappa_b^w = \kappa_b^w + 1$ ,  $z_{i,b}^w = \kappa_b^w$  and initialize  $\mu_{\kappa_b^w,b}^w$  with (4.14). Otherwise, set  $z_{i,b}^w = \arg \min_{1 \leq k \leq \kappa_b^w} d_{ik}^w$ .
- 9:             Drop empty clusters if they exist.
- 10:         **end for**
- 11:         For each cluster  $k$ , update its cluster centroid  $\mu_{k,b}^w$  with (4.14).
- 12:         Update  $\Sigma_b^w$  with (4.15).
- 13:         **if**  $z_{i,b}^{w,old} = z_{i,b}^w$  for all  $i$  **then**
- 14:             Store  $\kappa_b^w, \Sigma_b^w, \mu_{k,b}^w$  for  $k = 1, \dots, \kappa_b^w$  and  $z_{i,b}^w$  for  $i = 1, \dots, n$ .
- 15:         **end if**
- 16:     **end while**
- 17: **end for**

**Ensure:**  $B$  samples of number of clusters  $\{\kappa_b^w\}_{1 \leq b \leq B}$ , covariance term  $\{\Sigma_b^w\}_{1 \leq b \leq B}$ , cluster centroids  $\{\mu_{k,b}^w\}_{1 \leq k \leq \kappa_b^w; 1 \leq b \leq B}$ , and cluster assignments  $\{z_{i,b}^w\}_{1 \leq i \leq n; 1 \leq b \leq B}$ .

---

Algorithm 4 outlines this random-weighting procedure in detail. Notice that this algorithm is trivially parallelizable over  $b \in \{1, \dots, B\}$ , which enhances its scalability to large datasets. We refer readers to the supplementary material Chapter 5.2 for other implementation details of the algorithm.

**Lemma 4.3. (Local Convergence of RW SDP-rich)** *For any given sets of positive weights  $(W_1, \dots, W_n)$ , the while-loop (lines 4–16) of Algorithm 4 monotonically decreases the objective given in (4.11) until local convergence.*

Lemma 4.3 ensures local convergence of the RW SDP-rich algorithm. Its proof is given in

the supplementary material Chapter 5.2. Similar to the DP-rich algorithm, the RW SDP-rich procedure also depends on the order in which data points are processed (i.e., the order in which  $y_i : i \in \{1, \dots, n\}$  is processed in the for-loop (lines 6–10) of Algorithm 4). Again, we suggest that for each set of random weights  $(W_1, \dots, W_n)$ , we repeat the while-loop (lines 4–16) of Algorithm 4 several times in which we process the data points with different permuted order, and pick the set of solutions with the smallest objective.

#### 4.2.5. RW SDP-rich: related models

There are several variations (or simplifications) to the RW SDP-rich model, which could be useful in different situations. Figure 4.3 summarizes these variations of the random-weighting procedures.

##### Diagonal Covariance Structure

For high-dimensional datasets with high correlation among features, the scalability and chain-mixing problems of standard MCMC procedures become more prominent. The analyst may choose to first apply some dimension-reduction tools (e.g. Scrucca et al., 2016), such as the Principal Component Analysis (PCA) on the data points  $\{y_1, \dots, y_n\}$ , or the Multidimensional Scaling (MDS) approach on the pairwise distances of the data points (e.g. Hastie et al., 2009), and then perform clustering on these principal components (PCs) or eigenvectors from the MDS. Since these dimensionally-reduced datasets are uncorrelated by construction, the analyst could then apply the RW SDP-rich model with a diagonal covariance structure instead:

$$\begin{aligned} & \mathcal{L}_{(\lambda_1, \lambda_2)}^{\text{rwSDP-rich}}(\mathbf{z}, \kappa, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \\ & := \sum_{j=1}^d \frac{1}{2\sigma_j^2} \left[ \sum_{k=1}^{\kappa} \sum_{i: z_i=k} W_i (y_{ij} - \mu_{kj})^2 + \xi_{0j} \sum_{k=1}^{\kappa} (\mu_{kj} - \mu_{0j})^2 + 2b_{0j} \right] \\ & + \frac{1}{2} \sum_{j=1}^d \left( \sum_{i=1}^n W_i + 2a_{0j} - 2 \right) \log(\sigma_j^2) + \lambda_1 \cdot \kappa - \lambda_2 \sum_{k=1}^{\kappa} \log[\Gamma(n_k)], \end{aligned} \quad (4.16)$$

where  $W_i \stackrel{iid}{\sim} Exp(1)$  and  $b_{0,j}, \xi_{0,j} > 0$  and  $a_{0,j} > 1$  for all  $j = 1, \dots, d$ . In fact, this objective function (4.16) is derived by *modifying* (4.4) where a common *diagonal* covariance structure is adopted, i.e.  $\Sigma = diag(\sigma_1^2, \dots, \sigma_d^2)$ ,  $h(\Sigma) = diag\left(\frac{\sigma_1^2}{\xi_{0,1}}, \dots, \frac{\sigma_d^2}{\xi_{0,d}}\right)$ , and  $\sigma_j^2 \sim IG(a_{0,j}, b_{0,j})$  for  $j = 1, \dots, d$ . Similar procedure (Algorithm 4) could be used to optimize (4.16), with slightly different formulae for parameter updates and costs of contribution to the objective. We refer readers to the supplementary material Chapter 5.2 for these formulae. Notice that this algorithm runs faster than its full-covariance counterpart since matrix inversion of cluster covariance  $\Sigma$  is avoided in this case.

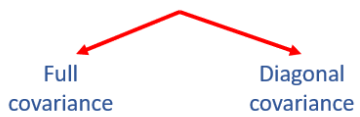
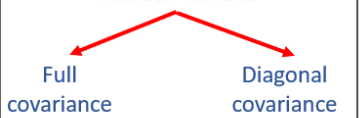
		'rich-gets-richer' (rgr) property	
		Yes ( $\lambda_2 > 0$ )	No ( $\lambda_2 = 0$ )
Feature Scaling	Yes	<p style="text-align: center;"><b>RW SDP-rich</b></p>  <p style="text-align: center;">Full covariance      Diagonal covariance</p>	<p style="text-align: center;"><b>RW SDP-means</b></p>  <p style="text-align: center;">Full covariance      Diagonal covariance</p>
	No ( $\Sigma = I_d$ and $\xi_0 = 0$ )	<b>RW DP-rich</b>	<b>RW DP-means</b>

Figure 4.3: Schematic depicting different variations of the random-weighting models.

### No rich-gets-richer (rgr) property

If the analyst decides that the “rich-gets-richer” (rgr) property does not reflect the underlying data generating processes (e.g. Jensen & Liu, 2008), then the *rgr* penalty in the RW SDP-rich could be discarded by setting  $\lambda_2 = 0$  in (4.11) or (4.16). To distinguish its lack of *rgr* property, we name this procedure as **random-weighting scaled DP-means (RW SDP-means)**.

### No feature scaling

If the underlying true sampling distribution is indeed a mixture model where each cluster of observations has uncorrelated features (dimensions) with unit variance, then the analyst may choose to simplify the random-weighting procedures by setting  $\Sigma = I_d$  in (4.3). In

addition, the analyst may also choose to specify  $h(\Sigma) = I_d/\xi_0$ , where  $\xi_0 \rightarrow 0$  signifying a very noisy or uninformative prior for  $\mu_k$ . In this case, the objective functions (4.11) and (4.16) could be simplified into the **random-weighting DP-rich (RW DP-rich)** procedure

$$\mathcal{L}_{(\lambda_1, \lambda_2)}^{\text{rwDP-rich}}(\mathbf{z}, \kappa, \boldsymbol{\mu}) := \sum_{k=1}^{\kappa} \sum_{i: z_i=k} W_i \|y_i - \mu_k\|_2^2 + \lambda_1 \cdot \kappa - \lambda_2 \sum_{k=1}^{\kappa} \log[\Gamma(n_k)]. \quad (4.17)$$

(4.17) is indeed the random-weighting version of (4.7). Again, Algorithm 4 could be used to optimize (4.17), except that the weighted squared Mahalanobis distance (with the  $1/2$  factor) in the formulae would be replaced with weighted squared Euclidean distance (without the  $1/2$  factor). Another notable difference is that no optimization w.r.t.  $\Sigma$  is required here for (4.17), thus leading to faster computation as compared to the RW SDP-rich setup. Note that setting  $\lambda_2 = 0$  in (4.17) reduces the objective function to the **random-weighting DP-means (RW DP-means)**:

$$\mathcal{L}_{\lambda_1}^{\text{rwDPmeans}}(\mathbf{z}, \kappa, \boldsymbol{\mu}) := \sum_{k=1}^{\kappa} \sum_{i: z_i=k} W_i \|y_i - \mu_k\|_2^2 + \lambda_1 \cdot \kappa. \quad (4.18)$$

Again, (4.18) is the random-weighting version of (4.5).

Finally, if the analyst pre-specify a fixed number of clusters  $K$  instead of letting  $\kappa$  to be a data-driven parameter, then the RW DP-means setup is reduced to the **random-weighting K-means (RW K-means)** algorithm:

$$\mathcal{L}_K^{\text{rwKmeans}}(\boldsymbol{\mu}, \mathbf{z}) := \sum_{k=1}^K \sum_{i: z_i=k} W_i \|y_i - \mu_k\|_2^2. \quad (4.19)$$

We refer readers to the supplementary material Chapter 5.3 for an algorithm to deploy RW K-means, as well as a simple proof about how RW K-means serves as the small-variance asymptotics of Fong et al. (2019)'s random-weighting Gaussian finite-mixture model.

#### 4.2.6. RW SDP-rich: computational complexity

For every set of random weights  $(W_1, \dots, W_n)$ , the random-weighting algorithms are either repeated until local convergence where cluster assignments no longer change, or capped at  $t_{max}$  times, whichever is achieved earlier.

**RW K-means.** Thus, the computational complexity for RW K-means is at most

$$\mathcal{O}(B \cdot t_{max} \cdot K \cdot n \cdot d),$$

where  $B$  denotes the number of posterior draws and  $K$  denotes the number of clusters specified by the analyst. The factor  $n \cdot d$  results from the squared Euclidean distance computed for every data point in the cluster reassignment step.

**RW DP-rich.** Similarly, the order of complexity for RW DP-rich is given by

$$\mathcal{O}(B \cdot t_{max} \cdot \bar{\kappa}_{\text{rwDP-rich}} \cdot n \cdot d),$$

where  $\bar{\kappa}_{\text{rwDP-rich}}$  denotes the average estimated number of clusters by the RW DP-rich algorithm. (See, for example, Paul and Das (2020) on how they accounted for the computational complexity of their algorithm which extends the DP-means approach.)

**RW SDP-rich (full covariance structure).** Meanwhile, for the RW Scaled DP-rich approach (4.11), the order of complexity is given by

$$\mathcal{O}(B \cdot t_{max} \cdot [\bar{\kappa}_{\text{rwSDP-rich (full)}} \cdot n \cdot d^2 + d^3]),$$

where  $\bar{\kappa}_{\text{rwSDP-rich (full)}}$  denotes the average estimated number of clusters by the RW SDP-rich algorithm under the full covariance structure. The factor  $n \cdot d^2$  results from the squared Mahalanobis distance computed for every data point in the cluster reassignment step, whereas the  $d^3$  factor results from the inversion of the common (across all clusters) covariance term  $\Sigma_w$ .

**RW SDP-rich (diagonal covariance structure).** Since the RW Scaled DP-rich approach

with diagonal covariance structure (4.16) does not involve calculation of Mahalanobis distance or inversion of covariance matrix, its computational complexity is reduced to  $\mathcal{O}(B \cdot t_{\max} \cdot \bar{\kappa}_{\text{rwSDP-rich (diag)}} \cdot n \cdot d)$ , where  $\bar{\kappa}_{\text{rwSDP-rich (diag)}}$  denotes the average estimated number of clusters by the algorithm.

#### 4.2.7. RW SDP-rich: calibrating regularization parameters

We first focus on the *rich-gets-richer* (*rgr*) tuning parameter  $\lambda_2$  for both RW SDP-rich and RW DP-rich approaches. Based on the construction of DP-rich in Section 4.1 as well as Equation (4.12), we propose to specify  $\lambda_2^{\text{rwSDP-rich}} = \frac{1}{2}$  and  $\lambda_2^{\text{rwDP-rich}} = \hat{\sigma}^2$ , where  $\hat{\sigma}^2$  is the analyst's estimate about the variance in each feature of the data points in the same cluster. Then, we have

$$\begin{aligned} d_{ik}^{\text{rwSDP-rich}} &= \frac{1}{2} [W_i (y_i - \mu_k^w)' \Sigma_w^{-1} (y_i - \mu_k^w) - \log(n_{k,-i})] \\ d_{ik}^{\text{rwDP-rich}} &= W_i \|y_i - \mu_k^w\|_2^2 - \hat{\sigma}^2 \log(n_{k,-i}). \end{aligned} \quad (4.20)$$

Notice that if  $\Sigma_w = I_d$  and  $\hat{\sigma}^2 = 1$  in (4.20), then the weighted squared Mahalanobis distance or weighted squared Euclidean distance of a data point  $y_i$  from a centroid  $\mu_k$  is “discounted” by the same factor of  $\log(n_{k,-i})$ . We find that these choices of  $\lambda_{n,2}$  lead to reasonable performance by the algorithms in our numerical experiments. We illustrate via a simulation example in the supplementary material Chapter 5.2 to compare the performance of these two approaches using different *rgr* tuning parameters.

From (4.20), it is evident that under the RW DP-rich approach, the scales of the data directly affect the ratio between the squared Euclidean distance and  $\log(n_{k,-i})$ . Thus, the onus is on the analyst to estimate  $\hat{\sigma}^2$ , or the RW SDP-rich approach should be adopted instead, because the issue of non-unitary feature-scales is already taken into consideration by the RW SDP-rich algorithm via the variable  $\Sigma_w$ .

After determining  $\lambda_2$ , we now turn our attention to the tuning parameter  $\lambda_1$  that directly regulates  $\kappa$  for all the random-weighting approaches. We opine that calibration of  $\lambda_1$  depends on the purpose of the analyst's clustering exercise. Here are some examples of

benchmark measurements that may be considered by the analyst:

- The analyst may wish to tune  $\lambda_1$  such that the average of  $\{\kappa_b^w\}_{1 \leq b \leq B}$  mimics the MAP estimate of  $\kappa$ . There are numerous existing approximate methods to obtain MAP estimates for the DPM (without full MCMC procedure); see, for example, Zuanetti et al. (2019), Karabatsos (2020) and references therein.
- The analyst may be interested to compare the clustering patterns obtained from the random-weighting methods against other clustering procedure (such as agglomerative hierarchical clustering). Then, the analyst may choose to calibrate  $\lambda_1$  to maximize the average of some similarity measures (such as Normalized Mutual Information (NMI) or Adjusted Rand Index (ARI) (Vinh et al., 2010)) comparing the random-weighting partitions and the “benchmark” partition by the other clustering method.
- Other potential consideration could also be the notion of stability selection; see, for example, Fang and Wang (2012) as well as Paul and Das (2020).

We refer readers to the supplementary material Chapter 5.2 for a detailed algorithm that outlines the specific steps for calibrating  $\lambda_1$  after the analyst has decided on the benchmark measurement to be used for tuning this regularization parameter.

### 4.3. Numerical Experiments

We study performances of the random-weighting (RW) procedures, namely RW DP-rich (where RW DP-means is a special case) and RW SDP-rich (where RW SDP-means is a special case), and compare them with standard MCMC methods for the DPM of Normals and Blei and Jordan (2006)’s variational inference (henceforth abbreviated as VI). All simulation studies and data analyses are performed using R (R Core Team, 2019); the source code is available at the Github public repository <https://github.com/ngtunlee/random-weighting-mixture>. In particular, we accelerate the random-weighting and VI



algorithms in R with C++ implementations via `RcppArmadillo` (Eddelbuettel & Sanderson, 2014). In addition,

- standard MCMC procedure for the DPM of Normals with full covariance structure (i.e.,  $h(\Sigma_k) = \Sigma_k/\xi_0$  and  $\Sigma_k|\mathbf{z}, \kappa \sim IW(\nu_0, \psi_0)$  in (4.3)) is implemented with R package `DPpackage` (Jara, Hanson, Quintana, Mueller, & Rosner, 2011), and is compared with its variational inference counterpart (formulae provided in the supplementary material Chapter 5.6) as well as RW SDP-rich in (4.11).
- standard MCMC procedure for the DPM of Normals with diagonal covariance structure, (i.e.,  $\Sigma_k = \text{diag}(\sigma_{k,1}^2, \dots, \sigma_{k,d}^2)$ ,  $h(\Sigma_k) = \text{diag}(\frac{\sigma_{k,1}^2}{\xi_{0,1}}, \dots, \frac{\sigma_{k,d}^2}{\xi_{0,d}})$ , and  $\sigma_{k,j}^2|\mathbf{z}, \kappa \sim IG(a_{0,j}, b_{0,j})$  for  $j = 1, \dots, d$  in (4.3)), is implemented with R package `BNPmix` (Corradin, Canale, & Nipoti, 2021), and is compared with its variational inference counterpart (see supplementary material Chapter 5.6) as well as RW SDP-rich in (4.16). Again, note that  $\xi_0$  is a  $d$ -dimensional vector here, whereas  $\xi_0$  under the full covariance structure is a scalar.

Notice that, even though our DPM working model in (4.3) considers a common covariance term across the mixture components, the aforementioned existing software packages implement standard MCMC schemes that involve a more general form of DPM that allows cluster-specific covariance terms.

In order to facilitate meaningful comparison between MCMC posterior samples and surrogate samples from all the other approximate methods (VI and random-weighting) in all our numerical experiments, the same set of prior values are adopted across MCMC, VI and RW SDP-rich (where applicable), and we also calibrate the tuning parameter  $\lambda_1$  for all random-weighting methods to mimic the posterior mean of  $\kappa$  obtained by MCMC. Again, we fix  $\lambda_2^{\text{rwSDP-rich}} = \frac{1}{2}$  in each simulation and data analysis. The mixing of the MCMC chains is assessed with the trace plots of the posterior number of clusters sampled by MCMC. Each of the random-weighting (and VI) algorithms is repeated 5 times, and we pick the solution with the lowest objective (or, for VI, the highest evidence lower bound (ELBO)); see

supplementary material Chapter 5.6 for more details). Computational times for all these methods in all of our numerical experiments are provided in the supplementary material Chapter 5.5.

### 4.3.1. Simulations

We consider 3 simulation settings as explained below. For each simulation setting, we generate  $T = 10$  independent data sets. Each of these simulated data sets consists of  $n = 1000$  training samples and  $m = 500$  held-out (test) samples.

**Simulation Setting I.** We generate data from a 2-dimensional Gaussian finite-mixture model with  $K_{true} = 16$  and  $\Sigma_{true} = I_2$ . Each cluster has (almost) equal number of data points. The true centroids are equally spaced-out among  $(x, y)$ -coordinates  $\in \{-6, -2, 2, 6\}$ . For this simulation setting, we adopt the diagonal covariance structure on all the aforementioned methods (MCMC, VI and random-weighting). For  $j = 1, 2$ , the priors are specified to be:  $\mu_{0,j} = 0$ ,  $\xi_{0,j} = 0.1$ ,  $a_{0,j} = 2$  and  $b_{0,j} = 1$ , such that the inverse gamma priors have a mean of 1. We also specify  $\alpha_0 = 2.6$  such that the prior mean of  $\kappa$  under the CRP is approximately 16. Corresponding to the diagonals of  $\Sigma_{true}$ , we specify  $\lambda_2^{rwDP-rich} = 1$ .

**Simulation Setting II.** The simulation setting is similar to Simulation Setting I, except that now  $\Sigma_{true} = \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix}$ , so that we could compare the performances of the methods when the features/dimensions of the data are more highly-correlated. For this simulation setting, we adopt the full-covariance structure on all the aforementioned methods. Again, the same priors are specified, except that the inverse gamma priors are replaced with the inverse Wishart prior where  $\nu_0 = 5$  and  $\psi_0 = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ , such that the inverse Wishart prior has a mean that equals  $\Sigma_{true}$ . Again, corresponding to the diagonals of  $\Sigma_{true}$ , we specify  $\lambda_2^{rwDP-rich} = 1$ .

**Simulation Setting III.** We generate data from a 2-dimensional DPM of Normals with a CRP intensity parameter  $\alpha_0 = 2.6$ . The mixture component variance is fixed at  $\Sigma_{true} = \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix}$ , whereas the mixture component centroids are sampled from a Normal prior with parameters  $\mu_0 = (0, 0)'$  and  $\xi_0 = 0.1$ . Again, the full-covariance structure is adopted on all the aforementioned methods. Ground-truth prior values which are used to generate data

are also adopted for MCMC, VI, RW SDP-means and RW SDP-rich. Again, corresponding to the diagonals of  $\Sigma_{true}$ , we specify  $\lambda_2^{rwDP-rich} = 1$ .

For each simulated data set, we draw  $B = 5000$  posterior (or approximate posterior) samples for each of the aforementioned methods. For MCMC, we specify a burn-in period of 5000 and a thinning interval of 15 to reduce auto-correlation among posterior samples. For variational inference, we fix the stick-breaking threshold at  $K_{max} = 40$  for Simulation Settings I and II, and  $K_{max} = 60$  for Simulation Settings III. The MCMC and random-weighting implementations produce

$$\left\{ \kappa_{(MCMC)}^{(b,t)}, \kappa_{(rwDPmeans)}^{(b,t)}, \kappa_{(rwDP-rich)}^{(b,t)}, \kappa_{(rwSDPmeans)}^{(b,t)}, \kappa_{(rwSDP-rich)}^{(b,t)} \right\}$$

and

$$\left\{ z_{i(MCMC)}^{(b,t)}, z_{i(rwDPmeans)}^{(b,t)}, z_{i(rwDP-rich)}^{(b,t)}, z_{i(rwSDPmeans)}^{(b,t)}, z_{i(rwSDP-rich)}^{(b,t)} \right\}_{1 \leq i \leq n},$$

which represent the sampled/bootstrapped  $\kappa$ 's and cluster assignment for the  $i^{th}$  observation in the  $b^{th}$  iteration (i.e.  $b^{th}$  posterior draw) for the  $t^{th}$  simulated data set.

Meanwhile, the variational inference (VI) algorithm produces local solution to “variational parameters” of the “variational densities”. (Again, see supplementary material Chapter 5.6 for more detailed formulae.) In particular, the CRP prior of (4.3) could be reformulated as a stick-breaking prior (Sethuraman, 1994): for  $i = 1, \dots, n$ ,

$$z_i | \pi(\mathbf{v}) \sim Mult(1; \pi(\mathbf{v})) \quad \text{where} \quad \pi(\mathbf{v}) | \alpha_0 \sim GEM(\alpha_0). \quad (4.21)$$

The VI method approximates the multinomial components in (4.21) with variational multinomial probabilities  $\{\hat{\pi}_{i,k}\}_{1 \leq i \leq n, 1 \leq k \leq K_{max}}$ . We then draw  $B$  surrogate samples of cluster assignments based on these VI multinomial probabilities, i.e. for every  $i^{th}$  training data

point in the  $t^{th}$  simulated data set, we sample independently

$$z_{i(\text{VI})}^{(b,t)} \sim \text{Mult} \left( 1; \hat{\pi}_{i,1}, \dots, \hat{\pi}_{i,K_{\max}} \right)$$

during the  $b^{th}$  iteration (draw), and obtain  $\kappa_{(\text{VI})}^{(b,t)}$  as the number of unique cluster labels  $\left\{ z_{i(\text{VI})}^{(b,t)} \right\}_{1 \leq i \leq n}$ .

We then assess the performances of each of these 6 methods (MCMC, RW DP-means, RW SDP-means, RW DP-rich, RW SDP-rich and VI) in each simulation setting using the following comparison criteria:

### 1. Coefficient of variation (CoV) of cluster sizes

For each of the 6 methods, during the  $b^{th}$  posterior draw for  $t^{th}$  simulated training data set, we obtain the cluster labels  $z_{(\cdot)}^{(b,t)}$ , which tells us about the number of clusters  $\kappa_{(\cdot)}^{(b,t)}$  obtained by the method, as well as the cluster sizes  $\left\{ n_{k,(\cdot)}^{(b,t)} \right\}_{1 \leq k \leq \kappa_{(\cdot)}^{(b,t)}}$ . We keep track of the coefficient of variation (CoV) of these cluster sizes

$$\phi_{(\cdot)}^{(b,t)} := \text{CoV of } \left\{ n_{k,(\cdot)}^{(b,t)} \right\}_{1 \leq k \leq \kappa_{(\cdot)}^{(b,t)}}$$

for each of the 6 methods, and then obtain the ecdf of these CoV's

$$\hat{F}_{\phi_{(\cdot)}}^{(t)} = \text{ecdf of } \left\{ \phi_{(\cdot)}^{(b,t)} \right\}_{1 \leq b \leq B}. \quad (4.22)$$

$\hat{F}_{\phi_{(\text{MCMC})}}^{(t)}$  is treated as the ‘‘benchmark curve’’, and is used to compare with  $\hat{F}_{\phi_{(\cdot)}}^{(t)}$  from the other 5 methods, by keeping track of the total variation  $TV_{\phi_{(\cdot)}}^{(t)}$  between  $\hat{F}_{\phi_{(\text{MCMC})}}^{(t)}$  and  $\hat{F}_{\phi_{(\cdot)}}^{(t)}$ .

### 2. Average log posterior predictive density

Denote  $\tilde{y}_i^{(t)}$  as the held-out (test) data for  $\tilde{i} = 1, \dots, m$ , that is generated using the same simulation setting as the  $t^{th}$  set of simulated training data. For MCMC and the 4 random-weighting methods, we compute the average (over the  $t^{th}$  test set) log

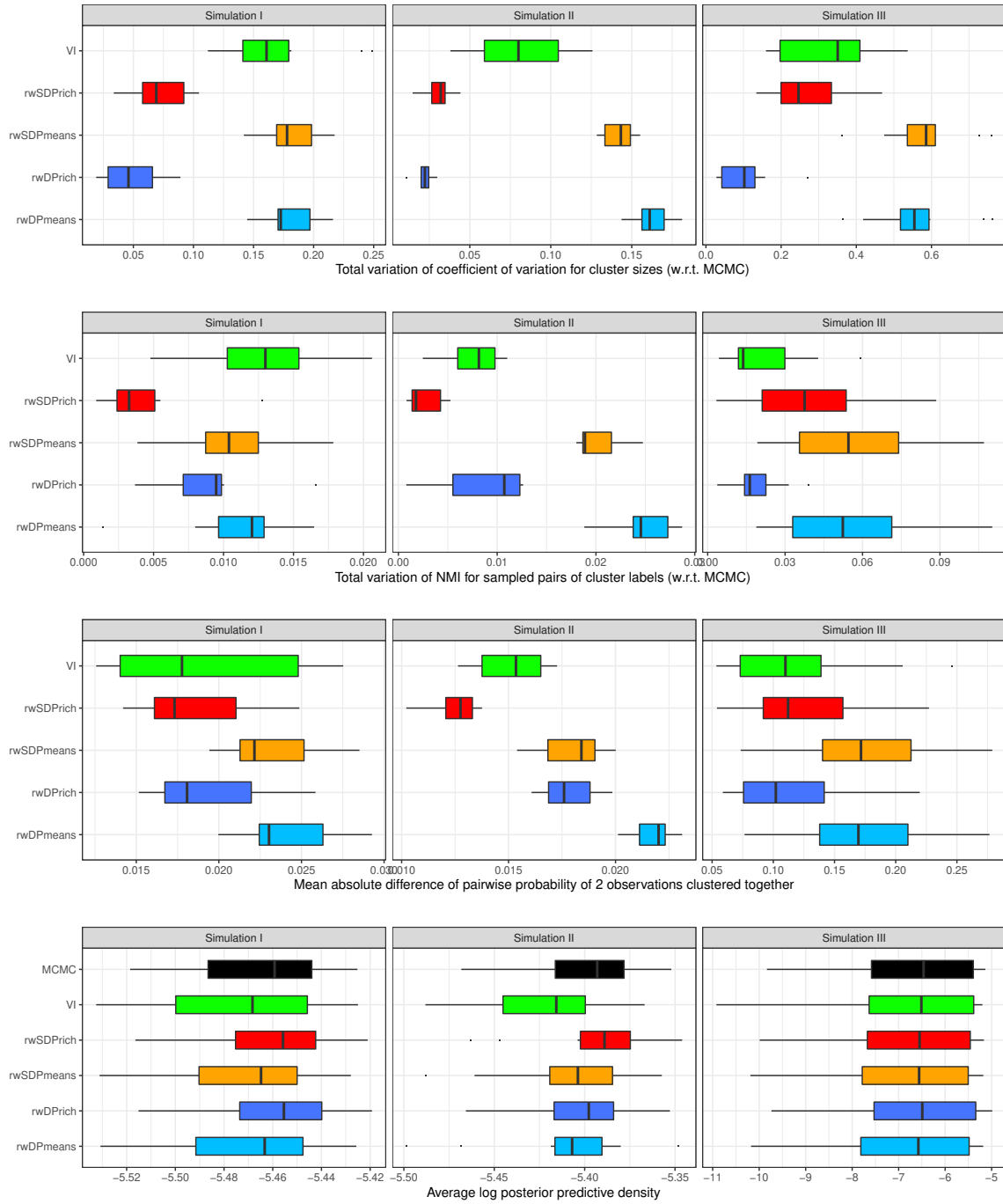


Figure 4.4: Sampling distribution for 4 comparison measurements among  $T = 10$  simulated data sets in 3 simulation settings:  $TV_{\phi(\cdot)}^{(t)}$  (Criterion (1)),  $TV_{\hat{\eta}(\cdot)}^{(t)}$  (Criterion (4)),  $\check{p}_{(\cdot)}^{(t)}$  (Criterion (3)), and  $\tilde{g}_{(\cdot)}^{(t)}$  (Criterion (2)).

posterior predictive density evaluated at these held-out data points as follows:

$$\begin{aligned} \tilde{g}_{(\cdot)}^{(t)} := & \frac{1}{m} \sum_{i=1}^m \log \left\{ \frac{1}{B} \sum_{b=1}^B \left[ \sum_{k=1}^{\kappa_{(\cdot)}^{(b,t)}} \frac{n_{k(\cdot)}^{(b,t)}}{n + \alpha_0} f_{T_d} \left( \tilde{y}_i^{(t)} \mid \tilde{\nu}_{k(\cdot)}^{(b,t)}, \tilde{\mu}_{k(\cdot)}^{(b,t)}, \tilde{\Sigma}_{k(\cdot)}^{(b,t)} \right) \right] \right. \\ & \left. + \frac{\alpha_0}{n + \alpha_0} f_{T_d} \left( \tilde{y}_i^{(t)} \mid \tilde{\nu}_0, \tilde{\mu}_0, \tilde{\Sigma}_0 \right) \right\}, \end{aligned} \quad (4.23)$$

where  $f_{T_d}(y|\nu, \mu, \Sigma)$  denotes the  $d$ -dimensional multivariate  $T$  density (with  $\nu$  degrees of freedom as well as location and scale parameters  $\mu$  and  $\Sigma$ ) evaluated at  $y$ . The subscript  $k(\cdot)$  and superscript  $(b, t)$  for the  $T$  density parameters represent their specific values computed based on the  $b^{\text{th}}$  posterior samples obtained by one of the 5 methods (MCMC or random weighting) for the  $k^{\text{th}}$  cluster in the  $t^{\text{th}}$  simulated data set. The formula for these multivariate  $T$  densities in (4.23) follows that of the posterior predictive density corresponding to a conjugate normal-inverse-Wishart prior. In fact, (4.23) computes the average log posterior predictive density under the full-covariance structure; the formulae for its counterpart under the diagonal-covariance structure are given in supplementary material Chapter 5.5. Meanwhile, the average log posterior predictive density for VI is computed based on its variational densities and its corresponding variational parameters. Detailed formulae are provided in the supplementary material Chapter 5.6.

### 3. Pairwise probability of any two observations clustered together

We keep track of the probability of clustering the  $i^{\text{th}}$  and  $j^{\text{th}}$  observations together by MCMC and the 4 random-weighting schemes in the  $t^{\text{th}}$  simulated training dataset

$$\check{p}_{ij(\cdot)}^{(t)} := \frac{1}{B} \sum_{b=1}^B \mathbb{1}_{\{z_{i(\cdot)}^{(b,t)} = z_{j(\cdot)}^{(b,t)}\}}$$

for all  $i, j \in \{1, \dots, n\}$  and  $i \neq j$ . Meanwhile,  $\check{p}_{ij(\text{VI})}^{(t)}$  is calculated based on the variational multinomial probabilities; see Chapter 5.6 for more detail. Then, we compare the other 5 methods against MCMC by computing the average absolute difference of

these pairwise probabilities for the  $t^{\text{th}}$  dataset

$$\hat{p}_{(\cdot)}^{(t)} := \frac{2}{n(n-1)} \sum_{i < j} \left| \hat{p}_{ij(\cdot)}^{(t)} - \hat{p}_{ij(\text{MCMC})}^{(t)} \right|. \quad (4.24)$$

#### 4. Normalized Mutual Information (NMI) based on randomly-sampled pairs of posterior cluster assignments

We could also compare “similarities” of cluster assignments between MCMC and the other 5 methods (VI and random-weighting) in terms of Normalized Mutual Information (e.g. Vinh et al., 2010) that ranges between 0 and 1, with 1 indicating perfect agreement between the two sets of cluster assignments and 0 otherwise. However, this would involve  $B^2$  NMI computations for each of the 5 methods when we compare them against MCMC, which is very computationally intensive. Hence, we would instead randomly sample (with replacement), say,  $\hat{B}$  pairs of cluster assignments and compute their NMI’s. Specifically, let  $\left\{ \hat{z}_{(\cdot)}^{(b,t)} \right\}_{1 \leq b \leq \hat{B}}$  be the random samples from the (standard or approximate) posterior cluster assignments  $\left\{ z_{(\cdot)}^{(b,t)} \right\}_{1 \leq b \leq B}$  obtained by one of the 6 aforementioned methods for the  $t^{\text{th}}$  simulated training dataset. Next, we compute NMI for the  $\hat{b}^{\text{th}}$  randomly-sampled pair of cluster assignments (where one of them is from MCMC) with

$$\hat{\eta}_{(\cdot)}^{(b,t)} := \text{NMI} \left( \hat{z}_{(\cdot)}^{(b,t)}, \hat{z}_{(\text{MCMC})}^{(b,t)} \right),$$

and then obtain the empirical distribution function (ecdf) of these NMI values

$$\hat{F}_{\hat{\eta}_{(\cdot)}}^{(t)} = \text{ecdf of } \left\{ \hat{\eta}_{(\cdot)}^{(b,t)} \right\}_{1 \leq b \leq \hat{B}}. \quad (4.25)$$

In particular,  $\hat{F}_{\hat{\eta}_{(\text{MCMC})}}^{(t)}$  is treated as the “benchmark curve”, and is used to compare with  $\hat{F}_{\hat{\eta}_{(\cdot)}}^{(t)}$  from the other 5 methods, by keeping track of the total variation  $TV_{\hat{\eta}_{(\cdot)}}^{(t)}$  between  $\hat{F}_{\hat{\eta}_{(\text{MCMC})}}^{(t)}$  and  $\hat{F}_{\hat{\eta}_{(\cdot)}}^{(t)}$ . Note that  $\hat{\eta}_{(\text{MCMC})}^{(b,t)}$  is computed as  $\text{NMI} \left( \hat{z}_{(\cdot)}^{(b,t)}, \hat{z}_{(\text{MCMC})}^{(b,t)} \right)$ ,

where  $\check{z}_{(\cdot)}^{(b,t)}$  is another independent random sample of MCMC posterior cluster assignments.

*Comments on comparison criteria.* First, note that all four comparison criteria here circumvent the label-switching problems that complicate many mixture-modeling calculations (Stephens, 2000). Comparison criterion (1) illustrates the variability in posterior cluster assignments obtained by the 6 methods. Ideally, the other 5 approximate methods should mimic the variability displayed by MCMC samples under criterion (1), so total variation distance (in comparison with MCMC) should ideally be small. Criterion (2) is popular in existing mixture-modeling and clustering literature, and higher average log posterior predictive density indicates “better prediction for the test data”. Meanwhile, we also consider criteria (3) and (4) in order to compare the “similarities” between MCMC posterior cluster assignments and those obtained by the other 5 methods. Higher degree of agreement in cluster assignments between MCMC and the other 5 methods should lead to lower  $\check{p}_{(\cdot)}^{(t)}$  and  $TV_{\check{\eta}(\cdot)}^{(t)}$ .

The simulation results are presented in Figure 4.4. Overall, RW SDP-rich obtains the best approximation to MCMC clustering results as compared to VI and the other 3 random-weighting setups, as it has the smallest total variation distance  $\left\{TV_{\check{\eta}(\cdot)}^{(t)}\right\}_{1 \leq t \leq 10}$  as well as the smallest mean absolute difference in pairwise probabilities of clustering any two observations  $\left\{\check{p}_{(\cdot)}^{(t)}\right\}_{1 \leq t \leq 10}$  across the 3 simulation settings. The presence of the cluster covariance term  $\Sigma$  in RW SDP-means and RW SDP-rich allows them to perform better than their respective counterparts without feature-scaling (i.e., RW DP-means and RW DP-rich, respectively) in Simulation Setting II where data features are more correlated. The boxplots for total variation distance of CoV of cluster sizes illustrate that the presence of *rgr* regularization in RW DP-rich and RW SDP-rich allows them to better mimic MCMC posterior variation in cluster samples than VI and their respective counterparts without *rgr* penalty – RW DP-means and RW SDP-means, in all 3 simulation settings. All 6 methods (MCMC, VI and the 4 random-weighting setups) have very similar average log posterior predictive



densities in all simulation settings (with VI and RW DP-rich register slightly lower values in Simulation Setting II).

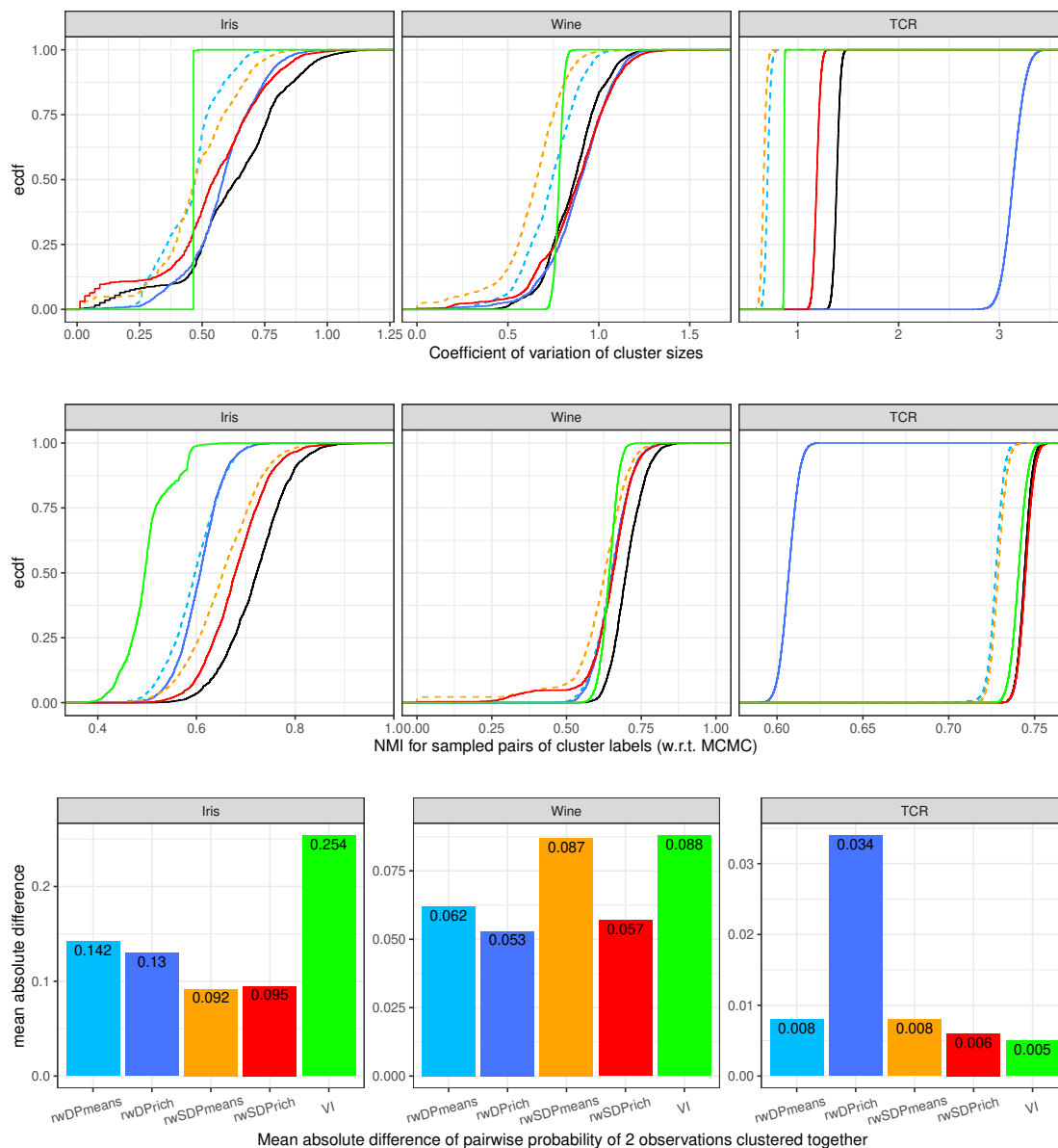


Figure 4.5: The ecdf curves of CoV of cluster sizes (see, Equation (4.22)) and the ecdf curves of NMI (see, Equation (4.25)) comparing randomly-sampled pairs of cluster assignments for all 6 methods – MCMC (solid black), VI (solid green), RW DP-means (dashed light-blue), RW DP-rich (solid dark-blue), RW SDP-means (dashed orange) and RW SDP-rich (solid red), as well as the barplots depicting mean absolute differences (in comparison with MCMC) of pairwise probabilities of clustering any two observations together (see, Equation (4.24)) for the other 5 methods, among the 3 benchmark and motivating data examples.

### 4.3.2. Benchmark Data Examples

Next, we deploy all the 6 aforementioned methods on two benchmark data sets – *iris* and *wine*, which are commonly found in many clustering and classification literature. Briefly, the *iris* data set (Anderson, 1935) gives the measurements of sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris (i.e.,  $n = 150$ ,  $d = 4$  and  $K_{true} = 3$ ). Meanwhile, the *wine* data set, which is available in the R package `rattle.data` (Williams, 2011), contains the results of 13 chemical analyses for 178 samples (that belong to either one of the three classes) of wine grown in a specific area of Italy (i.e.,  $n = 178$ ,  $d = 13$  and  $K_{true} = 3$ ).

For each benchmark data set, we draw  $B = 2000$  posterior (or approximate posterior) samples for each method. We refer readers to the supplementary material Chapter 5.5 for details about specifying the priors for MCMC, VI, RW SDP-rich and RW SDP-means.  $\lambda_2$  for RW DP-rich is then specified using the (average, across all features, of) prior mean of mixture-component variance. Since  $K_{true}$  is small for both benchmark data sets, the stick-breaking threshold for VI is fixed at  $K_{max} = 10$ . For *iris* data set, the full-covariance structure is adopted for MCMC, VI, RW SDP-rich and RW SDP-means. Meanwhile, we point out that MCMC has poor mixing (as indicated by the MCMC trace plot in the supplementary material Chapter 5.5) when we adopt the full covariance structure for the original *wine* data set. Consequently, we perform a PCA on the data set, and use the first 5 principal components (which explains more than 80% of variation in the data) as our transformed data set. The diagonal-covariance structure is thus adopted for MCMC, VI, RW SDP-rich and RW SDP-means, since principal components are uncorrelated by construction.

Based on the clustering results obtained by the 6 methods, we obtain their respective ecdf curves for coefficient of variation of cluster sizes  $\hat{F}_{\phi(\cdot)}$  (see, Equation (4.22)) and ecdf curves for NMI  $\hat{F}_{\eta(\cdot)}$  computed based on randomly-sampled pairs of posterior cluster assignments (see, Equation (4.25)). We also keep track of the mean absolute difference of pairwise probabilities  $\check{p}_{(\cdot)}$  (for any two observations to be clustered together) computed by the other 5 methods in comparison with MCMC (see, Equation (4.24)).

The results are presented in Figure 4.5. Overall, RW SDP-rich provides the best approximation to MCMC posterior cluster assignments among all other methods, since its (solid red) ecdf curve hugs the MCMC (solid black) ecdf curve the tightest. Furthermore, in both benchmark data sets, RW SDP-rich also has (nearly) the smallest mean absolute difference of pairwise probabilities of clustering any two observations together, whereas VI reports the highest value in this criterion. Notice that for `iris` data set, the ecdf curves of NMI for sampled pairs of cluster assignments for RW DP-means and RW DP-rich (dashed light-blue and solid dark-blue curves, respectively) are further away from the MCMC (solid black) ecdf curve than the ecdf curves for RW SDP-means and RW SDP-rich, due to the former’s lack of feature-scaling limitation in capturing the feature correlation in the `iris` data set. This pattern is not observed in the `wine` data set because we are working on the transformed data set via PCA and principal components are uncorrelated by construction. From the ecdf of CoV of cluster sizes, it is evident that VI severely underestimates posterior variation in cluster assignments in both benchmark data sets. In fact, most of the VI samples  $\{z_{(VI)}^{(b)}\}_{1 \leq b \leq B}$  show (almost) the same partition. This finding is also consistent with VI’s poor performance (in terms of approximating posterior variation) in the simulations; see Figure 4.4. Similar limitation has also been reported in Fong et al. (2019). Again, the lack of *rgr* regularization in RW DP-means and RW SDP-means causes their ecdf curves (dashed light-blue and dashed orange curves, respectively) for CoV of cluster sizes to be further away from the MCMC ecdf curve than RW DP-rich and RW SDP-rich.

### 4.3.3. Motivating Example: T-cell Receptor Data

Now we consider our motivating T-cell Receptor (TCR) data example. Specifically, Zahm, Ng, Newton, and McNeel (2022) sequenced 13387 TCR sequences from 70 mice, which were administered with different experimental antigens in order to study antigen specificity of TCR sequences in mice. Clustering of TCR based on sequence “similarities” to reflect antigen specificity has gained traction in literature recently (e.g. Vujovic et al., 2020), which has been aided by availability of software packages such as `tcrdist3` that computes the

pairwise distances of TCR sequences based on their sequencing reads (Dash et al., 2017). We are interested in the uncertainty quantification of clustering these TCR sequences, using the methods that are developed in this paper.

We note that pairwise distances of data points could be utilized by certain clustering methods, such as hierarchical clustering or K-medoids. However, all methods that are mentioned or developed in this paper (MCMC, VI and random-weighting) work on Euclidean data points. Consequently, using the classical multidimensional-scaling (MDS) approach (Hastie et al., 2009), we map these  $(13387 \times 13386)/2$  pairwise distances into a 3-dimensional Euclidean space, which leaves us with a data set where  $n = 13387$  and  $d = 3$ . We then draw  $B = 20,000$  posterior (or approximate posterior) samples for each method. Again, since these 3-dimensional eigenvectors are uncorrelated by construction, we adopt the diagonal-covariance structure for MCMC, VI, RW SDP-means and RW SDP-rich. The priors for these methods are estimated using a hierarchical-clustering procedure that is implemented based on the pairwise distances.  $\lambda_2$  for RW DP-rich is then specified using the (average, across all features, of) prior mean of mixture-component variance. We refer readers to the supplementary material Chapter 5.5 for details about specifying the priors and the stick-breaking threshold for VI.

Again, we compare the posterior cluster assignments from all the 6 methods with the same criteria that we used for our benchmark data examples. From Figure 4.5, the RW SDP-rich (solid red) ecdf curves are the closest to the MCMC (solid black) ecdf curves, thus suggesting better approximation to posterior cluster assignments than VI and the other random-weighting schemes. We also note that  $\check{p}_{(VI)}$  is the smallest, which is closely followed by  $\check{p}_{(rwSDP-rich)}$  (see, Criterion (3)). It is worth noting that RW DP-rich has the worst performance in this case; in fact, it is way off from all the other methods. This data analysis example illustrates the tricky nature of calibrating  $\lambda_2^{rwDP-rich}$ : if  $\lambda_2^{rwDP-rich}$  is too small, then its performance is no different from a RW DP-means implementation; on the other hand, if  $\lambda_2^{rwDP-rich}$  is too big, then the performance of RW DP-rich is adversely affected. Hence, in practice, RW SDP-rich is preferred over RW DP-rich since the variance of the

mixture components is part of the model variables to be optimized under the RW SDP-rich approach, instead of being a tuning parameter that needs to be carefully calibrated by the analyst under the RW DP-rich setup.

#### 4.4. Theoretical Properties

We first furnish additional definitions relating the Bayesian NPL framework (Lyddon et al., 2018) from Section 4.1 to the clustering parameters in our random-weighting approach for mixture models from Section 4.2. Then, we present our asymptotic results under this framework. The proofs for all the theorems in this section are collected in the supplementary material Chapter 5.4.

Specifically, from Section 4.1, we are interested in expected loss  $\mathcal{L}(t, F)$  where posterior sampling of  $F$  is approximated with Bayesian bootstrap  $F_w$ , which leads to

$$\mathcal{L}(t, F_w) = \int_{\Omega} \tilde{l}(t, y) dF_w(y) = \int_{\Omega} l(t, y) dF_w(y) + \lambda_0 l_0(t) = \sum_{i=1}^n w_i l(t, y_i) + \lambda_0 l_0(t), \quad (4.26)$$

where  $(w_1, \dots, w_n) \sim \text{Dir}(1, \dots, 1)$  and  $\lambda_0 > 0$  is supplied by the analyst. Then, we arrive at (4.2) by normalizing the standard dirichlet random weights into i.i.d. standard Exponential random weights, as well as replacing  $\lambda_0 \sum_{i=1}^n W_i$  with  $\lambda$  in (4.2). We refer interested readers to the supplementary material Chapter 5.4 for detailed derivation of this Bayesian NPL approach. The following subsections will instead focus on the setup in (4.26).

#### 4.4.1. Definitions under Bayesian NPL Framework

##### RW K-means

First, let  $A_K = \{a_1, \dots, a_K\}$  be a set of  $K$  points on  $\mathbb{R}^d$ , and we want to find  $A_K$  that minimizes

$$\inf_{A_K} \mathcal{L}(A_K, F_w) = \inf_{A_K} \left\{ \int_{\Omega} \min_{a \in A_K} \|y - a\|_2^2 dF_w(y) \right\} = \min_{(\mu, \mathbf{z})} \left\{ \sum_{k=1}^K \sum_{i: z_i=k} w_i \|y_i - \mu_k\|_2^2 \right\}, \quad (4.27)$$

where  $(w_1, \dots, w_n) \sim \text{Dir}(1, \dots, 1)$ , and  $F_w$  is the Bayesian bootstrap defined in (4.26). From the discussion in Section 4.1, it is evident that the RHS of (4.27) is related to (4.19). The subscript  $K$  in  $A_K$  highlights the fact that the variable depends on the choice of  $K$  specified by the analyst under the RW K-means approach.

Furthermore, denote  $V_k$  as the Voronoi region generated by  $a_k$

$$V_k := \left\{ y_i \in \Omega : \|y_i - a_k\|_2^2 < \|y_i - a_{k'}\|_2^2 \text{ for all } k' \neq k \right\}. \quad (4.28)$$

Then,  $\bigcup_k V_k$  is the Voronoi tessellation (e.g., Urschel, 2017) of  $\Omega$ , and the set  $\Omega \setminus (\bigcup_k V_k)$  consists of data points that are equidistant from more than one centroid. In this subsection, we shall refer to the collection of Voronoi regions and  $\Omega \setminus (\bigcup_k V_k)$  as the *Voronoi partition* (denoted as  $\mathcal{P}$ ).

Let  $A_{n,K}^w := \arg \min_{A_K} \mathcal{L}(A_K, F_w)$  be the minimizer of (4.27), where the subscript  $n$  indicates that the set of centroids changes with dataset. Then, the NPL posterior distribution  $\Pi_n(A_K|y)$  has a corresponding (approximate) posterior density

$$\pi(A_K|y) = \int \pi(A_K|F_w) d\pi(F_w), \quad (4.29)$$

where the approximation comes from the fact that the integral in (4.29) is performed w.r.t.

the Bayesian bootstrap approximation  $F_w$ , and

$$\pi(A_K | F_w) = \delta_{A_{n,K}^w(F_w)}(A_K). \quad (4.30)$$

The delta arises because  $A_K$  is a deterministic functional of  $F_w$  from (4.27). Notice that  $A_{n,K}^w$  depends on  $F_w$ , i.e.  $A_{n,K}^w$  depends on the independent dirichlet weights  $(w_1, \dots, w_n)$ . See also, Section 2.3 of Fong et al. (2019) for a discussion of Bayesian NPL posterior. In addition, for the delta in (4.30) to be well-defined, we implicitly assume that  $A_{n,K}^w$  is unique a.s.  $P_{F_w}$ , i.e. the set of centroids that minimizes (4.27) is unique for almost every set of dirichlet weights.

Similarly, let  $\mathcal{P}_{n,K}^w$  be the *Voronoi partition* associated with  $A_{n,K}^w$ . Then, the NPL posterior distribution  $\Pi_n(\mathcal{P}_K | y)$  has a corresponding (approximate) posterior density

$$\pi(\mathcal{P}_K | y) = \int \pi(\mathcal{P}_K | F_w) d\pi(F_w), \quad (4.31)$$

where  $\pi(\mathcal{P}_K | F_w) = \delta_{\mathcal{P}_{n,K}^w(F_w)}(\mathcal{P}_K)$ .

### RW DP-means

To derive the RW DP-means objective function from the Bayesian NPL perspective, let  $A_{\lambda_0} = \{a_1, \dots, a_\kappa\}$  be a set of  $\kappa$  points on  $\mathbb{R}^d$  for  $\kappa = |A_{\lambda_0}| \in \mathbb{N}$ , and we want to find  $A_{\lambda_0}$  that minimizes

$$\begin{aligned} \inf_{A_{\lambda_0}} \mathcal{L}(A_{\lambda_0}, F_w) &= \inf_{A_{\lambda_0}} \left\{ \int_{\Omega} \min_{a \in A_{\lambda_0}} \|y - a\|_2^2 dF_w(y) + \lambda_0 \cdot |A_{\lambda_0}| \right\} \\ &= \min_{(\mu, \mathbf{z}, \kappa)} \left\{ \sum_{k=1}^{\kappa} \sum_{i: z_i=k} w_i \|y_i - \mu_k\|_2^2 + \lambda_0 \kappa \right\}, \end{aligned} \quad (4.32)$$

where  $\lambda_0 > 0$  is a tuning parameter,  $F_w$  is the Bayesian bootstrap defined in (4.26), and  $(w_1, \dots, w_n) \sim \text{Dir}(1, \dots, 1)$ . In addition, denote  $\mathcal{P}_{\lambda_0}$  as the *Voronoi partition* associated with  $A_{\lambda_0}$ . The subscript  $\lambda_0$  in  $A_{\lambda_0}$  and  $\mathcal{P}_{\lambda_0}$  highlights the fact that the variables depend on

the tuning parameter  $\lambda_0$ . We note that (4.32) is also in line with the concept of Loss NPL introduced in Section 2.6 of Fong et al. (2019). Again, from Section 4.1, it is evident that (4.32) is related to (4.18).

Let  $A_{n,\lambda_0}^w := \arg \min_{A_{\lambda_0}} \mathcal{L}(A_{\lambda_0}, F_w)$  be the minimizer of (4.32), and let  $\mathcal{P}_{n,\lambda_0}^w$  be the Voronoi partition associated with  $A_{n,\lambda_0}^w$ . Again, the subscript  $n$  indicates that the solutions change with dataset. Then, the NPL posterior distribution  $\Pi_n(A_{\lambda_0}|y)$  has a corresponding (approximate) posterior density

$$\pi(A_{\lambda_0}|y) = \int \pi(A_{\lambda_0}|F_w) d\pi(F_w), \quad (4.33)$$

where  $\pi(A_{\lambda_0}|F_w) = \delta_{A_{n,\lambda_0}^w(F_w)}(A_{\lambda_0})$ , whereas the NPL posterior distribution  $\Pi_n(\mathcal{P}_{\lambda_0}|y)$  has a corresponding (approximate) posterior density

$$\pi(\mathcal{P}_{\lambda_0}|y) = \int \pi(\mathcal{P}_{\lambda_0}|F_w) d\pi(F_w), \quad (4.34)$$

where  $\pi(\mathcal{P}_{\lambda_0}|F_w) = \delta_{\mathcal{P}_{n,\lambda_0}^w(F_w)}(\mathcal{P}_{\lambda_0})$ .

### RW SDP-means

We also analyze the random-weighting scaled DP-means setup for the case where  $\xi_0 = 0$  and the common covariance term  $\Sigma$  is pre-specified with a symmetric positive-definite matrix  $\Sigma_0$  (For the case of  $\xi_0 > 0$ , the notation is only slightly more cumbersome but uninteresting, because its associated "prior" term will be overwhelmed by data information as sample size  $n$  increases). From Section 4.1, by recognizing  $(2\lambda_1)$  to be  $(\lambda_0 \sum_{i=1}^n W_i)$  for  $W_i \stackrel{iid}{\sim} Exp(1)$ , we can re-specify the RW SDP-means setup into the form of

$$\begin{aligned} & \inf_{A_{(\lambda_0, \Sigma_0)}} \mathcal{L}(A_{(\lambda_0, \Sigma_0)}, F_w) \\ &= \inf_{A_{(\lambda_0, \Sigma_0)}} \left\{ \int_{\Omega} \min_{a \in A_{(\lambda_0, \Sigma_0)}} (y - a)' \Sigma_0^{-1} (y - a) dF_w(y) + \lambda_0 \cdot |A_{(\lambda_0, \Sigma_0)}| \right\} \end{aligned} \quad (4.35)$$



$$= \min_{(\boldsymbol{\mu}, \boldsymbol{z}, \kappa)} \left\{ \sum_{k=1}^{\kappa} \sum_{i: z_i=k} w_i (y_i - \mu_k)' \Sigma_0^{-1} (y_i - \mu_k) + \lambda_0 \kappa \right\},$$

where  $\kappa = |A_{(\lambda_0, \Sigma_0)}|$ , and  $(w_1, \dots, w_n) \sim \text{Dir}(1, \dots, 1)$ . We also denote  $\mathcal{P}_{(\lambda_0, \Sigma_0)}$  as the *Voronoi partition* associated with  $A_{(\lambda_0, \Sigma_0)}$ . The subscript  $(\lambda_0, \Sigma_0)$  highlights the fact that the variables depend on the choices of  $\lambda_0$  and  $\Sigma_0$ . Basically, the setup in (4.32) is a special case of (4.35) with  $\Sigma_0 = I_d$ .

Again, let  $A_{n, (\lambda_0, \Sigma_0)}^w := \arg \min_{A_{(\lambda_0, \Sigma_0)}} \mathcal{L}(A_{(\lambda_0, \Sigma_0)}, F_w)$  be the minimizer of (4.35), and let  $\mathcal{P}_{n, (\lambda_0, \Sigma_0)}^w$  be the *Voronoi partition* associated with  $A_{n, (\lambda_0, \Sigma_0)}^w$ . Then, the NPL posterior distribution  $\Pi_n(A_{(\lambda_0, \Sigma_0)} | y)$  has a corresponding (approximate) posterior density

$$\pi(A_{(\lambda_0, \Sigma_0)} | y) = \int \pi(A_{(\lambda_0, \Sigma_0)} | F_w) d\pi(F_w), \quad (4.36)$$

where  $\pi(A_{(\lambda_0, \Sigma_0)} | F_w) = \delta_{A_{n, (\lambda_0, \Sigma_0)}^w(F_w)}(A_{(\lambda_0, \Sigma_0)})$ , whereas the NPL posterior distribution  $\Pi_n(\mathcal{P}_{(\lambda_0, \Sigma_0)} | y)$  has a corresponding (approximate) posterior density

$$\pi(\mathcal{P}_{(\lambda_0, \Sigma_0)} | y) = \int \pi(\mathcal{P}_{(\lambda_0, \Sigma_0)} | F_w) d\pi(F_w), \quad (4.37)$$

where  $\pi(\mathcal{P}_{(\lambda_0, \Sigma_0)} | F_w) = \delta_{\mathcal{P}_{n, (\lambda_0, \Sigma_0)}^w(F_w)}(\mathcal{P}_{(\lambda_0, \Sigma_0)})$ .

#### 4.4.2. Asymptotic Results

Lyddon et al. (2018) and Fong et al. (2019) mentioned about the Bayesian NPL strong consistency property of the solutions or samples  $\theta_w := \arg \min_{t \in \Theta} \mathcal{L}(t, F_w)$  in (4.26), i.e.

$$\theta_w \longrightarrow \theta_* := \arg \min_{t \in \Theta} \mathcal{L}(t, F_*)$$

almost surely  $P_{F_*}^{(\infty)}$ , which relies on the strong consistency property of the Bayesian bootstrap

$$F_w \longrightarrow F_* \quad a.s. \ P_{F_*}^{(\infty)}, \quad (4.38)$$

where the convergence of random measure in (4.38) takes place on a space of probability measures under the weak topology characterized by the Portmanteau Theorem as outlined in Section A.2 of Ghosal and van der Vaart (2017).

In this subsection, we present a rigorous discussion on how the solutions or samples obtained by our random-weighting mixture models satisfy the Bayesian NPL strong consistency property under certain regularity conditions.

First, we consider the metric space  $(\mathcal{A}, \mathcal{D}_{\mathcal{H}})$ , where  $\mathcal{A}$  is the set of all centroid sets (i.e. sets of  $k$  discrete Euclidean points, where  $k = 1, 2, \dots$ ), and  $\mathcal{D}_{\mathcal{H}}$  is the Hausdorff metric. Under this metric space, we establish that, as sample size increases, the posterior distributions of  $A_K$ ,  $A_{\lambda_0}$  and  $A_{(\lambda_0, \Sigma_0)}$  congregate at their respective asymptotic limits  $A_{*,K}$ ,  $A_{*,\lambda_0}$  and  $A_{*,(\lambda_0, \Sigma_0)}$  which are defined below in (4.39), (4.41) and (4.43).

**Theorem 4.4. (Bayesian NPL Strong Consistency for the set of centroids)** *Assume that  $F_*$  has finite second moment. Furthermore,*

- (a) **(RW K-means)** *suppose that under  $F_*$  and the choice of  $K \geq 1$ , there exists a unique set  $A_{*,K}$  of  $K$  points on  $\mathbb{R}^d$  such that*

$$A_{*,K} = \arg \min_{A_K} \mathcal{L}(A_K, F_*) = \arg \min_{A_K} \left\{ \int_{\Omega} \min_{a \in A_K} \|y - a\|_2^2 dF_*(y) \right\}. \quad (4.39)$$

*Then, for every  $\epsilon > 0$ , as  $n \rightarrow \infty$ ,*

$$\Pi_n (A_K : \mathcal{D}_{\mathcal{H}}(A_K, A_{*,K}) > \epsilon | y) \rightarrow 0 \quad a.s. P_{F_*}^{(\infty)}, \quad (4.40)$$

*where the posterior distribution  $\Pi_n$  of  $A_K$  is defined in (4.29).*

- (b) **(RW DP-means)** *suppose that under  $F_*$  and the choice of  $\lambda_0 > 0$ , there exists a unique set  $A_{*,\lambda_0}$  of  $\kappa = |A_{*,\lambda_0}|$  points on  $\mathbb{R}^d$  such that*

$$A_{*,\lambda_0} = \arg \min_{A_{\lambda_0}} \mathcal{L}(A_{\lambda_0}, F_*) = \arg \min_{A_{\lambda_0}} \left\{ \int_{\Omega} \min_{a \in A_{\lambda_0}} \|y - a\|_2^2 dF_*(y) + \lambda_0 \kappa \right\}. \quad (4.41)$$

Then, for every  $\epsilon > 0$ , as  $n \rightarrow \infty$ ,

$$\Pi_n (A_{\lambda_0} : \mathcal{D}_{\mathcal{H}} (A_{\lambda_0}, A_{*,\lambda_0}) > \epsilon | y) \rightarrow 0 \quad a.s. P_{F_*}^{(\infty)}, \quad (4.42)$$

where the posterior distribution  $\Pi_n$  of  $A_{\lambda_0}$  is defined in (4.33).

(c) **(RW SDP-means)** suppose that under  $F_*$  and the choices of  $\lambda_0 > 0$  and symmetric positive-definite  $\Sigma_0$ , there exists a unique set  $A_{*,(\lambda_0,\Sigma_0)}$  of  $\kappa = |A_{*,(\lambda_0,\Sigma_0)}|$  points on  $\mathbb{R}^d$  such that

$$A_{*,(\lambda_0,\Sigma_0)} = \arg \min_{A(\lambda_0,\Sigma_0)} \left\{ \int_{\Omega} \min_{a \in A(\lambda_0,\Sigma_0)} [(y - a)' \Sigma_0^{-1} (y - a)] dF_*(y) + \lambda_0 \kappa \right\}. \quad (4.43)$$

Then, for every  $\epsilon > 0$ , as  $n \rightarrow \infty$ ,

$$\Pi_n (A_{(\lambda_0,\Sigma_0)} : \mathcal{D}_{\mathcal{H}} (A_{(\lambda_0,\Sigma_0)}, A_{*,(\lambda_0,\Sigma_0)}) > \epsilon | y) \rightarrow 0 \quad a.s. P_{F_*}^{(\infty)}, \quad (4.44)$$

where the posterior distribution  $\Pi_n$  of  $A_{(\lambda_0,\Sigma_0)}$  is defined in (4.36).

**Comments on Assumptions of Theorem 4.4.** We point out that Pollard (1981) made the same uniqueness assumption on  $A_{*,K}$ . Here, we extend the uniqueness requirement to  $A_{n,K}^w$  to ensure that the posterior distribution  $\Pi_n$  of  $A_K$  is well-defined. Similar uniqueness assumptions are applicable to the RW DP-means and RW SDP-means setups. The uniqueness condition carries a lot of information – similar discussion could be found in the paragraph after the main theorem of Pollard (1981). Here, we shall illustrate this point with a simple example. Consider the case where  $\chi = [0, 1]$  and  $F_* = U(0, 1)$ . Let  $M_1 = \int_0^1 (y - 0.5)^2 dy$  and let  $M_2 = \int_0^{0.5} (y - 0.25)^2 dy + \int_{0.5}^1 (y - 0.75)^2 dy$ . Under RW DP-means, if  $\lambda_0 > (M_1 - M_2)$ , then  $\kappa_{*,\lambda_0} := |A_{*,\lambda_0}| = 1$ . However, if  $\lambda_0 = (M_1 - M_2)$ , then  $\kappa_{*,\lambda_0}$  could be either 1 or 2. We need additional/external rule(s) to resolve this conundrum. In addition, there are also well-known cases where  $A_{*,K}$  or  $A_{\lambda_0}$  or  $A_{(\lambda_0,\Sigma_0)}$  is not unique. For instance, consider the case where  $\Omega$  is a unit circle centered at the origin and  $F_*$  is a Uniform distribution covering the circle. Under RW K-means with  $K = 2$ , the asymptotic limit has infinitely many  $A_{*,2}$ ;

see, for example, Theorem 4.3 of Urschel (2017). We shall revisit the issue about uniqueness assumption when we comment on Theorem 4.5.

Next, we consider the metric space  $(\mathcal{P}, \mathcal{D}_{\mathcal{L}})$ , where  $\mathcal{P}$  is the set of all Voronoi partitions for  $\Omega \subset \mathbb{R}^d$ , and  $\mathcal{D}_{\mathcal{L}}$  is Leonardi and Tamanini (2002)'s metric. It is interesting to note that Leonardi and Tamanini (2002)'s metric  $\mathcal{D}_{\mathcal{L}}$  is not affected by the label-switching problem (Stephens, 2000), and that it could handle partitions with different number of clusters. Under this metric space, we establish that, as sample size increases, the posterior distributions of  $\mathcal{P}_K$ ,  $\mathcal{P}_{\lambda_0}$  and  $\mathcal{P}_{(\lambda_0, \Sigma_0)}$  congregate at their respective asymptotic limits  $\mathcal{P}_{*,K}$ ,  $\mathcal{P}_{*,\lambda_0}$  and  $\mathcal{P}_{*,(\lambda_0, \Sigma_0)}$ , which are the Voronoi partitions corresponding to  $A_{*,K}$ ,  $A_{*,\lambda_0}$  and  $A_{*,(\lambda_0, \Sigma_0)}$  respectively.

**Theorem 4.5. (Bayesian NPL Strong Consistency for partition)** *Assume that  $F_*$  is absolutely continuous (w.r.t. the Lebesgue measure) and has bounded support, i.e.  $\Omega \subset \mathbb{R}^d$ . Furthermore,*

- (a) *adopt the assumptions in part (a) of Theorem 4.4. Let  $\mathcal{P}_{*,K}$  be the Voronoi partition corresponding to  $A_{*,K}$ . Then, for every  $\epsilon > 0$ , as  $n \rightarrow \infty$ ,*

$$\Pi_n (\mathcal{P}_K : \mathcal{D}_{\mathcal{L}} (\mathcal{P}_K, \mathcal{P}_{*,K}) > \epsilon | y) \rightarrow 0 \quad a.s. P_{F_*}^{(\infty)}, \quad (4.45)$$

*where the posterior distribution  $\Pi_n$  of  $\mathcal{P}_K$  is defined in (4.31).*

- (b) *adopt the assumptions in part (b) of Theorem 4.4. Let  $\mathcal{P}_{*,\lambda_0}$  be the Voronoi partition corresponding to  $A_{*,\lambda_0}$ . Then, for every  $\epsilon > 0$ , as  $n \rightarrow \infty$ ,*

$$\Pi_n (\mathcal{P}_{\lambda_0} : \mathcal{D}_{\mathcal{L}} (\mathcal{P}_{\lambda_0}, \mathcal{P}_{*,\lambda_0}) > \epsilon | y) \rightarrow 0 \quad a.s. P_{F_*}^{(\infty)}, \quad (4.46)$$

*where the posterior distribution  $\Pi_n$  of  $\mathcal{P}_{\lambda_0}$  is defined in (4.34).*

- (c) *adopt the assumptions in part (c) of Theorem 4.4. Let  $\mathcal{P}_{*,(\lambda_0, \Sigma_0)}$  be the Voronoi partition corresponding to  $A_{*,(\lambda_0, \Sigma_0)}$ . Then, for every  $\epsilon > 0$ , as  $n \rightarrow \infty$ ,*

$$\Pi_n (\mathcal{P}_{(\lambda_0, \Sigma_0)} : \mathcal{D}_{\mathcal{L}} (\mathcal{P}_{(\lambda_0, \Sigma_0)}, \mathcal{P}_{*,(\lambda_0, \Sigma_0)}) > \epsilon | y) \rightarrow 0 \quad a.s. P_{F_*}^{(\infty)}, \quad (4.47)$$

where the posterior distribution  $\Pi_n$  of  $\mathcal{P}_{(\lambda_0, \Sigma_0)}$  is defined in (4.37).

The following result is a direct consequence of the assumptions adopted in Theorem 4.5.

**Lemma 4.6 (Zero-measure of decision boundaries).** *Adopt assumptions in Theorem 4.5. Then, the decision-boundary set  $\Omega \setminus \bigcup_k V_k$  (i.e., the set of points which are equidistant from more than one centroid) of a Voronoi partition has measure zero.*

**Comments on Assumptions of Theorem 4.5.** The assumptions about bounded support and absolute continuity of  $F_*$  are required for Leonardi and Tamanini (2002)'s metric  $\mathcal{D}_{\mathcal{L}}$ . Next, note that the bounded support assumption also immediately ensures finite second moment for  $F_*$ , which allows us to continue adopting the same sets of assumptions from Theorem 4.4. Meanwhile, under Leonardi and Tamanini (2002)'s metric  $\mathcal{D}_{\mathcal{L}}$ , uniqueness of the *Voronoi partitions* is defined up to sets of measure zero; the metric does not distinguish different allocation of data points that fall on the decision-boundary set which has measure zero due to Lemma 4.6. For example, consider, again, the case where  $\Omega = [0, 1]$  and  $F_* = U(0, 1)$ . Under RW K-means approach where  $K = 2$ ,  $\mathcal{P}_{*,2}$  could be either  $\{[0, 1/2], (1/2, 1]\}$  or  $\{[0, 1/2), [1/2, 1]\}$ , because  $\mathcal{D}_{\mathcal{L}}(\{[0, 1/2], (1/2, 1]\}, \{[0, 1/2), [1/2, 1]\}) = 0$ .

**Connection to Centroidal Voronoi Tessellation.** We also want to point out that the objective function

$$\min_{a \in A_K} \left\{ \int_{\mathcal{P}_K} g(y - a) dF_*(y) \right\}, \quad (4.48)$$

where  $g(y - a)$  could be either  $\|y - a\|_2^2$  or  $(y - a)' \Sigma_0^{-1} (y - a)$  for a given symmetric positive-definite  $\Sigma_0$ , is related to the topic of Centroidal Voronoi Tessellation; see, for example, Urschel (2017), Richter and Alexa (2015) and references therein. The asymptotic limit for RW K-means in Theorems 4.4 and 4.5 is exactly (4.48) with squared Euclidean distance, whereas for RW DP-means, its asymptotic limit in Theorems 4.4 and 4.5 could be thought of as applying (4.48) with squared Euclidean distance on the grid of positive integers  $\mathbb{N}$  and then picking the solution that corresponds to the smallest objective (that has been penalized with  $\lambda_0 K$  for

$K = 1, 2, \dots$ ). Similar argument is also applicable to the asymptotic limit of RW SDP-means (with a fixed  $\Sigma_0$ ) in Theorems 4.4 and 4.5, but this time with the Mahalanobis distance instead. We acknowledge that the uniqueness assumption on  $(A_{*,K}, \mathcal{P}_{*,K})$ ,  $(A_{*,\lambda_0}, \mathcal{P}_{*,\lambda_0})$  or  $(A_{*,(\lambda_0, \Sigma_0)}, \mathcal{P}_{*,(\lambda_0, \Sigma_0)})$  is rather strict; to the best of our knowledge, there are currently no general theorems that outline the (sufficient and/or necessary) conditions for uniqueness of solution to (4.48) that apply to every possible scenario. We refer interested readers to the aforementioned references on the characterization of (4.48) in certain specific settings, which is beyond the scope of this paper.

**Remark 4.2.** *In this paper, we examine the Bayesian NPL strong consistency properties of RW K-means, RW DP-means and RW SDP-means. In fact, the same asymptotic limits (for the sets of centroids) in Theorem 4.4 are also applicable to RW DP-rich and RW SDP-rich if we ensure that the rgr penalty terms vanish in the limit by shrinking  $\lambda_2 = o\left((n \log n)^{-1}\right)$ , due to the fact that*

$$\sum_{k=1}^{\kappa} \log \Gamma(n_k) \leq \log \Gamma(n) = \mathcal{O}(n \log n)$$

from Sterling's Formula. However, in this case, due to the presence of rgr penalty  $\lambda_2 > 0$  in finite samples, the solutions no longer respect a Voronoi partition.

Finally, we present a simple asymptotic result that is not related to the Bayesian NPL framework. Consider the special case where we already have a fixed partition  $\mathcal{P}_0$  of  $\Omega$  consisting of  $K_0 \geq 1$  disjoint clusters  $\{C_1^0, \dots, C_{K_0}^0\}$ . Conditional on this partition  $\mathcal{P}_0$ , the RW K-means (4.19), RW DP-means(4.18) and RW DP-rich (4.17) setups are reduced to obtaining random-weighting centroids

$$\mu_{n,k}^w = \frac{\sum_{i \in C_k^0} W_i y_i}{\sum_{i \in C_k^0} W_i} \quad (4.49)$$

for  $k = 1, \dots, K_0$  and  $W_i \stackrel{iid}{\sim} Exp(1)$ , since cluster-reassignment steps are no longer performed in this case. Similarly, the RW SDP-means and RW SDP-rich setups are reduced to

obtaining (random-weighting  $\Sigma_w$  and) random-weighting centroids

$$\mu_{n,k}^w = \frac{\sum_{i \in \mathcal{C}_k^0} W_i y_i + \xi_0 \mu_0}{\sum_{i \in \mathcal{C}_k^0} W_i + \xi_0} \quad (4.50)$$

for  $k = 1, \dots, K_0$  and  $W_i \stackrel{iid}{\sim} Exp(1)$ . Conditional on data with a fixed partition  $\mathcal{P}_0$  of  $\Omega$ , we prove that these random-weighting centroids, which are centered on their corresponding sample mean of the cluster

$$\hat{\mu}_{n,k} = \frac{1}{n_k} \sum_{i \in \mathcal{C}_k^0} y_i \quad (4.51)$$

for  $n_k = |\mathcal{C}_k^0|$ , are asymptotically normal as  $n \rightarrow \infty$ . To simplify notation, denote

$$V_{*,k}^{\mathcal{P}_0} := \int_{\mathcal{C}_k^0} yy' dF_*(y) - \left[ \int_{\mathcal{C}_k^0} y dF_*(y) \right] \left[ \int_{\mathcal{C}_k^0} y dF_*(y) \right]'$$

**Theorem 4.7 (Asymptotic Normality).** *Assume that  $F_*$  has finite second moments. Suppose  $\Omega$  has a fixed partition  $\mathcal{P}_0$  with  $K_0 \geq 1$  disjoint clusters. Conditional on  $\mathcal{P}_0$ , consider the sample mean  $\hat{\mu}_{n,k}$  defined in (4.51) and the random-weighting centroid  $\mu_{n,k}^w$  defined in (4.49) or (4.50). Then, for  $k = 1, \dots, K_0$  and for any Borel set  $B \subset \mathbb{R}^d$ , as  $n_k \rightarrow \infty$ ,*

$$P\left(\sqrt{n_k}(\mu_{n,k}^w - \hat{\mu}_{n,k}) \in B \mid y\right) \rightarrow P(Z \in B) \quad a.s. P_{F_*}^{(\infty)},$$

where  $Z \sim N_d(0, V_{*,k}^{\mathcal{P}_0})$ .

## Chapter 5

# Supplementary Material for Chapter 4

### 5.1. Implementation details of DP-rich

#### 5.1.1. Singleton clusters

Suppose we are now in the cluster reassignment step of the DP-rich and we are considering the data point  $y_i$  that is sitting alone in cluster  $\mathcal{C}_{k'}$ , i.e.  $i \in \mathcal{C}_{k'}$  where  $n_{k'} = 1$ . Then, problems will arise in the DP-rich algorithm. First,  $\log(n_{k',-i}) = \log(n_{k'} - 1)$  would be undefined. Furthermore, we should not consider  $d_{i,\kappa+1}$  in this case, since taking  $y_i$  out of  $\mathcal{C}_{k'}$  (i.e.  $\mathcal{C}_{k'}$  is emptied and subsequently dropped) and putting it into a brand new cluster does not increase the total number of clusters  $\kappa$  in the objective function.

Therefore, if  $i \in \mathcal{C}_{k'}$  where  $n_{k'} = 1$ , we should do the following instead: first, update

$$\mu_{k'} = y_i, \tag{5.1}$$

which is in fact the centroid that we would initialize had we created a new cluster. Then, update the cost to join an existing cluster to be

$$d_{ik} = \|y_i - \mu_k\|_2^2 - \lambda_2 \log(n_{k',-i}) - \lambda_1 \tag{5.2}$$



for all  $k \in \{1, \dots, \kappa\} \setminus \{k'\}$ , whereas

$$d_{ik'} = 0, \quad (5.3)$$

and finally, set  $d_{i,\kappa+1} = \infty$  since we do not consider creating another new cluster in this case. If  $\arg \min_{k \in \{1, \dots, \kappa\}} d_{ik} \neq k'$ , then cluster  $C_{k'}$  is dropped and no observation will ever be allocated to this cluster in subsequent steps. Again, all these modified formulae still ensure that the objective function never increase, and the local convergence property of the DP-rich algorithm is still ensured.

### 5.1.2. Initialization of algorithm

Here are some details that we need to consider when we initialize the DP-rich algorithm.

#### Rich-gets-richer penalty

Recall that the cost to join an existing cluster is given by

$$d_{ik} = \|y_i - \mu_k\|_2^2 - \lambda_2 \log(n_{k,-i}).$$

When we first initialize the algorithm where all data points are grouped together, the term  $\lambda_2 \log(n_{k,-i})$  may be too overwhelming, and the algorithm may fail to break up this single initial cluster.

To overcome this problem, we suggest to set  $\lambda_2 = 0$  in the first (few) epoch(s) of the DP-rich algorithm. One epoch here is defined as one iteration of cluster reassignment through all data points.

However, care has to be taken here, because too many epochs with  $\lambda_2 = 0$  before allowing  $\lambda_2 > 0$  may lead to the DP-rich algorithm to saddle at a local solution that is too similar to the DP-means algorithm. From our numerical experiments, we find that one epoch with  $\lambda_2 = 0$  (before allowing  $\lambda_2 > 0$ ) leads to reasonable performance by the algorithm. We report that

Raykov et al. (2016) faced similar problem with their own algorithm and suggested similar workarounds.

### **Initial cluster labels**

We note that the DP-rich algorithm could also be initialized with more than one cluster. However, we caution the readers against attempts to game or “improve” initialization by using other methods such as the standard K-means, as this might lead to the DP-rich algorithm saddling at suboptimal local solution, i.e. the DP-rich algorithm might produce a solution with a smaller objective had we initialized the algorithm by randomly assigning the observations. Therefore, in this paper, we follow the convention of Kulis and Jordan (2012) as well as Paul and Das (2020), where all observations are grouped together when we initialize the algorithm.

## **5.2. Additional details of RW SDP-rich**

### **5.2.1. Modifying DPM’s negative log-posterior**

Here, we provide further details about the modification of the negative log-posterior of the DPM of Normals to arrive at the objective function of RW SDP-rich. Specifically, we begin with  $h(\Sigma) = \Sigma/\xi_0$  and  $p(\Sigma)$  is inverse-Wishart with  $\nu_0$  degrees of freedom and a symmetric positive-definite scale matrix  $\psi_0$ . Following the Bayesian NPL framework where we assign i.i.d. standard Exponential random weights  $(W_1, \dots, W_n)$  on the likelihood component of

the DPM, we have:

$$\begin{aligned}
& p_w(\mathbf{Y}, \mathbf{z}, \kappa, \{\mu_k\}_{k=1}^\kappa, \Sigma) \\
& := p_w(\mathbf{Y} \mid \mathbf{z}, \kappa, \{\mu_k\}_{k=1}^\kappa, \Sigma) \times p(\{\mu_k\}_{k=1}^\kappa \mid \Sigma, \mathbf{z}, \kappa) \times p(\Sigma) \times p(\mathbf{z}, \kappa) \\
& \propto (2\pi)^{-\frac{d}{2} \sum_{i=1}^n W_i} |\Sigma|^{-\frac{1}{2} \sum_{i=1}^n W_i} \exp \left\{ -\frac{1}{2} \sum_{k=1}^\kappa \sum_{i:z_i=k} W_i (y_i - \mu_k)' \Sigma^{-1} (y_i - \mu_k) \right\} \\
& \times (2\pi)^{-\frac{d\kappa}{2}} \xi_0^{\frac{d\kappa}{2}} |\Sigma|^{-\frac{\kappa}{2}} \exp \left\{ -\frac{\xi_0}{2} \sum_{k=1}^\kappa (\mu_k - \mu_0)' \Sigma^{-1} (\mu_k - \mu_0) \right\} \\
& \times |\Sigma|^{-(\nu_0+d+1)/2} \exp \left\{ -\frac{1}{2} \text{Tr}(\psi_0 \Sigma^{-1}) \right\} \times \alpha_0^{\kappa-1} \frac{\Gamma(\alpha_0 + 1)}{\Gamma(\alpha_0 + n)} \prod_{k=1}^\kappa \Gamma(n_k).
\end{aligned}$$

Then, taking negative log,

$$\begin{aligned}
& -\log p_w(\mathbf{Y}, \mathbf{z}, \kappa, \boldsymbol{\mu}, \Sigma) \\
& = \frac{1}{2} \left[ \sum_{k=1}^\kappa \sum_{i:z_i=k} W_i (y_i - \mu_k)' \Sigma^{-1} (y_i - \mu_k) + \xi_0 \sum_{k=1}^\kappa (\mu_k - \mu_0)' \Sigma^{-1} (\mu_k - \mu_0) + \text{Tr}(\psi_0 \Sigma^{-1}) \right] \\
& + \left( \sum_{i=1}^n W_i + \nu_0 + d + 1 + \kappa \right) \log |\Sigma|^{1/2} \tag{5.4} \\
& + \kappa \log \left( \left( \frac{2\pi}{\xi_0} \right)^{d/2} \cdot \frac{1}{\alpha_0} \right) - \sum_{k=1}^\kappa \log [\Gamma(n_k)] + \text{other terms.}
\end{aligned}$$

Borrowing the idea of the DP-rich algorithm, we replace the coefficient of  $\kappa$  in (5.4) with a tuning parameter  $\lambda_1 > 0$ , and introduce another tuning parameter  $\lambda_2 > 0$  for the *rgr* term in (5.4). Recall that  $\lambda_1$  allows direct calibration by the analyst to tune the number of clusters obtained by the algorithm, whereas  $\lambda_2$  controls the magnitude of the algorithm's *rgr* effect.

Then, we are left with

$$\begin{aligned} & \frac{1}{2} \left[ \sum_{k=1}^{\kappa} \sum_{i:z_i=k} W_i (y_i - \mu_k)' \Sigma^{-1} (y_i - \mu_k) + \xi_0 \sum_{k=1}^{\kappa} (\mu_k - \mu_0)' \Sigma^{-1} (\mu_k - \mu_0) + \text{Tr} (\psi_0 \Sigma^{-1}) \right] \\ & + \left( \sum_{i=1}^n W_i + \nu_0 + d + 1 + \kappa \right) \log |\Sigma^{1/2}| + \lambda_1 \kappa - \lambda_2 \sum_{k=1}^{\kappa} \log [\Gamma(n_k)]. \end{aligned} \quad (5.5)$$

Notice how (5.5) looks very similar to our RW SDP-rich objective function, except for the coefficient of  $\log |\Sigma^{1/2}|$ . Now, if we had adopted (5.5) as our objective function, then solving for  $\Sigma$  (while holding  $\kappa$ ,  $\mathbf{z}$  and  $\{\mu_k\}_{1 \leq k \leq \kappa}$  constant) would have yielded

$$\frac{\sum_{k=1}^{\kappa} \sum_{i:z_i=k} W_i (y_i - \mu_k)(y_i - \mu_k)' + \xi_0 \sum_{k=1}^{\kappa} (\mu_k - \mu_0)(\mu_k - \mu_0)' + \psi_0}{\sum_{i=1}^n W_i + \nu_0 + d + 1 + \kappa}. \quad (5.6)$$

The presence of the term  $\kappa$  in the denominator of (5.6) is problematic. First, recall from the construction of DP-rich,  $\xi_0$  would be small if there is prior belief for larger number of clusters. Thus, the term  $\sum_{k=1}^{\kappa} (\mu_k - \mu_0)(\mu_k - \mu_0)'$  is moderated by  $\xi_0$ , and as sample size increases, the term  $\sum_{k=1}^{\kappa} \sum_{i:z_i=k} W_i (y_i - \mu_k)(y_i - \mu_k)'$  would dominate the other two terms in the numerator of (5.6). However, in the denominator of (5.6), if  $\kappa$  also increases with  $\sum_{i=1}^n W_i$  as sample size increases,  $|\Sigma|^{1/2}$  becomes smaller. This problem becomes evident when we consider the cost to create a new cluster:

$$d_{i,\kappa+1}^w = \frac{1}{2} \frac{\xi_0 W_i}{\xi_0 + W_i} (y_i - \mu_0)' \Sigma^{-1} (y_i - \mu_0) + \lambda_1.$$

The larger  $\kappa$ , the smaller  $|\Sigma|^{1/2}$ , the smaller the cost  $d_{i,\kappa+1}^w$  to create a new cluster, which leads to a cascade of more clusters getting created and so on.

To break this vicious cycle, we modify (5.5) by replacing the coefficient of  $\log |\Sigma^{1/2}|$  in (5.5) with

$$\sum_{i=1}^n W_i + \nu_0 - d - 1, \quad (5.7)$$

such that when sample size is small (and thus number of clusters  $\kappa$  is not huge),  $\Sigma$  is approximately equal to its inverse-Wishart prior mean  $\psi_0/(\nu_0-d-1)$ . This helps to ensure stability of the variable  $\Sigma$  (and thus the stability of the algorithm itself) especially when sample size is small, and also justifies initialization of  $\Sigma_w$  with  $\psi_0/(\nu_0-d-1)$  in the beginning of the RW SDP-rich algorithm. As sample size increases, the denominator of  $\Sigma$  is heavily influenced by the term  $\sum_{i=1}^n W_i$ , and so  $\Sigma$  will be approximately

$$\frac{\sum_{k=1}^{\kappa} \sum_{i:z_i=k} W_i (y_i - \mu_k)(y_i - \mu_k)'}{\sum_{i=1}^n W_i}.$$

### 5.2.2. Proof of Lemma 4.3 (Local convergence of RW SDP-rich)

This proof is an extension of the proof for Kulis and Jordan (2012)'s Theorem 3.1.

*Proof.* The reassignment step results in a non-increasing objective since the weighted Mahalanobis distance between a point and its newly-assigned weighted cluster centroid (discounted by the corresponding *rgr* “gravitational pull”) is smaller than that before the re-allocation occurs. If an observation is assigned to a new cluster, the cost of creating the new cluster is cheaper than to assign the observation to any one of the existing clusters, which results in a reduction in objective. Dropping empty cluster(s) – for example, dropping cluster  $\mathcal{C}_k$  – decreases the objective by  $\lambda_1 + \xi_0(\mu_k - \mu_0)' \Sigma^{-1}(\mu_k - \mu_0)$ . Similarly, the cluster parameter updates lead to a non-increasing objective since the objective function of the RW SDP-rich is convex in  $\mu$  and  $\Sigma$  conditional on  $(\kappa, z)$ . The algorithm will converge locally because the objective function cannot increase, and that there are only a finite number of possible clusterings of the data.  $\square$

### 5.2.3. Singleton clusters

The issue discussed in Section 5.1.1 is also applicable to the RW SDP-rich algorithm, except that now we have to replace Equation (5.1) with

$$\mu_{k'}^w = \frac{W_i y_i + \xi_0 \mu_0}{W_i + \xi_0},$$

and replace Equation (5.2) with

$$d_{ik}^w = \frac{1}{2} W_i (y_i - \mu_k)' \Sigma^{-1} (y_i - \mu_k) - \lambda_2 \log(n_{k',-i}) - \lambda_1,$$

and replace Equation (5.3) with

$$d_{ik'}^w = \frac{1}{2} \frac{\xi_0 W_i}{\xi_0 + W_i} (y_i - \mu_0)' \Sigma^{-1} (y_i - \mu_0).$$

#### 5.2.4. Initialization of algorithm

Again, the issues about the *rgr* penalty and initial cluster assignments, which are discussed in Section 5.1.2, are also relevant when we initialize the RW SDP-rich algorithm.

Furthermore, here we also need to consider about the issue regarding initialization of  $\Sigma$ .

Recall that  $\Sigma$  is updated with the formula

$$\frac{\sum_{k=1}^{\kappa} \sum_{i:z_i=k} W_i (y_i - \mu_k)(y_i - \mu_k)' + \xi_0 \sum_{k=1}^{\kappa} (\mu_k - \mu_0)(\mu_k - \mu_0)' + \psi_0}{(\sum_{i=1}^n W_i + \nu_0) - d - 1}.$$

When we first initialize the algorithm where all data points are grouped together, the corresponding initialized  $\Sigma$  will be approximately the overall weighted sum-of-squares

$$\frac{\sum_{i=1}^n W_i (y_i - \bar{\mu}_w)(y_i - \bar{\mu}_w)'}{\sum_{i=1}^n W_i},$$

where  $\bar{\mu}_w = (\sum_{i=1}^n W_i y_i + \xi_0 \mu_0) / (\sum_{i=1}^n W_i + \xi_0)$  represents the weighted grand centroid. We may be “overestimating” the dispersion among data points from the same cluster in this case. Hence, we suggest fixing  $\Sigma = \psi_0 / (\nu_0 - d - 1)$  (or, for the case of diagonal covariance structure,  $\sigma_j^2 = b_{0,j} / (a_{0,j} - 1)$  for all  $j$ ) during the first epoch of the RW SDP-rich algorithm to ensure more stable performance by the algorithm, based on our experience in the numerical experiments.

### 5.2.5. Formulae for diagonal covariance structure

Conditional on an existing partition  $(\kappa, \mathbf{z})$ , for  $j = 1 \cdots, d$ ,

$$\mu_{kj}^w = \frac{\sum_{i:z_i=k} W_i y_{ij} + \xi_{0j} \mu_{0j}}{\sum_{i:z_i=k} W_i + \xi_{0j}},$$

and

$$(\sigma_w^2)_j = \frac{\sum_{k=1}^{\kappa} \sum_{i:z_i=k} W_i (y_{ij} - \mu_{kj})^2 + \xi_{0j} \sum_{k=1}^{\kappa} (\mu_{kj} - \mu_{0j})^2 + 2b_{0j}}{(\sum_{i=1}^n w_i + 2a_{0j}) - 2},$$

and in the beginning of the algorithm, we initialize  $(\sigma_w^2)_j$  with  $b_{0j}/(a_{0j}-1)$ . In the cluster reassignment step, the cost  $d_{ik}^w$  of assigning observation  $y_i$  to an existing cluster  $\mathcal{C}_k$  is now

$$d_{ik}^w = \sum_{j=1}^d \frac{W_i (y_{ij} - \mu_{kj})^2}{2\sigma_j^2} - \lambda_{n,2} \log(n_k)$$

for  $k = 1, \dots, \kappa$ , whereas the cost to create a new cluster for observation  $y_i$  is

$$d_{i,\kappa+1}^w = \sum_{j=1}^d \frac{\xi_{0j} W_i}{\xi_{0j} + W_i} \frac{(y_{ij} - \mu_{0j})^2}{2\sigma_j^2} + \lambda_{n,1}.$$

Similarly, if  $i \in \mathcal{C}_{k'}$  where  $n_{k'} = 1$ , update  $\mu_{k'}^w$  as we have discussed in Section 5.2.3, and update

$$d_{ik}^w = \sum_{j=1}^d \frac{W_i (y_{ij} - \mu_{kj})^2}{2\sigma_j^2} - \lambda_{n,2} \log(n_{k',-i}) - \lambda_{n,1} \quad \text{for } k \in \{1, \dots, \kappa\} \setminus \{k'\},$$

$$d_{ik'}^w = \sum_{j=1}^d \frac{\xi_{0j} W_i}{\xi_{0j} + W_i} \frac{(y_{ij} - \mu_{0j})^2}{2\sigma_j^2},$$

$$d_{i,\kappa+1}^w = \infty.$$

### 5.2.6. Regularization parameters

#### Choice of $\lambda_2$

We now compare the performances of RW DP-rich and RW SDP-rich specified with different values of  $\lambda_2^{\text{rwDP-rich}}$  and  $\lambda_2^{\text{rwSDP-rich}}$  using a set of simulations. Specifically, we adopt the “full-covariance higher-correlation” Simulation Setting as well as its corresponding MCMC prior specifications, which we already outlined in the Main Text.

Here, we compare  $\lambda_2^{\text{rwDP-rich}} \in \{0, 0.5, 1, 2\}$  and  $\lambda_2^{\text{rwSDP-rich}} \in \{0, 0.5, 1, 2\}$ , using the different comparison criteria that we described in the Main Text. Recall that setting  $\lambda_2^{\text{rwDP-rich}} = 0$  corresponds to RW DP-means whereas specifying  $\lambda_2^{\text{rwSDP-rich}} = 0$  corresponds to RW SDP-means.

Figure 5.1 shows the performances of these different methods. Ideally, we want all performance criteria for these methods that involve total variation to be as close to zero as possible, which indicates higher degree of “similarity” to MCMC samples. Meanwhile, the average held-out log probability for these methods should be as high as possible, and the average of absolute difference in pairwise probability (of two observations clustered together) as compared to MCMC samples should be as close to zero as possible.

From Figure 5.1, it appears that setting  $\lambda_2^{\text{rwDP-rich}} = 1$  (denoted as `rwDPri ch2`) leads to the best performance in most of the comparison criteria among the RW DP-rich contenders (which is unsurprising, since the true variance of the mixture components is indeed equal to 1), whereas specifying  $\lambda_2^{\text{rwSDP-rich}} = 0.5$  (denoted as `rwSDPri ch1`) leads to the best performance in most of the comparison criteria among the RW SDP-rich candidates. In particular, the boxplots for RW SDP-rich with  $\lambda_2 = 2$  (denoted as `rwSDPri ch3`) are not shown in Figure 5.1 because their performances are the worst (way worse than all the other methods).

#### Calibrating $\lambda_1$

Here, we use a Binary Search procedure (e.g. Raykov et al., 2016) to tune  $\lambda_1$  such that the (average of) random-weighting samples of  $\kappa$  “matches” a targeted number of clusters



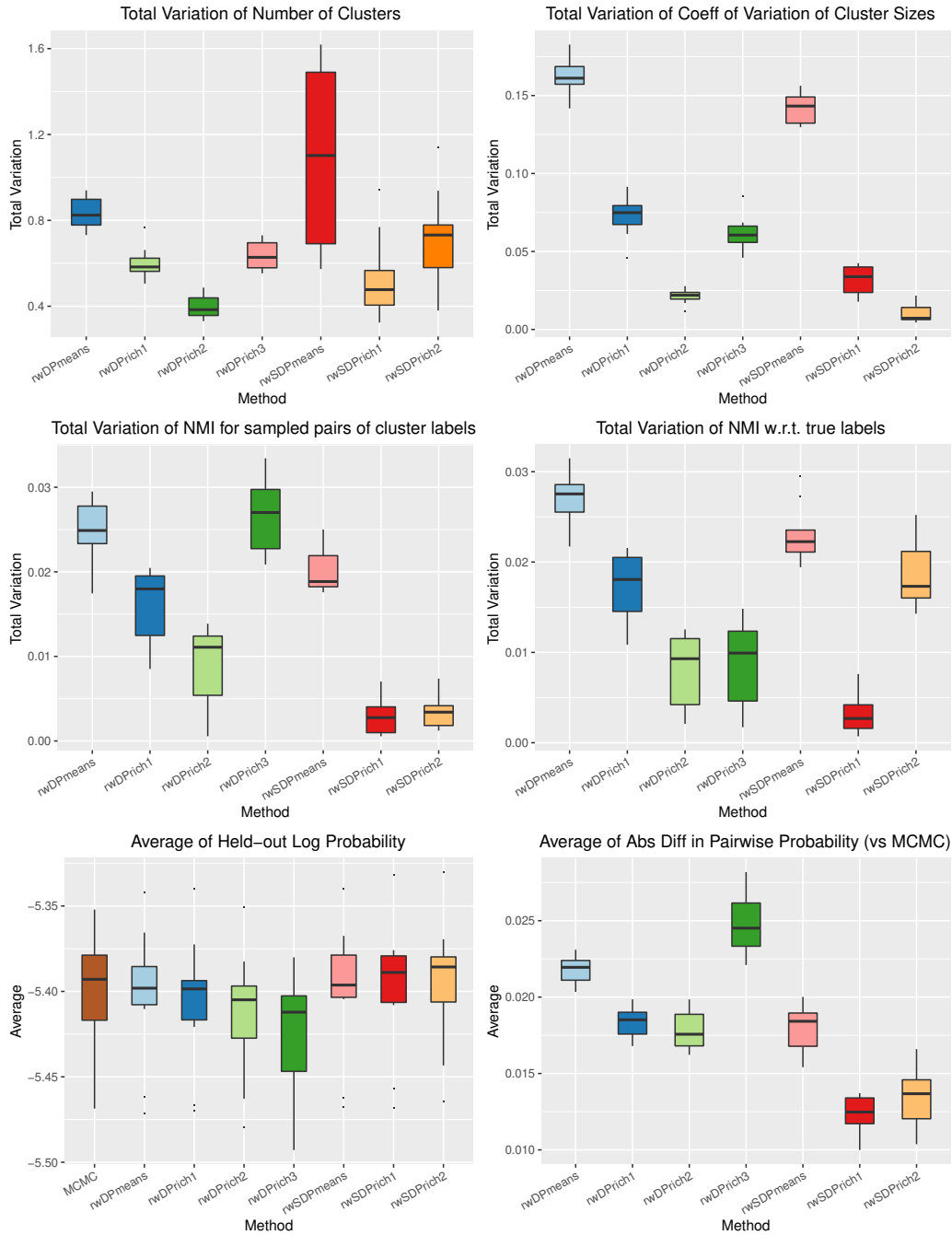


Figure 5.1: Comparing performances of RW DP-rich and RW SDP-rich using different  $rgr$  tuning parameters. For RW DP-rich, we specify  $\lambda_2^{\text{rWDP-rich}}$  to be 0 (denoted rWDPmeans), 0.5 (denoted rWDPrich1), 1 (denoted rWDPrich2) and 2 (denoted rWDPrich3). For RW SDP-rich, we specify  $\lambda_2^{\text{rWSDP-rich}}$  to be 0 (denoted rWSDPmeans), 0.5 (denoted rWSDPrich1) and 1 (denoted rWSDPrich2).

$K_{\text{targ}}$  for any one of the four random-weighting algorithms: RW DP-means, RW DP-rich, RW SDP-means or RW SDP-rich. In our numerical experiments, we specify  $K_{\text{targ}}$  to be the posterior mean of  $\kappa$  obtained from standard MCMC method, which (in most cases of our numerical experiments) are also close to the MAP of  $\kappa$ .

Briefly, we start off by picking a starting value of  $\lambda_1^{(0)}$  via, say, a farthest-first approach (e.g. Kulis & Jordan, 2012). Next, we generate and store  $B'$  sets of i.i.d. standard Exponential random weights  $\{\mathbf{W}_b^{1:n}\}_{1 \leq b \leq B'} := \{(W_{1,b}, \dots, W_{n,b})\}_{1 \leq b \leq B'}$ . For calibration purpose, we could use a ‘‘cheaper’’ random-weighting scheme by using a smaller number of draws, say,  $B' = 1000$  in order to save computational time, and yet perform reasonably well in our numerical experiments.

Each  $b = 1, \dots, B'$  set of these random weights  $\mathbf{W}_b^{1:n}$  is then fed into the random-weighting procedure specified with  $\lambda_1 = \lambda_1^{(0)}$  to obtain  $\{\kappa_b^w\}_{1 \leq b \leq B'}$ . Next, we compute the average of  $\kappa_b^w - K_{\text{targ}}$

$$d_{\lambda_1^{(0)}} := \frac{1}{B'} \sum_{b=1}^{B'} (\kappa_b^w - K_{\text{targ}}). \quad (5.8)$$

If  $d_{\lambda_1^{(0)}} > 0$ , this indicates that on average, there are more clusters obtained by the random-weighting procedure than  $K_{\text{targ}}$ , i.e. we want *less* clusters, and thus we need to scale up  $\lambda_1$  to inflict a heavier penalty on  $\kappa$  in the objective function. Therefore, we set  $\underline{\lambda}_1 = \lambda_1^{(0)}$  to be the lower bound of  $\lambda_1$ , and specify the next potential tuning parameter value  $\lambda_1^{(1)} = 2 \times \lambda_1^{(0)}$ .

On the other hand, if  $d_{\lambda_1^{(0)}} < 0$ , we want *more* clusters, and thus we need to reduce  $\lambda_1$  to inflict less penalty on  $\kappa$  in the objective function. Therefore, we set  $\bar{\lambda}_1 = \lambda_1^{(0)}$  to be the upper bound of  $\lambda_1$ , and specify the next potential tuning parameter value  $\lambda_1^{(1)} = \frac{\lambda_1^{(0)}}{2}$ . We substitute  $\lambda_1^{(1)}$  into the random-weighting procedure and repeat the steps above to obtain  $d_{\lambda_1^{(1)}}$  and so on.

Suppose after the  $(t - 1)^{\text{th}}$  step, we have our lower bound  $\underline{\lambda}_1$  and upper bound  $\bar{\lambda}_1$ . Then,

we could specify

$$\lambda_1^{(t)} = \frac{\lambda_1 + \bar{\lambda}_1}{2} \quad (5.9)$$

and subsequently obtain  $d_{\lambda_1^{(t)}}$ . If  $d_{\lambda_1^{(t)}} < 0$ , update the lower bound  $\lambda_1 = \lambda_1^{(t)}$  and set  $\lambda_1^{(t+1)}$  via the same formula (5.9) using this new lower bound. On the other hand, if  $d_{\lambda_1^{(t)}} > 0$ , update the upper bound  $\bar{\lambda}_1 = \lambda_1^{(t)}$ , and compute  $\lambda_1^{(t+1)}$  with (5.9) using this new upper bound. Repeat this process until  $|d_{\lambda_1}| \leq \epsilon_\lambda$ , where  $\epsilon_\lambda$  is some minute tolerance level determined by the analyst.

Alternatively, we can also use this Binary Search procedure to nail down a reasonable range of  $[\lambda_1, \bar{\lambda}_1]$ , so that we can then use a grid search approach to find the value of  $\lambda_1$  that produces the smallest  $|d_{\lambda_1}|$ .

### 5.3. Additional details of RW K-means

The RW K-means procedure mentioned in the main text is outlined in Algorithm 5. In particular, the standard K-means optimization procedures, such as that by Hartigan and Wong (1979), could still be used to optimize the RW K-means objective function, except that now, weighted Euclidean distance is considered in the cluster reassignment step and weighted centroids are updated instead. Arthur and Vassilvitskii (2007)'s discussion about careful seeding is also relevant here to improve the local solutions obtained by the RW K-means procedure.

While the regular K-means has long been known to be the small-variance asymptotics (SVA) of the Gaussian finite-mixture model (GMM) (e.g., Hastie et al., 2009), we verify in Lemma 5.1 that this SVA property remains applicable to their random-weighting counterparts.

**Lemma 5.1 (RW K-means as the SVA of RW GMM).** *For a Gaussian finite mixture with common variance  $\Sigma_k = \sigma^2 I_d \forall k = 1, \dots, K$ , the negative loglikelihood of its random-weighting*

---

**Algorithm 5** Random-weighting K-means
 

---

**Require:** data  $\{y_1, \dots, y_n\}$ , number of clusters  $K$ , number of posterior draws  $B$ .

- 1: **for**  $b = 1, \dots, B$  **do**
- 2:   Initialize with  $K$  centroids.
- 3:   Draw  $W_i \stackrel{iid}{\sim} Exp(1) \forall i = 1, \dots, n$ .
- 4:   Optimize the RW K-means objective function, and store  $\mu_{k,b}^w$  for  $k = 1, \dots, K$ , and  $z_{i,b}^w$  for  $i = 1, \dots, n$ .
- 5: **end for**

**Ensure:**  $B$  samples of cluster centroids  $\{\mu_{k,b}^w\}_{1 \leq k \leq K; 1 \leq b \leq B}$ , and  $B$  samples of cluster assignments  $\{z_{i,b}^w\}_{1 \leq i \leq n; 1 \leq b \leq B}$ .

---

counterpart (Fong et al., 2019)

$$\prod_{k=1}^K \prod_{i:z_i=k} [p_k f_k(y_i | \mu_k, \Sigma_k)]^{W_i}, \quad (5.10)$$

multiplied with  $2\sigma^2$ , converges to the objective function of random-weighting K-means (4.19) when we push  $\sigma^2 \rightarrow 0$ .

**Proof of Lemma 5.1.** Taking negative log of (5.10) where  $\Sigma_k = \sigma^2 I_d \forall k = 1, \dots, K$ , we have

$$-\sum_{k=1}^K (\log p_k) \sum_{i:z_i=k} W_i + \sum_{k=1}^K \sum_{i:z_i=k} W_i \left[ \frac{d}{2} \log \sigma^2 + \frac{\|y_i - \mu_k\|_2^2}{2\sigma^2} \right] + \frac{d \log(2\pi)}{2} \sum_{i=1}^n W_i. \quad (5.11)$$

Multiply (5.11) with  $2\sigma^2$ , then push  $\sigma^2 \rightarrow 0$  to obtain the objective function in (4.19).  $\square$

## 5.4. Additional details for theoretical properties

### 5.4.1. Probability space

There are two sources of variation in the random-weighting setup under the Bayesian NPL framework, namely the data  $\{y_1, y_2, \dots\}$  and the random weights  $\{w_1, w_2, \dots\}$ . Consequently, we consider a common probability space with the common probability measure  $P = P_{F_*}^{(\infty)} \times P_{\tilde{F}_w}$ , where  $P_{F_*}^{(\infty)}$  is the probability measure of the observed data, and  $P_{\tilde{F}_w}$

is the probability measure of the triangular array of random weights (Mason & Newton, 1992) that arises from Bayesian bootstrap  $\tilde{F}_W$ . The use of product measure reflects the independence of data and random weights. The study of asymptotic properties under the random-weighting framework is not new; see, for example, Mason and Newton (1992), Lyddon et al. (2019) and Ng and Newton (2020).

#### 5.4.2. Derivation for Bayesian NPL framework

Under the Bayesian NPL framework (Fong et al., 2019), since  $F_*$  is unknown, we place a Dirichlet process (DP) prior on the sampling distribution

$$F | (\alpha_0, F_0) \sim DP(\alpha_0, F_0), \quad (5.12)$$

where  $\alpha_0$  is the concentration parameter and  $F_0$  is the prior centering measure. We want to remind readers that  $F_*$  is not required to be in some “neighborhood” of  $F_0$ , and  $DP(\alpha_0, F_0)$  in (5.12) is NOT related to the DPM working model that we mentioned in the main text.

From the conjugacy of the DP (e.g., Ghosal & van der Vaart, 2017), the posterior of  $F$  becomes

$$F | y := \tilde{F} \sim DP \left( \alpha_0 + n, \frac{\alpha_0}{\alpha_0 + n} F_0 + \frac{1}{\alpha_0 + n} \sum_{i=1}^n \delta_{y_i} \right), \quad (5.13)$$

where  $\delta$  denotes the dirac measure. Based on the stick-breaking construction (Sethuraman, 1994) of the DP, we have

$$\arg \min_{t \in \Theta} \mathcal{L}(t, \tilde{F}) = \arg \min_{t \in \Theta} \int l(t, y) d\tilde{F}(y) = \arg \min_{t \in \Theta} \left\{ \sum_{j=1}^{\infty} \check{w}_j l(t, \check{y}_j) \right\}, \quad (5.14)$$

where  $\{\check{w}_j\}_{j=1}^{\infty} \sim GEM(\alpha_0 + n)$  and

$$\check{y}_j \stackrel{iid}{\sim} \left( \frac{\alpha_0}{\alpha_0 + n} F_0 + \frac{1}{\alpha_0 + n} \sum_{i=1}^n \delta_{y_i} \right)$$

for all  $j$  (Ishwaran & Zarepour, 2002). Exact posterior calculation of (5.14) requires infinite sampling, but could be approximated with

$$\arg \min_{t \in \Theta} \left\{ \sum_{i=1}^n w_i l(t, y_i) + \sum_{j=1}^T \tilde{w}_j l(t, \tilde{y}_j) \right\} \quad (5.15)$$

for large truncation limit  $T$ , where  $\tilde{y}_j \stackrel{iid}{\sim} F_0$  and

$$(w_1, \dots, w_n, \tilde{w}_1, \dots, \tilde{w}_T) \sim Dir(1, \dots, 1, \alpha_0/T, \dots, \alpha_0/T).$$

As  $n \rightarrow \infty$  such that  $n \gg T$ , data realizations overwhelm prior information, which motivates Rubin (1981)'s Bayesian bootstrap approximation of  $\tilde{F}$  (henceforth denoted as  $F_w$ ) by setting  $\alpha_0 = 0$  in (5.15):

$$\mathcal{L}(t, F_w) = \int_{\Omega} \tilde{l}(t, y) dF_w(y) = \int_{\Omega} l(t, y) dF_w(y) + \lambda_0 l_0(t) = \sum_{i=1}^n w_i l(t, y_i) + \lambda_0 l_0(t), \quad (5.16)$$

where  $(w_1, \dots, w_n) \sim Dir(1, \dots, 1)$ . See also Muliere and Secchi (1996) on further interpretations of the Bayesian bootstrap. Since

$$(w_1, \dots, w_n) \stackrel{d}{=} \left( \frac{W_1}{\sum_{i=1}^n W_i}, \dots, \frac{W_n}{\sum_{i=1}^n W_i} \right)$$

where  $W_i \stackrel{iid}{\sim} Exp(1)$ , solving  $\min_{t \in \Theta} \mathcal{L}(t, F_w)$  in (5.16) is equivalent to optimizing

$$\min_{t \in \Theta} \left\{ \sum_{i=1}^n [W_i \cdot l(t, y_i)] + \left( \lambda_0 \sum_{i=1}^n W_i \right) \cdot l_0(t) \right\}.$$

In practice, we replace the  $(\lambda_0 \sum_{i=1}^n W_i)$  term with a unifying regularization parameter  $\lambda > 0$  to be calibrated by the analyst, and finally we arrive at (4.2).

### 5.4.3. Collection of proofs for Chapter 4.4

**Proof of Lemma 4.6.** The decision-boundaries of a *Voronoi partition* are linear discriminant functions under the RW K-means, RW DP-means or RW SDP-means (with a fixed  $\Sigma_0$ ) setup. Simple rank-nullity exercise reveals that these decision boundaries have  $(d - 1)$  dimensions, and thus have measure zero due to absolute continuity of  $F_*$ .  $\square$

Before we prove Theorem 4.4, we need the following results.

**Lemma 5.2 (Finiteness of asymptotic limits).** *Adopt assumptions in Theorem 4.4. Then,*

(a)  $\kappa_{*,\lambda_0} < \infty$  and  $\kappa_{*,(\lambda_0,\Sigma_0)} < \infty$ .

(b) all centroids in  $A_{*,K}$ ,  $A_{*,\lambda_0}$  and  $A_{*,(\lambda_0,\Sigma_0)}$  are finite.

**Proof of Lemma 5.2.** The finite second moment requirement on  $F_*$  leads to

$$\int_{\Omega} \|y\|_2^2 dF_*(y) < \infty \quad \text{and} \quad \int_{\Omega} y \Sigma_0^{-1} y \, dF_*(y) < \infty$$

for any symmetric positive definite  $\Sigma_0$ . Then, for any point  $r \in \mathbb{R}^d$ ,

$$\int_{\Omega} \min_{a \in A_K} \|y - a\|_2^2 \, dF_*(y) \leq \int_{\Omega} \|y - r\|_2^2 \, dF_*(y) \leq 4\|r\|_2^2 + 4 \int_{\Omega} \|y\|_2^2 \, dF_*(y), \quad (5.17)$$

where the RHS of (5.17) is finite only if point  $r$  is finite. Thus, all centroids in  $A_{*,K}$  have to be finite, otherwise contradiction occurs. Similarly, for any point  $r \in \mathbb{R}^d$  and for any symmetric positive-definite  $\Sigma_0$ ,

$$\begin{aligned} \int_{\Omega} \min_{a \in A_K} [(y - a)' \Sigma_0^{-1} (y - a)] \, dF_*(y) &\leq \int_{\Omega} (y - r)' \Sigma_0^{-1} (y - r) \, dF_*(y) \\ &\leq 4r' \Sigma_0^{-1} r + 4 \int_{\Omega} y' \Sigma_0^{-1} y \, dF_*(y), \end{aligned} \quad (5.18)$$

where the second line of (5.18) is finite only if point  $r$  is finite. We remind the readers that for RW DP-means and RW SDP-means,  $\kappa$  is data-driven instead of being pre-specified by the analyst, but we need (5.17) and (5.18) to prove part (a) of the lemma.

For RW DP-means, we have, from (5.17),

$$\int_{\Omega} \min_{a \in A} \|y - a\|_2^2 dF_*(y) = \mathcal{O}(1)$$

for any partition  $\mathcal{P}$  of  $\Omega$  that is associated with the set of centroids  $A = \{a_1, \dots, a_{\kappa}\}$  where  $\kappa = |A|$ . Increasing the number of clusters indefinitely shrinks the integral to 0 but increases  $\kappa \rightarrow \infty$ , which in turn pushes the objective to  $\infty$ . Thus, a minimizer must arrive at finite  $\kappa_{*,\lambda_0}$ .

Next, since  $\kappa_{*,\lambda_0} < \infty$ , we could think of minimizing

$$\left\{ \int_{\Omega} \min_{a \in A_{\lambda_0}} \|y - a\|_2^2 dF_*(y) + \lambda_0 \kappa \right\}$$

as minimizing the LHS of (5.17) on the grid of positive integers  $\mathbb{N}$ , and evaluate the corresponding objectives penalized with  $\lambda_0 K$  for  $K = 1, \dots, \kappa_{*,\lambda_0}^{\max}$  where  $\kappa_{*,\lambda_0} \leq \kappa_{*,\lambda_0}^{\max}$ . Finally, pick the clustering that has the lowest objective. Using similar argument in (5.17), we ensure that all the centroids in  $A_{*,\lambda_0}$  are finite.

Finally, For RW SDP-means with a fixed symmetric positive-definite  $\Sigma_0$ , we could use (5.18) and similar arguments to establish that  $\kappa_{*,(\lambda_0,\Sigma_0)} < \infty$ , and that all centroids in  $A_{*,(\lambda_0,\Sigma_0)}$  are finite.  $\square$

**Lemma 5.3 (Continuity for sets of centroids).** *Adopt assumptions in Theorem 4.4. Then,*

- (a)  $\mathcal{L}(A_K, F_*)$  is a continuous function of  $A_K$  in (4.39).
- (b)  $\mathcal{L}(A_{\lambda_0}, F_*)$  is a continuous function of  $A_{\lambda_0}$  in (4.41).
- (c)  $\mathcal{L}(A_{(\lambda_0,\Sigma_0)}, F_*)$  is a continuous function of  $A_{(\lambda_0,\Sigma_0)}$  in (4.43).

**Proof of Lemma 5.3.** To prove continuity, we need to first invoke Lemma 5.2 to ensure that  $\kappa_{*,\lambda_0} < \infty$  and  $\kappa_{*,(\lambda_0,\Sigma_0)} < \infty$ , and all centroids in  $A_{*,K}$ ,  $A_{*,\lambda_0}$  and  $A_{*,(\lambda_0,\Sigma_0)}$  are finite. Then, we immediately obtain part (a) of the Lemma by invoking the established result in Pollard (1981) about the continuity of  $\mathcal{L}(A_K, F_*)$  as a function of  $A_K$  in (4.39).



To obtain part (b), we still need to construct a similar  $\epsilon - \zeta$  proof; i.e., for every  $\epsilon > 0$ , there exists  $\zeta_\epsilon > 0$  such that  $\mathcal{D}_{\mathcal{H}}(A_{\lambda_0}, A'_{\lambda_0}) < \zeta_\epsilon$  implies  $|\mathcal{L}(A_{\lambda_0}, F_*) - \mathcal{L}(A'_{\lambda_0}, F_*)| < \epsilon$ . First, note that if  $|A_{\lambda_0}| \neq |A'_{\lambda_0}|$ , there is always another set of centroids  $A''_{\lambda_0}$  such that  $|A_{\lambda_0}| = |A''_{\lambda_0}|$  and  $|\mathcal{L}(A_{\lambda_0}, F_*) - \mathcal{L}(A''_{\lambda_0}, F_*)| < |\mathcal{L}(A_{\lambda_0}, F_*) - \mathcal{L}(A'_{\lambda_0}, F_*)|$ , because  $A''_{\lambda_0}$  can be exactly the same as  $A_{\lambda_0}$  except for one of its centroids, where its coordinates are slightly perturbed such that the resulting change in the sum of squares of its cluster is smaller than  $|\mathcal{L}(A_{\lambda_0}, F_*) - \mathcal{L}(A'_{\lambda_0}, F_*)|$ . Hence, we only need to consider the case where  $|A_{\lambda_0}| = |A'_{\lambda_0}|$  in the  $\epsilon - \zeta$  proof, and Pollard (1981)'s continuity result immediately follows.

Similar arguments can also be applied to prove part (c) by replacing the Euclidean distance with the Mahalanobis distance.  $\square$

**Proof of Theorem 4.4.** We first invoke Lemma 5.3 to establish that  $\mathcal{L}(A, F_*)$  is a continuous function of  $A$ , where  $A$  stands for  $A_K$ ,  $A_{\lambda_0}$  and  $A_{(\lambda_0, \Sigma_0)}$  in (4.39), (4.41) and (4.43) respectively. Then, by noticing that  $A_K$ ,  $A_{\lambda_0}$  and  $A_{(\lambda_0, \Sigma_0)}$  are deterministic functionals of  $F_w$ , whereas  $A_{*,K}$ ,  $A_{*,\lambda_0}$  and  $A_{*,(\lambda_0, \Sigma_0)}$  are deterministic functionals of  $F_*$ , the convergence of the posterior distribution  $\Pi_n$  of  $A_K$ ,  $A_{\lambda_0}$  and  $A_{(\lambda_0, \Sigma_0)}$  follows immediately from (4.38).  $\square$

Before we prove Theorem 4.5, we need the following results.

**Lemma 5.4 (Continuity for partitions).** *Adopt assumptions in Theorem 4.5. Then,*

- (a)  $\mathcal{L}(\mathcal{P}_K(A_K), F_*)$  is a continuous function of  $\mathcal{P}_K$ , where  $\mathcal{P}_K$  is the Voronoi partition associated with  $A_K$  in (4.39).
- (b)  $\mathcal{L}(\mathcal{P}_{\lambda_0}(A_{\lambda_0}), F_*)$  is a continuous function of  $\mathcal{P}_{\lambda_0}$ , where  $\mathcal{P}_{\lambda_0}$  is the Voronoi partition associated with  $A_{\lambda_0}$  in (4.41).
- (c)  $\mathcal{L}(\mathcal{P}_{(\lambda_0, \Sigma_0)}(A_{(\lambda_0, \Sigma_0)}), F_*)$  is a continuous function of  $\mathcal{P}_{(\lambda_0, \Sigma_0)}$ , where  $\mathcal{P}_{(\lambda_0, \Sigma_0)}$  is the Voronoi partition associated with  $A_{(\lambda_0, \Sigma_0)}$  in (4.43).

**Proof of Lemma 5.4.** Again, due to Lemma 5.2, we ensure that  $\kappa_{*,\lambda_0} < \infty$  and  $\kappa_{*,(\lambda_0, \Sigma_0)} < \infty$ , and all centroids in  $A_{*,K}$ ,  $A_{*,\lambda_0}$  and  $A_{*,(\lambda_0, \Sigma_0)}$  are finite. Then, we need to construct an

$\epsilon - \zeta$  proof for continuity; i.e., for every  $\epsilon > 0$ , there exists  $\zeta_\epsilon > 0$  such that  $\mathcal{D}_{\mathcal{L}}(\mathcal{P}, \mathcal{P}') < \zeta_\epsilon$  implies  $|\mathcal{L}(\mathcal{P}(A), F_*) - \mathcal{L}(\mathcal{P}'(A'), F_*)| < \epsilon$ , where  $\mathcal{P}$  and  $\mathcal{P}'$  have their respective subscripts stipulated in parts (a), (b) and (c) of the Lemma. However, note that  $\mathcal{P}$  and  $\mathcal{P}'$  are *Voronoi partitions* that are associated with specific sets of centroids  $A$  and  $A'$ . That is, based on  $\mathcal{P}$  and  $\mathcal{P}'$ , we could compute both  $\mathcal{D}_{\mathcal{L}}(\mathcal{P}, \mathcal{P}')$  and  $\mathcal{D}_{\mathcal{H}}(A, A')$ , and that we actually compute  $|\mathcal{L}(A, F_*) - \mathcal{L}(A', F_*)|$ .

To understand the following arguments, we require readers to understand Section 3.1 of Leonardi and Tamanini (2002). Now, for part (a) of the Lemma, it is clear that two partitions  $\mathcal{P}_K$  and  $\mathcal{P}'_K$  are “close” to each other (i.e.  $\mathcal{D}_{\mathcal{L}}(\mathcal{P}_K, \mathcal{P}'_K)$  is small) only if every cluster  $V_k$  in  $\mathcal{P}_K$  largely overlaps its counterpart  $V'_k$  in  $\mathcal{P}'_K$  (the notion of “counterpart” makes sense here, because Leonardi and Tamanini (2002)’s metric  $\mathcal{D}_{\mathcal{L}}$  actually considers the minimum of the Lebesgue measure of non-overlapping regions for every permutation of the clusters in the two partitions). High degree of overlapping occurs when the cluster centroid  $a_k$  of  $V_k$  is close to the centroid  $a'_k$  of  $V'_k$  for all  $k = 1, \dots, K$ ; i.e.,  $\mathcal{D}_{\mathcal{H}}(A_K, A'_K)$  is small when  $\mathcal{D}_{\mathcal{L}}(\mathcal{P}_K, \mathcal{P}'_K)$  is small. Hence, there is only an additional layer to be inserted in the  $\epsilon - \zeta$  proof: for every  $\epsilon > 0$ , pick partition  $\mathcal{P}'_K$  that has  $\mathcal{D}_{\mathcal{L}}(\mathcal{P}_K, \mathcal{P}'_K) < \zeta_{\mathcal{P}}(\zeta_\epsilon)$  such that their corresponding sets of centroids have  $\mathcal{D}_{\mathcal{H}}(A_K, A'_K) < \zeta_\epsilon$ , then plug in Pollard (1981)’s proof for continuity to ensure that  $|\mathcal{L}(\mathcal{P}_K(A_K), F_*) - \mathcal{L}(\mathcal{P}'_K(A'_K), F_*)| = |\mathcal{L}(A_K, F_*) - \mathcal{L}(A'_K, F_*)| < \epsilon$ .

For part (b), we can deploy a similar argument in the proof for part (b) of Lemma 5.3 to restrict our consideration to partitions  $\mathcal{P}_{\lambda_0}$  and  $\mathcal{P}'_{\lambda_0}$  with the same number of clusters. Then, the result for part (b) immediately follows from part (a). Part (c) is also the same, except that now Mahalanobis distance is involved.  $\square$

**Proof of Theorem 4.5.** We first invoke Lemma 5.4 to establish that  $\mathcal{L}(\mathcal{P}(A), F_*)$  is a continuous function of  $\mathcal{P}$ , where  $\mathcal{P}$  stands for the *Voronoi partitions*  $\mathcal{P}_K$ ,  $\mathcal{P}_{\lambda_0}$  and  $\mathcal{P}_{(\lambda_0, \Sigma_0)}$  that are associated with  $A_K$ ,  $A_{\lambda_0}$  and  $A_{(\lambda_0, \Sigma_0)}$  in (4.39), (4.41) and (4.43) respectively. Then, by noticing that  $\mathcal{P}_K$ ,  $\mathcal{P}_{\lambda_0}$  and  $\mathcal{P}_{(\lambda_0, \Sigma_0)}$  are deterministic functionals of  $F_w$ , whereas  $\mathcal{P}_{*,K}$ ,  $\mathcal{P}_{*,\lambda_0}$  and  $\mathcal{P}_{*,(\lambda_0, \Sigma_0)}$  are deterministic functionals of  $F_*$ , the convergence of the posterior distribution  $\Pi_n$  of  $\mathcal{P}_K$ ,  $\mathcal{P}_{\lambda_0}$  and  $\mathcal{P}_{(\lambda_0, \Sigma_0)}$  follows immediately from (4.38).  $\square$

**Proof of Theorem 4.7.** We first consider the cases for RW K-means, RW DP-means and RW DP-rich. Notice that

$$\begin{aligned}\sqrt{n_k} (\mu_{n,k}^w - \hat{\mu}_{n,k}) &= \frac{\sqrt{n_k}}{\sum_{i \in \mathcal{C}_k^0} W_i} \left[ \sum_{i \in \mathcal{C}_k^0} W_i y_i - \left( \sum_{i \in \mathcal{C}_k^0} W_i \right) \hat{\mu}_{n,k} \right] \\ &= \frac{1}{\bar{W}_{n,k}} \cdot \frac{1}{\sqrt{n_k}} \sum_{i \in \mathcal{C}_k^0} W_i (y_i - \hat{\mu}_{n,k})\end{aligned}$$

where  $\bar{W}_{n,k} = \frac{1}{n_k} \sum_{i \in \mathcal{C}_k^0} W_i \xrightarrow{a.s.} 1$  as  $n_k \rightarrow \infty$ . Then, for every  $z \in \mathbb{R}^d$ ,

$$\begin{aligned}z' [\sqrt{n_k} (\mu_{n,k}^w - \hat{\mu}_{n,k})] &= \frac{1}{\bar{W}_{n,k}} \cdot \frac{1}{\sqrt{n_k}} \sum_{i \in \mathcal{C}_k^0} W_i [z'(y_i - \hat{\mu}_{n,k})] \\ &= \frac{1}{\bar{W}_{n,k}} \cdot \sqrt{\frac{1}{n_k} \sum_{i \in \mathcal{C}_k^0} [z'(y_i - \hat{\mu}_{n,k})]^2} \cdot \frac{\sum_{i \in \mathcal{C}_k^0} [z'(y_i - \hat{\mu}_{n,k})] W_i}{\sqrt{\sum_{i \in \mathcal{C}_k^0} [z'(y_i - \hat{\mu}_{n,k})]^2}},\end{aligned}$$

where

$$\begin{aligned}\frac{1}{n_k} \sum_{i \in \mathcal{C}_k^0} [z'(y_i - \hat{\mu}_{n,k})]^2 &= z' \left( \frac{1}{n_k} \sum_{i \in \mathcal{C}_k^0} [y_i - \hat{\mu}_{n,k}] [y_i - \hat{\mu}_{n,k}]' \right) z \\ &\rightarrow z' (V_{*,k}^{\mathcal{P}_0}) z \quad a.s. \quad P_{F_*}^{(\infty)},\end{aligned}$$

and

$$\begin{aligned}\mathbb{E} \left( \sum_{i \in \mathcal{C}_k^0} [z'(y_i - \hat{\mu}_{n,k})] W_i \middle| y \right) &= \sum_{i \in \mathcal{C}_k^0} [z'(y_i - \hat{\mu}_{n,k})] = 0, \\ \text{Var} \left( \sum_{i \in \mathcal{C}_k^0} [z'(y_i - \hat{\mu}_{n,k})] W_i \middle| y \right) &= \sum_{i \in \mathcal{C}_k^0} [z'(y_i - \hat{\mu}_{n,k})]^2 = \mathcal{O}(n), \\ \mathbb{E} \left( \sum_{i \in \mathcal{C}_k^0} [z'(y_i - \hat{\mu}_{n,k})]^4 W_i^4 \middle| y \right) &= \sum_{i \in \mathcal{C}_k^0} [z'(y_i - \hat{\mu}_{n,k})]^4 \mathbb{E} (W_i^4) = \mathcal{O}(n),\end{aligned}$$

where the last two lines follow from the finite (second and) fourth moment(s) property of the standard-Exponentially-distributed random weights. Hence, the Liapounov's sufficient

condition is satisfied to apply the Lindeberg's Central Limit Theorem (coupled with Slutsky's Theorem) to obtain

$$P(z' [\sqrt{n_k} (\mu_{n,k}^w - \hat{\mu}_{n,k})] \in B | y) \rightarrow P(Z_1 \in B) \quad a.s. \ P_{F_*}^{(\infty)}$$

for any Borel set  $B \subset \mathbb{R}$  and  $Z_1 \sim N(0, z' (V_{*,k}^{\mathcal{P}_0}) z)$ . Finally, apply the Cramer-Wold device to obtain the result.

For RW SDP-means and RW SDP-rich, notice that

$$\begin{aligned} \sqrt{n_k} (\mu_{n,k}^w - \hat{\mu}_{n,k}) &= \frac{\sqrt{n_k}}{\sum_{i \in \mathcal{C}_k^0} W_i + \xi_0} \left[ \sum_{i \in \mathcal{C}_k^0} W_i y_i - \left( \sum_{i \in \mathcal{C}_k^0} W_i + \xi_0 \right) \hat{\mu}_{n,k} + \xi_0 \mu_0 \right] \\ &= \frac{\sum_{i \in \mathcal{C}_k^0} W_i}{\sum_{i \in \mathcal{C}_k^0} W_i + \xi_0} \cdot \frac{1}{\bar{W}_{n,k}} \cdot \frac{1}{\sqrt{n_k}} \left[ \sum_{i \in \mathcal{C}_k^0} W_i (y_i - \hat{\mu}_{n,k}) + \xi_0 (\mu_0 - \hat{\mu}_{n,k}) \right]. \end{aligned}$$

The first term converges to one almost surely as  $n_k$  increases, whereas the extra term

$$\frac{\xi_0 (\mu_0 - \hat{\mu}_{n,k})}{\sqrt{n_k}} \rightarrow 0 \quad a.s. \ P_{F_*}^{(\infty)}.$$

The rest of the terms are the same from before, and the result follows from applying Slutsky's theorem to deal with these extra terms.  $\square$

## 5.5. Additional information for numerical experiments

### 5.5.1. Additional comparison for simulations

First, we provide the formula for the computation of average log posterior predictive density (Comparison Criterion 2) under the diagonal-covariance structure:

$$\tilde{g}_{(\cdot)}^{(t)} := \frac{1}{m} \sum_{\tilde{i}=1}^m \log \left\{ \frac{1}{B} \sum_{b=1}^B \left[ \sum_{k=1}^{\kappa_{(\cdot)}^{(b,t)}} \frac{n_{k(\cdot)}^{(b,t)}}{n + \alpha_0} \prod_{j=1}^d f_{T_1} \left( \tilde{y}_{\tilde{i}j}^{(t)} \mid \tilde{\nu}_{kj(\cdot)}^{(b,t)}, \tilde{\mu}_{kj(\cdot)}^{(b,t)}, \tilde{\sigma}_{kj(\cdot)}^{(b,t)} \right) \right] \right\}$$

$$+ \frac{\alpha_0}{n + \alpha_0} \prod_{j=1}^d f_{T_1} \left( \tilde{y}_{i_j}^{(t)} \mid \tilde{\nu}_{0j}, \tilde{\mu}_{0j}, \tilde{\sigma}_{0j} \right) \Bigg\},$$

where  $f_{T_1}(y|\nu, \mu, \sigma)$  denotes the univariate  $T$  density (with  $\nu$  degrees of freedom as well as location and scale parameters of  $\mu$  and  $\sigma$ ) evaluated at  $y$ . Again, the formula for these univariate  $T$  densities follow that of the posterior predictive density corresponding to a conjugate normal-inverse-gamma prior.

Next, we consider an additional comparison criterion for our simulation results: **NMI w.r.t. ground truth cluster labels**. Specifically, we compute the NMI (Vinh et al., 2010) value that compares the  $b^{th}$  posterior draw of cluster assignments and the ground-truth cluster labels for the  $t^{th}$  simulated training data set

$$\eta_{(\cdot)}^{(b,t)} := \text{NMI} \left( \mathbf{z}_{(\cdot)}^{(b,t)}, \mathbf{z}_{(\text{truth})}^{(t)} \right),$$

and then plot the boxplots for the mean of these NMI's from each of these 6 methods for  $t = 1, \dots, T$  datasets

$$\bar{\eta}_{(\cdot)}^{(t)} = \frac{1}{B} \sum_{b=1}^B \eta_{(\cdot)}^{(b,t)}. \quad (5.19)$$

Basically, we want to compare how well these methods in “recovering” the true cluster partition (one for perfect recovery of true partition, and zero otherwise). We note that this comparison criterion is popular in existing classification and/or clustering literature.

**Results.** Overall the RW DP-rich and RW SDP-rich have average NMI values that are comparable to those of MCMC and VI, and they are also higher than those of RW DP-means and RW SDP-means. This could be attributable to the presence of *rgr* regularization in the RW DP-rich and RW SDP-rich setups.

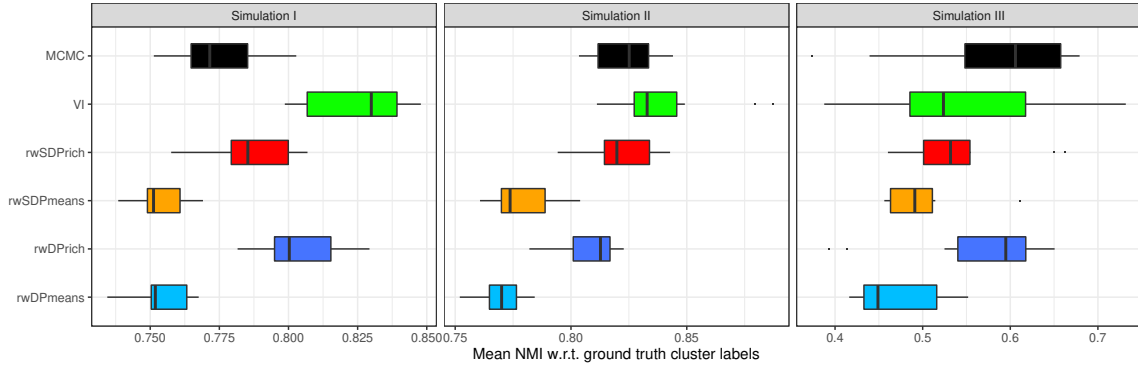


Figure 5.2: Sampling distribution of average NMI  $\eta_{(\cdot)}^{(b,t)}$  in comparison with ground-truth cluster assignments (Equation (5.19)) among  $T = 10$  simulated data sets in 3 simulation settings for each of the 6 methods: MCMC, VI and the 4 random-weighting setups.

### 5.5.2. Specifying priors for benchmark and motivating data examples

For both benchmark data sets, we specify a burn-in period of 2000 and a thinning interval of 15 for MCMC. Meanwhile, the priors for these data sets are specified with an Empirical Bayes approach (i.e., priors are estimated using some information from the data itself), and then the same set of priors are adopted for MCMC, VI and RW SDP-rich (where applicable) to facilitate meaningful comparison among all these methods.

#### Benchmark data examples

For iris data set, the full-covariance structure is adopted for MCMC, VI and RW SDP-rich. Based on the ground-truth cluster labels, we find their corresponding cluster-specific centroids and covariances. Let  $\tilde{\mu}_0$  and  $\tilde{\Sigma}_0$  be the corresponding weighted (by true mixing proportion) average of these centroids and covariances. Then, we specify  $\mu_0 = \tilde{\mu}_0$ ,  $\nu_0 = d + 3 = 7$ , and  $\psi_0 = 2 \times \tilde{\Sigma}_0$ , such that the inverse-Wishart prior mean of  $\Sigma$  is equal to  $\tilde{\Sigma}_0$  whereas the prior variance of  $\Sigma$  is huge.  $\xi_0$  is estimated to be the average of element-wise ratios between the diagonals of  $\tilde{\Sigma}_0$  and the diagonals of the covariance of all data. Finally,  $\alpha_0$  is fixed at 0.4 such that the CRP prior mean of  $\kappa$  (e.g., Teh, 2010) is equal to  $K_{true} = 3$ .

Similar method of prior specification is also used for the wine data set, except that we adopt the diagonal covariance structure here, since we are analyzing the transformed data

set via PCA. Again, let  $\tilde{\mu}_0$  and  $\{\tilde{\sigma}_{0,j}^2\}_{1 \leq j \leq d}$  be the corresponding weighted (by true mixing proportion) average of the cluster-specific centroids and variances for each dimension  $j = 1, \dots, d$ . Then, we fix  $\mu_0 = \tilde{\mu}_0$ ,  $a_{0,j} = 2$ , and  $b_{0,j} = 2 \times \tilde{\sigma}_{0,j}^2$  for all  $1 \leq j \leq d$ , such that the inverse-Gamma prior mean of  $\sigma_j^2$  is equal to  $\tilde{\sigma}_{0,j}^2$  whereas the prior variance of  $\sigma_j^2$  is huge. Similarly, for each dimension  $j$ ,  $\xi_{0,j}$  is taken to be ratio between  $\tilde{\sigma}_{0,j}^2$  and variance of  $\{y_{ij}\}_{1 \leq i \leq n}$ . Again, we also fix  $\alpha_0 = 0.4$  to equate the CRP prior mean of  $\kappa$  to  $K_{true} = 3$ .

### TCR Data Example

We first perform an agglomerative hierarchical clustering (HC) with average linkage on the 3-dimensional TCR data set ( $n = 13387$  TCR sequences) to obtain a solution path of partitions starting from singleton/atomic clusters to all observations lumped together in a degenerate cluster.

Next, we need to determine a suitable cutoff on the HC dendrogram which would give us a corresponding partition to help us specify our priors. Now, we could obtain Shannon's entropy for each partition along the hierarchical clustering (HC) solution path. Intuitively, a "good" partition is one that clusters homogeneous observations together and separates non-homogeneous ones apart, which leads to a significant drop in Shannon's entropy. Consequently, for each partition along the HC solution path, we repeatedly permute the cluster labels and recalculate the corresponding (permuted) entropy. By keeping track of the percentage of permuted entropies that are smaller (more extreme) than the observed entropies, we are able to obtain a series of permutation p-values associated with the HC solution path. This permutation exercise reveals that the partition consisting of 1477 clusters is the finest partition we could get before the permutation p-values rise up sharply, i.e. subsequent agglomeration of the clusters no longer leads to any significant drop in Shannon's entropy.

Based on these 1477 clusters obtained from hierarchical clustering, we specify our priors for the DPM under the diagonal-covariance structure: we fix  $\alpha_0 = 420$  so that the CRP prior mean of  $\kappa$  is approximately 1470, and that the VI stick-breaking threshold is fixed at

$K_{max} = 2000$ . Subsequently, similar to our preceding benchmark data analyses, based on these 1477 HC clusters, we compute the weighted (by the mixing proportion as indicated by the 1477 HC clusters) average  $\tilde{\mu}_0$  of the cluster centroids, weighted variance (denoted with  $\tilde{\mu}_0$ ) of the cluster centroids, weighted average  $\tilde{\sigma}_{0,j}^2$  of the cluster variances as well as the weighted variance (denoted with  $\check{\sigma}_{0,j}^2$ ) of cluster variances for  $j = 1, \dots, d$ . Finally, we specify the priors  $\mu_{0,j}$ ,  $\xi_{0,j}$ ,  $a_{0,j}$  and  $b_{0,j}$  via method-of-moments:

$$\begin{aligned}\mu_{0,j} &= \tilde{\mu}_{0,j}, \\ \xi_{0,j} &= \frac{\tilde{\sigma}_{0,j}^2}{\tilde{\mu}_{0,j}}, \\ a_{0,j} &= \frac{[\tilde{\sigma}_{0,j}^2]^2}{\check{\sigma}_{0,j}^2} + 2, \\ b_{0,j} &= \check{\sigma}_{0,j}^2 \times (a_{0,j} - 1).\end{aligned}$$

We also specify a burn-in period of 2000 and a thinning interval of 10 for MCMC.

### 5.5.3. Additional plots and tables

This subsection serves as a placeholder for additional plots and tables for the numerical experiments in this paper.

Computational times for the numerical experiments are tabulated in Tables 5.1 and 5.2 for comparison. Recall, from the main text in Chapter 4, that each random-weighting and VI algorithm is repeated 5 times to improve their respective local solutions. TCR data analysis is performed using UW Madison Biomedical Computing Group (BCG) computational hosts (URL: <https://bcg.biostat.wisc.edu/computational-hardware/>). The random-weighting schemes (as well as the sampling of  $\{z_{i(VI)}^{(b,t)}\}$  for  $b = 1, \dots, B$ ) are parallelized over 10 computing nodes. All other simulations and benchmark data analyses are performed using a laptop computer with Intel Core i7-8559U 2.7 GHz processor and 16GB RAM, which has 8 computing nodes for parallelization of the random-weighting schemes and the



sampling of  $\left\{z_{i(\text{VI})}^{(b,t)}\right\}_{1 \leq b \leq B}$ .

Methods	Simulation I	Simulation II	Simulation III
MCMC	81.0 s (100 %)	367.5 s (100 %)	300.0 s (100 %)
RW DP-means	9.3 s (11.5 %)	9.9 s (2.7 %)	14.7 s (4.9 %)
RW DP-rich	10.3 s (12.7 %)	11.0 s (3.0 %)	13.1 s (4.4 %)
RW SDP-means	23.1 s (28.5 %)	30.9 s (8.4 %)	53.5 s (17.8 %)
RW SDP-rich	20.5 s (25.3 %)	29.9 s (8.1 %)	42.5 s (14.2 %)
VI	3.6 s (4.4 %)	4.8 s (1.3 %)	6.5 s (2.1 %)

Table 5.1: Average (across  $T$  simulated data sets) computational times for various methods in our simulations. The proportion of average computational time (as a percentage of that of MCMC) for each method in each simulation setting is presented in parenthesis. Unit ‘s’ stands for seconds.

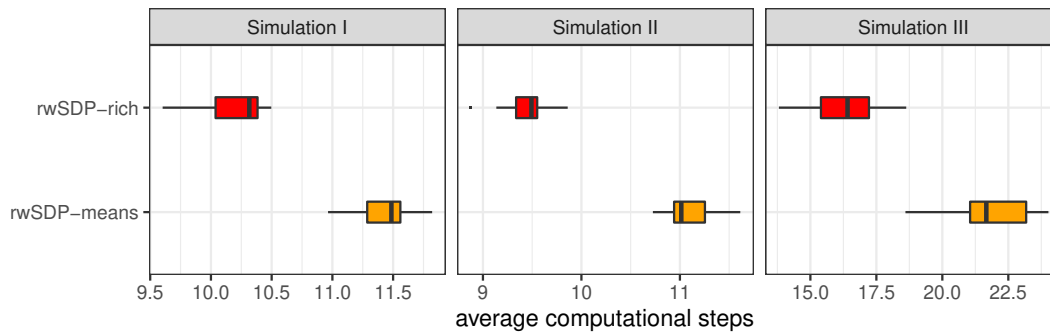


Figure 5.3: Sampling distribution of average (over  $B$  random-weighting draws) computational times for RW SDP-means and RW SDP-rich across  $T = 10$  simulated data sets in the 3 simulation settings.

The computational times for the random-weighting schemes tabulated in Tables 5.1 and 5.2 largely reflect the orders of complexity (see, Chapter 4.2) for these methods. We remind readers that our random-weighting schemes are trivially parallelizable over multiple

Methods	Iris Data	Wine Data (after PCA)	TCR Data
MCMC	25.0 s (100 %)	1.34 s (100 %)	10.17 d (100 %)
RW DP-means	0.24 s (1.0 %)	0.32 s (23.9 %)	7.93 h (3.2 %)
RW DP-rich	0.29 s (1.2 %)	0.38 s (28.3 %)	12.33 h (5.0 %)
RW SDP-means	0.70 s (2.8 %)	0.85 s (63.4 %)	14.25 h (5.8 %)
RW SDP-rich	0.73 s (2.9 %)	0.81 s (54.5 %)	15.54 h (6.4 %)
VI	0.11 s (0.4 %)	0.22 s (16.4 %)	6.78 h (2.8 %)

Table 5.2: Computational times for various methods in our benchmark and motivating data examples. The proportion of computational time (as a percentage of that of MCMC) for each method in each data set is presented in parenthesis. Units ‘s’, ‘h’ and ‘d’ represent seconds, hours and days respectively.

computing nodes, and we could further shorten their respective computational times by increasing the number of available computing resources. It is interesting to note that the average computational times for RW SDP-rich in the simulation settings are slightly shorter than those of RW SDP-means, considering the fact that the RW SDP-rich setup involves calculation of an additional logarithmic term in  $d_{ik}^w$ . This is due to fewer computational steps taken by RW SDP-rich to achieve local convergence, as illustrated in Figure 5.3.

Meanwhile, Figure 5.4 shows the trace plots for posterior number of clusters obtained by MCMC for all benchmark and motivating data sets. In particular, the trace plot corresponding to the MCMC scheme deployed on the original wine data set (original number of features  $d = 13$ ) using R package DPpackage shows poor mixing of the MCMC chain. This MCMC implementation involves a full covariance structure where the priors are specified using similar approach that we described in the preceding subsection. A closer inspection of the covariance among the features reveals a highly-correlated data structure, which justifies

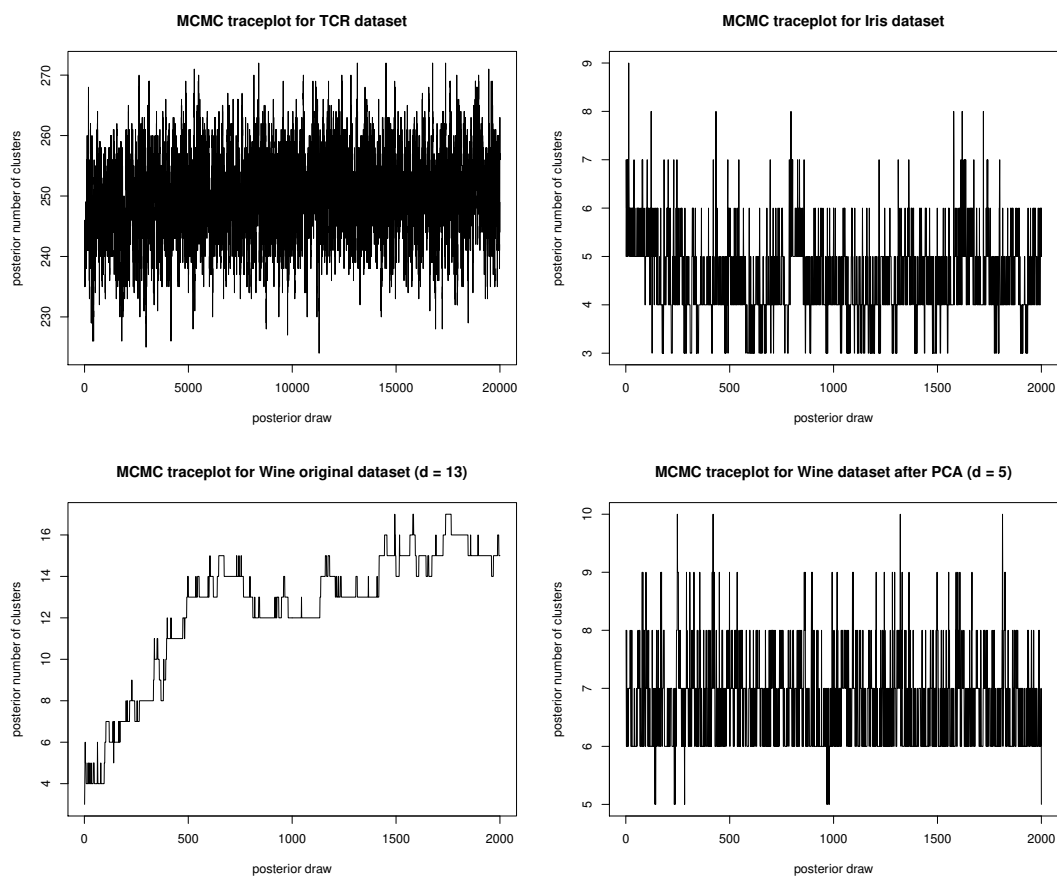


Figure 5.4: Trace plots for posterior number of clusters obtained by MCMC for all benchmark and motivating data sets.

our approach of first transforming the data set with PCA and use only the first 5 principal components which explain more than 80% of the variation in the data as illustrated in Figure 5.5. Subsequent MCMC implementation based on this transformed data set demonstrates reasonable mixing of the MCMC chain.

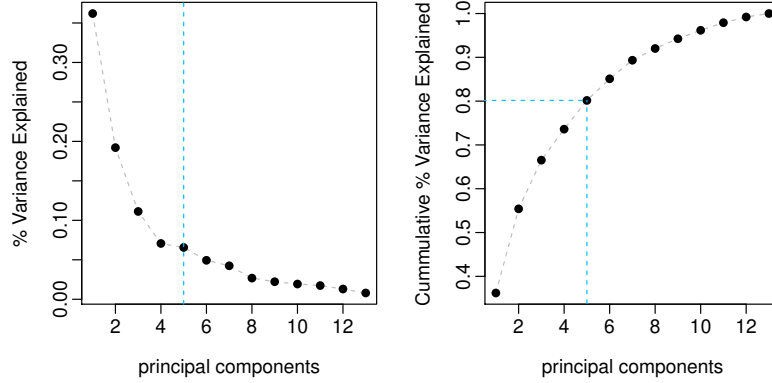


Figure 5.5: PCA scree plots for wine data set depicting percentage of variance explained in the data across the principal components. The blue dashed lines represent linear mapping of the original data set to a 5-dimensional subspace. The gray dashed lines only serve as interpolation between the points to ease visual inspection.

## 5.6. Variational Inference

The DPM of Normals (with Normal-inverse-Wishart prior) can be expressed with a stick-breaking prior (Sethuraman, 1994):

$$\begin{aligned}
 y_i | (z_i = k, \mu_k, \Sigma_k) &\sim N_d(\mu_k, \Sigma_k) \\
 \mu_k | \Sigma_k &\sim N_d(\mu_0, h(\Sigma_k)) \\
 \Sigma_k &\sim p(\Sigma_k) \\
 z_i | \pi(\mathbf{v}) &\sim \text{Mult}(\pi(\mathbf{v})) \\
 \pi(\mathbf{v}) | \alpha_0 &\sim \text{GEM}(\alpha_0) \iff \pi_k = v_k \prod_{l=1}^{k-1} (1 - v_l) \text{ for } v_k \sim \text{Beta}(1, \alpha_0).
 \end{aligned} \tag{5.20}$$

Note that in the main text, we have considered a DPM working model which shares a common mixture-component covariance term. However, since the MCMC schemes deployed in the numerical experiments adopt the more general form of DPM that allows cluster-specific covariance terms, we also adopt the same generalized DPM model for variational inference (VI) so that the results obtained by VI are more closely aligned to the MCMC samples.

### 5.6.1. Full-covariance structure

Under the full-covariance structure,  $h(\Sigma_k) = \Sigma_k/\xi_0$ , and here we specify a Wishart prior on the precision

$$\Sigma_k^{-1} \sim \text{Wishart}_d(\nu_0, \psi_0) \quad (5.21)$$

Note that the parameterization used for Wishart distribution is such that  $\mathbb{E}(\Sigma_k^{-1}) = \nu_0\psi_0$ . Applying the mean-field variational inference at a truncation level  $K_{max}$ , we approximate

$$p(\mu, \Sigma, z, \mathbf{v} | \mathbf{y})$$

with

$$\prod_{k=1}^{K_{max}} [q(\mu_k | \Sigma_k) q(\Sigma_k) q(v_k)] \times \prod_{i=1}^n q(z_i),$$

where the variational densities  $q$  are specified as below:

$$\begin{aligned} \mu_k | \Sigma_k &\sim N_d\left(\hat{\mu}_k, \frac{1}{\hat{\xi}_k} \Sigma_k\right) \\ \Sigma_k^{-1} &\sim \text{Wishart}_d(\hat{\nu}_k, \hat{\psi}_k) \\ z_i &\sim \text{Mult}(\hat{\pi}_i) \\ v_k &\sim \text{Beta}(\hat{\alpha}_{k1}, \hat{\alpha}_{k2}). \end{aligned} \quad (5.22)$$

We need to solve for

$$\{\hat{\alpha}_{k1}, \hat{\alpha}_{k2}\}_{1 \leq k \leq K_{max}} \quad \text{and} \quad \{\hat{\mu}_k, \hat{\xi}_k, \hat{\nu}_k, \hat{\psi}_k\}_{1 \leq k \leq K_{max}} \quad \text{and} \quad \{\hat{\pi}_{ik}\}_{1 \leq i \leq n, 1 \leq k \leq K_{max}}.$$

Using techniques outlined in Section 4.1 of Nakajima, Watanabe, and Sugiyama (2019), we obtain

$$\begin{aligned}
\hat{\mu}_k &= \frac{\xi_0 \mu_0 + \sum_{i=1}^n \hat{\pi}_{ik} y_i}{\xi_0 + \sum_{i=1}^n \hat{\pi}_{ik}} \\
\hat{\xi}_{kj} &= \xi_{0j} + \sum_{i=1}^n \hat{\pi}_{ik} \\
\hat{\nu}_k &= \nu_0 + \sum_{i=1}^n \hat{\pi}_{ik} \\
\hat{\psi}_k^{-1} &= \sum_{i=1}^n \hat{\pi}_{ik} y_i y_i' + \xi_0 \mu_0 \mu_0' - \hat{\xi}_k \hat{\mu}_k \hat{\mu}_k' + \psi_0^{-1} \\
\hat{\alpha}_{k1} &= 1 + \sum_{i=1}^n \hat{\pi}_{ik} \\
\hat{\alpha}_{k2} &= \alpha_0 + \sum_{l=k+1}^{K_{max}} \sum_{i=1}^n \hat{\pi}_{il},
\end{aligned} \tag{5.23}$$

and

$$\begin{aligned}
\bar{\pi}_{ik} = \exp \left\{ \right. & \left. [\Psi(\hat{\alpha}_{k1}) - \Psi(\hat{\alpha}_{k1} + \hat{\alpha}_{k2})] + \sum_{l=1}^{k-1} [\Psi(\hat{\alpha}_{k2}) - \Psi(\hat{\alpha}_{k1} + \hat{\alpha}_{k2})] \right. \\
& + \frac{1}{2} \left[ \sum_{j=1}^d \Psi \left( \frac{\hat{\nu}_k + 1 - j}{2} \right) + d \log 2 + \log |\hat{\psi}_k| \right. \\
& \left. \left. - \frac{d}{\hat{\xi}_k} - \hat{\nu}_k (y_i - \hat{\mu}_k)' \hat{\psi}_k (y_i - \hat{\mu}_k) \right] \right\}
\end{aligned} \tag{5.24}$$

where  $\Psi(\cdot)$  denotes the digamma function, such that

$$\hat{\pi}_{ik} = \frac{\bar{\pi}_{ik}}{\sum_{l=1}^{K_{max}} \bar{\pi}_{il}}. \tag{5.25}$$

The coordinate ascent algorithm is used to iteratively update the parameters of the variational distributions until the **evidence lower bound (ELBO)** converges:

$$\begin{aligned}
\mathcal{L}_q = & - \sum_{k=1}^{K_{max}} \sum_{i=1}^n \hat{\pi}_{ik} \log \hat{\pi}_{ik} - \sum_{k=1}^{K_{max}} \log \frac{\Gamma(\hat{\alpha}_{k1} + \hat{\alpha}_{k2})}{\Gamma(\hat{\alpha}_{k1}) \Gamma(\hat{\alpha}_{k2})} \\
& + \sum_{k=1}^{K_{max}} \sum_{j=1}^d \log \Gamma \left( \frac{\hat{\nu}_k + 1 - j}{2} \right) + \frac{1}{2} \sum_{k=1}^{K_{max}} \hat{\nu}_k \log |\hat{\psi}_k| \\
& + \frac{d \log 2}{2} \sum_{k=1}^{K_{max}} \hat{\nu}_k - \frac{d}{2} \sum_{k=1}^{K_{max}} \log (\hat{\xi}_k).
\end{aligned} \tag{5.26}$$

The predictive distribution  $p(y_{n+1} | \mathbf{y}_{1:n})$  is approximated with

$$\begin{aligned}
& \sum_{k=1}^{K_{max}} \left\{ \frac{\hat{\alpha}_{k1}}{\hat{\alpha}_{k1} + \hat{\alpha}_{k2}} \times \prod_{j=1}^{k-1} \frac{\hat{\alpha}_{j2}}{\hat{\alpha}_{j1} + \hat{\alpha}_{j2}} \right. \\
& \left. \times \pi^{-d/2} \times \left( \frac{\hat{\xi}_k}{1 + \hat{\xi}_k} \right)^{\frac{d}{2}} \times \frac{\Gamma \left( \frac{\hat{\nu}_k + 1}{2} \right)}{\Gamma \left( \frac{\hat{\nu}_k + 1 - d}{2} \right)} \times \frac{|\hat{\psi}_k^{-1}|^{\frac{\hat{\nu}_k}{2}}}{|\hat{\psi}_{new,k}^{-1}|^{\frac{\hat{\nu}_k + 1}{2}}} \right\},
\end{aligned} \tag{5.27}$$

where

$$\hat{\psi}_{new,k}^{-1} = \hat{\psi}_k^{-1} + \frac{\hat{\xi}_k}{1 + \hat{\xi}_k} (y_{n+1} - \hat{\mu}_k)(y_{n+1} - \hat{\mu}_k)'.$$

The pairwise probability of clustering the  $i^{th}$  and  $j^{th}$  observations together under the VI approach is given by

$$\check{p}_{ij}^{(t)}(\text{VI}) := \sum_{k=1}^{K_{max}} \hat{\pi}_{ik} \hat{\pi}_{jk}.$$

### 5.6.2. Diagonal-covariance structure

For diagonal-covariance structure,  $\Sigma_k = \text{diag}(\sigma_{k1}^2, \dots, \sigma_{kd}^2)$  in (5.20), and

$$h(\Sigma_k) = \text{diag} \left( \frac{\sigma_{k1}^2}{\xi_{0,1}}, \dots, \frac{\sigma_{kd}^2}{\xi_{0,d}} \right).$$

Gamma priors are adopted for the precision terms, i.e. for  $j = 1, \dots, d$ ,

$$\frac{1}{\sigma_{kj}^2} \stackrel{ind}{\sim} \text{Gamma}(a_{0j}, b_{0j}).$$

Applying the mean-field variational inference at a truncation level  $K_{max}$ , we approximate

$$p(\mu, \sigma^2, z, \mathbf{v} | \mathbf{y})$$

with

$$\prod_{k=1}^{K_{max}} \left\{ q(v_k) \prod_{j=1}^d [q(\mu_{kj} | \sigma_{kj}^2) q(\sigma_{kj}^2)] \right\} \times \prod_{i=1}^n q(z_i),$$

where the variational densities  $q(z_i)$  and  $q(v_k)$  are the same as their counterparts in (5.22), and the other variational densities for the component parameters are

$$\begin{aligned} \mu_{kj} | \sigma_{kj}^2 &\sim N\left(\hat{\mu}_{kj}, \frac{1}{\hat{\xi}_{kj}} \sigma_{kj}^2\right) \\ \frac{1}{\sigma_{kj}^2} &\sim \text{Gamma}(\hat{a}_{kj}, \hat{b}_{kj}). \end{aligned}$$

We need to solve for  $\{\hat{\alpha}_{k1}, \hat{\alpha}_{k2}\}_{1 \leq k \leq K_{max}}$  and

$$\left\{ \hat{\mu}_{kj}, \hat{\xi}_{kj}, \hat{a}_{kj}, \hat{b}_{kj} \right\}_{1 \leq k \leq K_{max}; 1 \leq j \leq d} \quad \text{and} \quad \left\{ \hat{\pi}_{ik} \right\}_{1 \leq k \leq K_{max}; 1 \leq i \leq n}.$$

The solutions for  $\{\hat{\alpha}_{k1}, \hat{\alpha}_{k2}, \hat{\mu}_{kj}, \hat{\xi}_{kj}\}$  are the same as their counterparts in (5.23), whereas

$$\begin{aligned} \hat{a}_{kj} &= a_{0j} + \frac{1}{2} \sum_{i=1}^n \hat{\pi}_{ik} \\ \hat{b}_{kj} &= b_{0j} + \frac{1}{2} \sum_{i=1}^n \hat{\pi}_{ik} y_{ij}^2 + \frac{1}{2} \xi_{0j} \mu_{0j}^2 - \frac{1}{2} \hat{\xi}_{kj} \hat{\mu}_{kj}^2. \end{aligned}$$



We still use (5.25) to calculate  $\hat{\pi}_{ik}$ , but we need to modify the formula to calculate  $\bar{\pi}_{ik}$  by replacing the second and third line of (5.24) inside the exponent with

$$+\frac{1}{2}\sum_{j=1}^d\left[\Psi(\hat{a}_{kj})-\log(\hat{b}_{kj})-\frac{1}{\hat{\xi}_{kj}}-\frac{\hat{a}_{kj}}{\hat{b}_{kj}}(y_{ij}-\hat{\mu}_{kj})^2\right].$$

The formula for ELBO also needs to be modified by replacing the second and third line of (5.26) with

$$+\sum_{k=1}^{K_{max}}\sum_{j=1}^d\log\Gamma(\hat{a}_{kj})-\sum_{k=1}^{K_{max}}\sum_{j=1}^d\hat{a}_{kj}\log(\hat{b}_{kj})-\frac{1}{2}\sum_{k=1}^{K_{max}}\sum_{j=1}^d\log(\hat{\xi}_{kj}).$$

Finally, the formula to approximate the predictive distribution should be modified by replacing the second line of (5.27) with

$$(2\pi)^{-d/2}\times\prod_{j=1}^d\left(\frac{\hat{\xi}_{kj}}{1+\hat{\xi}_{kj}}\right)^{\frac{1}{2}}\times\prod_{j=1}^d\frac{\Gamma(\hat{a}_{kj}+\frac{1}{2})}{\Gamma(\hat{a}_{kj})}\times\prod_{j=1}^d\frac{\hat{b}_{kj}^{\hat{a}_{kj}}}{\check{b}_{kj}^{\hat{a}_{kj}+\frac{1}{2}}},$$

where

$$\check{b}_{kj}=\hat{b}_{kj}+\frac{\hat{\xi}_{kj}}{1+\hat{\xi}_{kj}}\frac{(y_{n+1,j}-\hat{\mu}_{kj})^2}{2}.$$

## References

- Anderson, E. (1935). The irises of the Gaspé Peninsula. *Bulletin of the American Iris Society*, 59, 2–5.
- Arthur, D., & Vassilvitskii, S. (2007). k-means++: the advantages of careful seeding. In *Proceedings of the eighteenth annual acm-siam symposium on discrete algorithms* (pp. 1027–1035).
- Belloni, A., & Chernozhukov, V. (2013). Least squares after model selection in high-dimensional sparse models. *Bernoulli*, 19(2), 521–547.
- Bhadra, A., Datta, J., Polson, N. G., & Willard, B. T. (2019). LASSO meets horseshoe: a survey. *Statistical Science*, 34(3), 405–427.
- Bissiri, P. G., Holmes, C. C., & Walker, S. G. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 78(5), 1103–1130.
- Blackwell, D., & MacQueen, J. B. (1973). Ferguson distribution via Polya Urn schemes. *The Annals of Statistics*, 1(2), 353–355. doi: 10.1214/aos/1176342372
- Blei, D. M., & Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1), 121–144.
- Broderick, T., Kulis, B., & Jordan, M. I. (2013). MAD-Bayes: MAP-based asymptotic derivations from bayes. *Proceedings of the 30th International Conference on Machine Learning*, 28(3), 226–234.
- Camponovo, L. (2015). On the validity of the pairs bootstrap for LASSO estimators. *Biometrika*, 102(4), 981–987.

- Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., ... Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of statistical software*, 76(1).
- Castillo, I., Schmidt-Hieber, J., & van der Vaart, A. (2015). Bayesian linear regression with sparse priors. *The Annals of Statistics*, 43(5), 1986–2018.
- Chatterjee, A., & Lahiri, S. N. (2010). Asymptotic properties of the residual bootstrap for LASSO estimators. *Proceedings of the American Mathematical Society*, 138(12), 4497–4509.
- Chatterjee, A., & Lahiri, S. N. (2011a). Bootstrapping LASSO estimators. *Journal of the American Statistical Association*, 106(494), 608–625.
- Chatterjee, A., & Lahiri, S. N. (2011b). Strong consistency of LASSO estimators. *Sankhya: The Indian Journal of Statistics, Series A*, 73(1), 55–78.
- Chatterjee, S., & Bose, A. (2005). Generalized bootstrap for estimating equations. *The Annals of Statistics*, 33(1), 414–436.
- Corradin, R., Canale, A., & Nipoti, B. (2021). BNPmix: Bayesian nonparametric mixture models [Computer software manual]. Retrieved from <https://CRAN.R-project.org/package=BNPmix> (R package version 0.2.8)
- Dahl, D. B. (2009). Modal clustering in a class of product partition models. *Bayesian Analysis*, 4(2), 243–264. doi: 10.1214/09-BA409
- Das, D., Gregory, K., & Lahiri, S. N. (2019). Perturbation bootstrap in Adaptive Lasso. *The Annals of Statistics*, 47(4), 2080–2116.
- Das, D., & Lahiri, S. N. (2019). Distributional consistency of the LASSO by perturbation bootstrap. *Biometrika*, 106(4), 957–964.
- Dash, P., Fiore-Gartland, A. J., Hertz, T., Wang, G. C., Sharma, S., Souquette, A., ... Thomas, P. G. (2017). Quantifiable predictive features define epitope-specific T cell receptor repertoires. *Nature*, 547, 89–93.
- Duchi, J. C., Jordan, M. I., Wainwright, M. J., & Wibisono, A. (2015). Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5), 2788–2806.

- Durrett, R. (2010). *Probability: Theory and examples (cambridge series in statistical and probabilistic mathematics)* (4th ed.). New York, USA: Cambridge: Cambridge University Press.
- Eddelbuettel, D., & Sanderson, C. (2014). RcppArmadillo: Accelerating R with high-performance C++ linear algebra. *Computational Statistics and Data Analysis*, 71, 1054–1063. Retrieved from <http://dx.doi.org/10.1016/j.csda.2013.02.005>
- Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456), 1348-1360.
- Fang, Y., & Wang, J. (2012). Selection of the number of clusters via the bootstrap method. *Computational Statistics and Data Analysis*, 56, 468–477. doi: 10.1016/j.csda.2011.09.003
- Fong, E., Lyddon, S., & Holmes, C. (2019). Scalable nonparametric sampling from multimodal posteriors with the posterior bootstrap. *Proceedings of the 36th International Conference on Machine Learning*, 97, 1952–1962.
- Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 1–22. Retrieved from <http://www.jstatsoft.org/v33/i01/>
- Geyer, C. (1996). *On the asymptotics of convex stochastic optimization*. (Unpublished manuscript)
- Ghosal, S., Ghosh, J. K., & Ramamoorthi, R. V. (1999, 03). Posterior consistency of Dirichlet mixtures in density estimation. *The Annals of Statistics*, 27(1), 143–158. Retrieved from <https://doi.org/10.1214/aos/1018031105> doi: 10.1214/aos/1018031105
- Ghosal, S., Ghosh, J. K., & van der Vaart, A. W. (2000, 04). Convergence rates of posterior distributions. *The Annals of Statistics*, 28(2), 500–531. Retrieved from <https://doi.org/10.1214/aos/1016218228> doi: 10.1214/aos/1016218228
- Ghosal, S., & van der Vaart, A. (2017). *Fundamentals of nonparametric Bayesian inference*. Cambridge University Press.
- Gramacy, R. B., Moler, C., & Turlach, B. A. (2019). monomvn: Estimation for mvn and student-t data with monotone missingness [Computer software manual]. Retrieved

- from <https://CRAN.R-project.org/package=monomvn> (R package version 1.9-13)
- Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4), 711–732.
- Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1), 100–108.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference and prediction* (Second ed.). Springer.
- Henderson, H. V., & Searle, S. R. (1981). On deriving the inverse of a sum of matrices. *SIAM Review*, 23(1), 53–60.
- Hjort, N. L., & Ongaro, A. (2005). Exact inference for random Dirichlet means. *Statistical Inference for Stochastic Processes*, 8(3), 227–254.
- Huber, P. J. (1967). The behavior of maximum likelihood estimates under nonstandard conditions. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, volume 1: Statistics* (pp. 221–233). Berkeley, Calif.: University of California Press. Retrieved from <https://projecteuclid.org/euclid.bsm/1200512988>
- Ishwaran, H., & Zarepour, M. (2002). Exact and approximate sum representations for the Dirichlet process. *Canadian Journal of Statistics*, 30(2), 269–283.
- Jara, A., Hanson, T., Quintana, F., Mueller, P., & Rosner, G. (2011). Bayesian semi- and nonparametric modeling in R. *Journal of Statistical Software*, 40(5), 1–30. Retrieved from <http://www.jstatsoft.org/v40/i05/>
- Jensen, S. T., & Liu, J. S. (2008). Bayesian clustering of transcription factor binding motifs. *Journal of the American Statistical Association*, 103(481), 188–200. doi: 10.1198/016214507000000365
- Jin, Z., Ying, Z., & Wei, L.-J. (2001). A simple resampling method by perturbing the minimand. *Biometrika*, 88(2), 381–390.
- Johnson, V., & Rossell, D. (2012). Bayesian model selection in high-dimensional settings. *Journal of the American Statistical Association*, 107(498), 649–660.
- Jordan, M. I. (2013). On statistics, computation and scalability. *Bernoulli*, 19(4), 1378–1390.

- Karabatsos, G. (2020). Fast search and estimation of Bayesian nonparametric mixture models using a classification annealing EM algorithm. *Journal of Computational and Graphical Statistics*, 1–12. doi: 10.1080/10618600.2020.1807995
- Kleijn, B., & van der Vaart, A. (2012). The bernstein-von-mises theorem under misspecification. *Electronic Journal of Statistics*, 6, 354–381. Retrieved from <https://doi.org/10.1214/12-EJS675> doi: 10.1214/12-EJS675
- Knight, K., & Fu, W. (2000). Asymptotics for LASSO-type estimators. *The Annals of Statistics*, 28(5), 1356-1378.
- Kulis, B., & Jordan, M. I. (2012). Revisiting k-means: New algorithms via Bayesian nonparametrics. *Proceedings of the 29th International Conference on Machine Learning*.
- Lai, T. L., Robbins, H., & Wei, C. Z. (1978). Strong consistency of least squares estimates in multiple regression. *Proceedings of National Academy of Sciences*, 75(7), 3034 - 3036.
- Leonardi, G. P., & Tamanini, I. (2002). Metric spaces of partitions, and Caccioppoli partitions. *Advances in Mathematical Sciences and Applications*, 12(2), 725–753.
- Liu, H., & Yu, B. (2013). Asymptotic properties of LASSO+mLS and LASSO+Ridge in sparse high-dimensional linear regression. *Electronic Journal of Statistics*, 7, 3124-3169.
- Lyddon, S., Holmes, C., & Walker, S. (2019). General Bayesian updating and the loss-likelihood bootstrap. *Biometrika*, 106(2), 465–478.
- Lyddon, S., Walker, S., & Holmes, C. (2018). Nonparametric learning from Bayesian models with randomized objective functions. In *Proceedings of the 32nd international conference on neural information processing systems* (pp. 2075–2085). Curran Associates Inc. Retrieved from <http://dl.acm.org/citation.cfm?id=3326943.3327135>
- Ma, X., Korthauer, K., Kendzierski, C., & Newton, M. A. (2021). A compositional model to assess expression changes from single-cell RNA-seq data. *The Annals of Applied Statistics*, 15(2), 880–901.
- Mason, D. M., & Newton, M. A. (1992). A rank statistics approach to the consistency of a general bootstrap. *The Annals of Statistics*, 20(3), 1611–1624.
- McLachlan, G. J., Lee, S. X., & Rathnayake, S. I. (2019). Finite mixture models. *Annual review*

*of statistics and its application*, 6, 355–378.

- Minnier, J., Tian, L., & Cai, T. (2011). A perturbation method for inference on regularized regression estimates. *Journal of the American Statistical Association*, 106(496), 1371–1382.
- Mossel, E., & Vigoda, E. (2006). Limitations of Markov Chain Monte Carlo algorithms for Bayesian inference of phylogeny. *The Annals of Applied Probability*, 16(4), 2215–2234.
- Muliere, P., & Secchi, P. (1996). Bayesian nonparametric predictive inference and bootstrap techniques. *Annals of the Institute of Statistical Mathematics*, 48(4), 663–673.
- Müller, P., Quintana, F. A., Jara, A., & Hanson, T. (2015). *Bayesian nonparametric data analysis*. Springer.
- Nakajima, S., Watanabe, K., & Sugiyama, M. (2019). *Variational Bayesian learning theory* (First ed.). Cambridge University Press.
- Narisetty, N. N., & He, X. (2014). Bayesian variable selection with shrinking and diffusing priors. *The Annals of Statistics*, 42(2), 789–817.
- Nemeth, C., & Fearnhead, P. (2021). Stochastic gradient Markov Chain Monte Carlo. *Journal of the American Statistical Association*, 116(533), 433–450. doi: 10.1080/01621459.2020.1847120
- Newton, M., Polson, N. G., & Xu, J. (2021). Weighted Bayesian bootstrap for scalable posterior distributions. *The Canadian Journal of Statistics*, 49(2), 421–437. Retrieved from <https://doi.org/10.1002/cjs.11570>
- Newton, M. A., & Raftery, A. (1994). Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 56(1), 3–48.
- Ng, T. L., & Newton, M. A. (2020). Random weighting in LASSO regression. *arXiv: 2002.02629*. (In revision at the Electronic Journal of Statistics.)
- Osborne, M. R., Presnell, B., & Turlach, B. A. (2000). On the LASSO and its dual. *Journal of Computational and Graphical Statistics*, 9(2), 319–337.
- Park, T., & Casella, G. (2008). The Bayesian LASSO. *Journal of the American Statistical Association*, 103(482), 681–686.

- Paul, D., & Das, S. (2020). A Bayesian non-parametric approach for automatic clustering with feature weighting. *Stat*, 9(1). doi: 10.1002/sta4.306
- Pitman, J. (1995). Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, 102(2), 145–158.
- Pollard, D. (1981). Strong consistency of k-means clustering. *The Annals of Statistics*, 9(1), 135-140.
- Pollard, D. (1991). Asymptotics for least absolute deviation regression estimators. *Econometric Theory*, 7(2), 186-199.
- R Core Team. (2019). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from <https://www.R-project.org/>
- Raykov, Y. P., Boukouvalas, A., & Little, M. A. (2016). Simple approximate MAP inference for Dirichlet processes mixtures. *Electronic Journal of Statistics*, 10, 3548–3578. doi: 10.1214/16-EJS1196
- Richter, R., & Alexa, M. (2015). Mahalanobis centroidal Voronoi tessellations. *Computers and Graphics*, 46, 48–54. doi: 10.1016/j.cag.2014.09.009
- Roeder, K. (1990). Density estimation with confidence sets exemplified by superclusters and voids in the galaxies. *Journal of the American Statistical Association*, 85, 617–624.
- Rubin, D. B. (1981). The Bayesian Bootstrap. *The Annals of Statistics*, 9(1), 130–134.
- Scrucca, L., Fop, M., Murphy, T. B., & Raftery, A. E. (2016). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1), 289–317. doi: 10.32614/RJ-2016-021
- Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 4, 639-650.
- Shao, J. (2003). *Mathematical Statistics* (Second ed.). New York, USA: Springer Texts in Statistics.
- Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 62(4), 795-809.
- Teh, Y. W. (2010). Dirichlet process [Computer software manual]. (<https://www.stats>



.ox.ac.uk/~teh/research/npbayes/Teh2010a.pdf)

- Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 58(1), 267-288.
- Tibshirani, R. J. (2013). The LASSO problem and uniqueness. *Electronic Journal of Statistics*, 7, 1456-1490.
- Tibshirani, R. J., & Taylor, J. (2011). The solution path of the generalized LASSO. *The Annals of Statistics*, 39(3), 1335–1371.
- Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3), 475–494.
- Urschel, J. C. (2017). On the characterization and uniqueness of centroidal Voronoi tessellations. *SIAM Journal on Numerical Analysis*, 55(3), 1525–1547. doi: 10.1137/15M1049166
- van der Vaart, A. W. (1998). *Asymptotic statistics (cambridge series in statistical and probabilistic mathematics)*. Cambridge University Press.
- Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with s* (Fourth ed.). New York: Springer. Retrieved from <http://www.stats.ox.ac.uk/pub/MASS4> (ISBN 0-387-95457-0)
- Vinh, N. X., Epps, J., & Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11, 2837–2854.
- Vujovic, M., Degn, K. F., Marin, F. I., Schaap-Johansen, A.-L., Chain, B., Andresen, T. L., ... Marcatili, P. (2020). T cell receptor sequence clustering and antigen specificity. *Computational and Structural Biotechnology Journal*, 18, 2166–2173. doi: 10.1016/j.csbj.2020.06.041
- Wade, S., & Ghahramani, Z. (2018). Bayesian cluster analysis: point estimation and credible balls (with discussion). *Bayesian Analysis*, 13(2), 559–626. doi: 10.1214/17-BA1073
- Welling, M., & Teh, Y. W. (2011). Bayesian Learning via Stochastic Gradient Langevin Dynamics. *Proceedings of International Conference on Machine Learning*, 681–688.
- Williams, G. J. (2011). *Data mining with Rattle and R: The art of excavating data for*

*knowledge discovery*. Springer. Retrieved from [http://www.amazon.com/gp/product/1441998896/ref=as\\_li\\_qf\\_sp\\_asin\\_tl?ie=UTF8&tag=togaware-20&linkCode=as2&camp=217145&creative=399373&creativeASIN=1441998896](http://www.amazon.com/gp/product/1441998896/ref=as_li_qf_sp_asin_tl?ie=UTF8&tag=togaware-20&linkCode=as2&camp=217145&creative=399373&creativeASIN=1441998896)

Zahm, C., Ng, T. L., Newton, M. A., & McNeel, D. (2022). Antigen specificity of T-cell receptors. *In preparation*.

Zhao, P., & Yu, B. (2006). On model selection consistency of LASSO. *Journal of Machine Learning Research*, 7, 2541-2563.

Zou, H. (2006). The adaptive LASSO and its oracle properties. *Journal of the American Statistical Association*, 101(476), 1418–1429.

Zuanetti, D. A., Muller, P., Zhu, Y., Yang, S., & Ji, Y. (2019). Bayesian nonparametric clustering for large data sets. *Statistics and Computing*, 29, 203–215. doi: 10.1007/s11222-018-9803-9