Bayesian Analysis (2022) **TBA**, Number TBA, pp. 1

Random Weighting in Discrete Mixture Models

Tun Lee Ng[*] and Michael A. Newton[†,*]

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 of discrete mixture models, with emphasis on surrogate posterior samples for latent clustering parameters. We examine several objective functions revealed as limiting forms from a Dirichlet Process Mixture (DPM) working model. Unlike techniques reliant on small-variance-asymptotics of the DPM, the proposed DP-rich setup retains the rich-gets-richer property of the DPM. We further apply the random-weighting mechanism under the Bayesian nonparametric learning (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 and confirm local convergence of solutions. We 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 the RW SDP-rich approach provides reasonable approximation to MCMC posterior clustering for the DPM model. Finally, we establish several large-sample asymptotic properties of random-weighting in the Bayesian NPL framework. Additional details for our random-weighting mixture models are collected in the supplementary material.

Keywords: Dirichlet process mixture model, Random Weighting, Weighted bootstrap, Bayesian bootstrap, DP-rich, DP-means, Finite mixture model, Bayesian non-parametric learning (NPL).

1 Introduction

Motivation. 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 and Ghahramani, 2018, and references therein), or it might be relevant when clustering is one element in sequence of data-analysis steps (e.g., Ma et al., 2021). The straightforward Bayesian approach to address clustering inference 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., Scrucca et al., 2016; Müller et al., 2015). When modeling elements reflect dominant features of the applied problem, and when diagnostic calculations confirm suitability of assumed relationships, no other data-analysis strategy is more well supported. However, one often seeks inference summaries that are not overly sensitive to modeling assumptions; a theme of some recent research asks how working models may guide inference summaries without fully specifying a generative

[*] Department of Statistics, University of Wisconsin-Madison, WI 53706. tng25@wisc.edu

[†] Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison, WI 53706. newton@stat.wisc.edu

model and prior. Generalized Bayesian inference (Bissiri et al., 2016) or Bayesian nonparametric learning (NPL) (Lyddon et al., 2018, 2019) rely on models only to guide optimization-based computations. Benefits may be computational (e.g., leveraging optimization tools; no MCMC diagnosis) as well as statistical (e.g., less reliance on model assumptions). Recognizing that discrete mixture models have long provided a model-based approach to clustering (e.g., McLachlan et al., 2019), we investigate new and potentially useful Bayesian NPL schemes for clustering inference.

Related work.. **Put DP means here** 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 several settings, especially given the problem to assure Monte Carlo error bounds with MCMC (e.g. Mossel and Vigoda, 2006) and the increased size of data sets (e.g., Welling and Teh, 2011). While many approximate Bayesian procedures are available for finite-mixture-models (e.g., Nemeth and 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., Zuanetti et al., 2019; Karabatsos, 2020) instead of uncertainty quantification. Ongoing challenges with existing techniques warrant further development of approximate posterior inference for discrete mixture models.

continue to refine ***Our contribution.*** 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 model of the paper: 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. We explore various related random-weighting mixture models via simplifications of our RW SDP-rich setup. Subsequently, in Section 4, 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, in Section 5, 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 in the supplementary material.

2 Preliminaries

2.1 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 $\{y_1, y_2, \cdots, y_n\}$ in a subset of d-dimensional Euclidean space: $\Omega \subseteq \mathbb{R}^d$. Our calculations presume the existence of a distribution F_* on Ω 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 Ω the parameter

$$\theta := \arg\min_{t \in \Theta} \mathscr{L}(t, F) := \arg\min_{t \in \Theta} \int_\Omega \tilde{l}(t, y)\, dF(y). \tag{2.1}$$

The choice of $\tilde{l}(t, y)$ establishes θ 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. Or, setting $\tilde{l}(t, y)$ to be the negative loglikelihood corresponding to some *working parametric model*, say $F_\theta \in \mathcal{F}_\Theta$, leads to $\theta_* := \arg\min_{t \in \Theta} \mathscr{L}(t, F_*)$; this minimizes the Kullback-Leibler divergence **citation**, and yields a well-defined parameter without having assumed that the working parametric model has captured the data-generating distribution F_*. Part of the present contribution is to generate novel loss functions $\tilde{l}(t, y)$ by considering *working nonparametric models*, all in the context of the clustering problem.

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 specializes (2.1),

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

The use of loss-functions to guide inference has been exceedingly effective; for instance, estimation is enabled by plugging the empirical distribution F_n into (2.1) and leveraging sophisticated optimization tools to solve the minimization problem (e.g., Hastie et al., 2009). Our interest is clustering, which may be aligned with the present framework through, for example, the expected loss:

$$\mathscr{L}(A_K, F) = \int_\Omega \min_{a \in A_K} \|y - a\|_2^2\, dF(y), \tag{2.2}$$

where $A_K = \{a_1, \cdots, a_K\}$ contains K distinct points on $\mathbb{R}^d$. While the $K-$means algorithm aims to minimize the empirical expected loss $\mathscr{L}(A_K, F_n)$ (Hartigan and Wong,

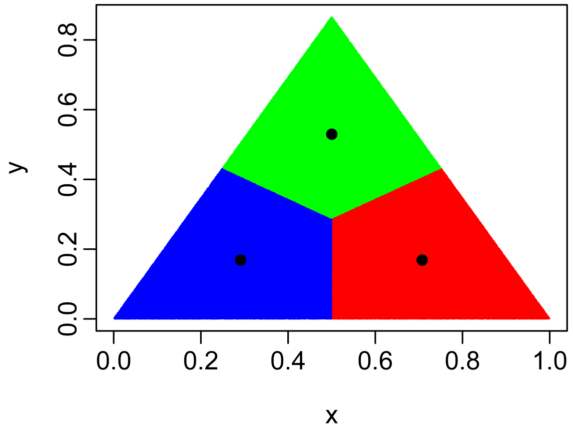


Figure 1: K-means clustering of a population that lacks intrinsic clustering, but rather is uniformly distributed on an equilateral triangle with vertices $\{(0,0),(1,0),(0.5,\sqrt{3}/2)\}$. The black dots represent the centroids obtained by minimizing 2.2 with $K = 3$, and colors distinguish the induced population-level clusters.

1979; Pollard, 1981), a functional parameter exists even when the population F_* is not induced by any underlying clustering mechanism. Figure 1 illustrates the point for a uniform distribution F_* on a triangular support in two dimensions. A contribution of the present paper is to explore elaborations of the K-means loss in (2.2) – elaborations that are guided by structured nonparametric models and that operate effectively in posterior clustering computations. In particular, the elaborations respond features that are not recognized in (2.2), including the number, size distribution, and scaling of clusters.

2.2 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 $\{y_1, y_2, \cdots, y_n\}$; 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 $\boldsymbol{W} = (W_1, W_2, \cdots, W_n)$, Then the expected loss $\mathscr{L}(t, F)$ associated with such a posterior-sampled F is proportional to

$$\mathscr{L}_\lambda(t, \boldsymbol{W}) := \sum_{i=1}^{n} W_i l(t, y_i) + \lambda l_0(t). \tag{2.3}$$

There are various ways to handle the regularization weight λ; for ~~simplicity~~ both theoretical and empirical reasons, here we ignore posterior variation this penalty, but other approaches have merit (e.g., Ng and Newton, 2022). ** I think we'll want a bit more justification of this treatment, to avoid reviewer concerns...maybe it's something we can pick up in a discussion section; I guess the main point is that the natural randomness term has variance on the order of $1/n$, right?** We refer readers to the supplementary material for a detailed derivation of our random-weighting model, in which the regularization term has a factor of $\frac{1}{n}\sum_{i=1}^{n} W_i$ converging to one almost surely, which renders a theoretical justification for setting unitary weights on the regularization term. In addition, we refer readers to Section 3 for our empirical findings about the calibration of the tuning parameter λ_0. Assignment of random weights on the regularization term may adversely affect our model performance; see Section 4 and supplementary material for more details.

The random-weighting/Bayesian-bootstrap approach amounts to repeated assignment of random weights $\boldsymbol{W} = (W_1, W_2, \cdots, W_n)$ and minimization in t of (2.3) to obtain a sample of the functional parameter values, θ. Utility of the approach depends in part on the suitability of loss functions $l(t, y_i)$ and $l_0(t)$. **it's been with us since Rubin and other early cites, and it persists thanks in part to advances in optimization** 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 new loss functions identified from parameter-limiting calculations within a class of nonparametric models.

2.3 Working model

In the search for working-model-based loss functions $\tilde{l}(t, y)$, we consider the Dirichlet Process Mixture (DPM) model linking Gaussian observations to a Chinese Restaurant Process (CRP) mixture specification (Blackwell and MacQueen, 1973): **maybe we should also provide a contemporary citation e.g. book by Hjort et al?**

$$\begin{gathered} y_i|(z_i = k, \mu_k, \Sigma) \sim N_d(\mu_k, \Sigma) \\ \mu_k \Big| (\Sigma, \boldsymbol{z}, \kappa) \sim N_d(\mu_0, h(\Sigma)) \\ \Sigma \sim p(\Sigma) \\ (\boldsymbol{z}, \kappa) \sim CRP(\alpha_0), \end{gathered} \tag{2.4}$$

where $\alpha_0 > 0$ and $\mu_0 \in \mathbb{R}^d$. As an example, $p(\Sigma)$ could be an inverse-Wishart density with ν_0 degrees of freedom and a symmetric positive-definite scale matrix ψ_0, whereas $h(\Sigma) = {}^{\Sigma}/_{\xi_0}$ for some $\xi_0 > 0$. Our working model has a common covariance structure

Σ across all mixture components (unless deliberately stated otherwise). Note that the number of clusters is denoted with κ in (2.4) 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 κ and cluster assignments $\boldsymbol{z} = \{z_1, \cdots, z_n\}$ characterize the partitioning of a DPM. In an extension of the Bayesian NPL setting, working model (2.4) itself is induced by nonparametric rather than parametric considerations. Elaborating (2.4), the working joint density is given by

$$\begin{aligned} & p\left(\boldsymbol{Y}, \boldsymbol{z}, \kappa, \{\mu_k\}_{k=1}^{\kappa}, \Sigma\right) \\ & := p\left(\boldsymbol{Y} \middle| \boldsymbol{z}, \kappa, \{\mu_k\}_{k=1}^{\kappa}, \Sigma\right) \times p\left(\{\mu_k\}_{k=1}^{\kappa} \middle| \Sigma, \boldsymbol{z}, \kappa\right) \times p\left(\Sigma\right) \times p(\boldsymbol{z}, \kappa) \\ & = (2\pi)^{-\frac{nd}{2}} |\Sigma|^{-\frac{n}{2}} \exp\left\{ -\frac{1}{2} \sum_{k=1}^{\kappa} \sum_{i:z_i=k} (y_i - \mu_k)' \Sigma^{-1} (y_i - \mu_k) \right\} \\ & \times (2\pi)^{-\frac{d\kappa}{2}} \mid h(\Sigma)|^{-\frac{\kappa}{2}} \exp\left\{ -\frac{1}{2} \sum_{k=1}^{\kappa} (\mu_k - \mu_0)' \left[h(\Sigma)\right]^{-1} (\mu_k - \mu_0) \right\} \\ & \times p\left(\Sigma\right) \times \alpha_0^{\kappa-1} \frac{\Gamma(\alpha_0+1)}{\Gamma(\alpha_0+n)} \prod_{k=1}^{\kappa} \Gamma(n_k), \end{aligned} \tag{2.5}$$

which serves as a guidepost in specifying the loss functions $l(t, y_i)$ and $l_0(t)$ in (2.3).

Kulis and Jordan (2012) and Broderick et al. (2013) approached (2.4) from the perspective of small-variance-asymptotics, identifying the DP-means objective function:

$$\mathscr{L}_{\lambda}^{\text{DPmeans}}(\boldsymbol{\mu}, \boldsymbol{z}, \kappa) := \sum_{k=1}^{\kappa} \sum_{i:z_i=k} \|y_i - \mu_k\|_2^2 + \lambda\kappa. \tag{2.6}$$

One goal of that work was to show that maximum a posteriori (MAP) estimation could be achieved approximately by K-means-like optimization. Broderick et al. (2013) constructed (2.6) with the small-variance asymptotics outlined in Remark 2.1. A diminishing σ^2 (indicating a decreasing variance of mixture components and more clusters) faces an opposing force of diminishing α_0 (i.e., lower intensity of the CRP to create new clusters), such that a balance is achieved via the regularization parameter λ.

Remark 2.1 (DP-means as small-variance limit of the DPM (Broderick et al., 2013))**.** *Consider the DPM model (2.4), where $\Sigma = \sigma^2 I_d$ and $h(\Sigma) = c^2 I_d$ for some finite c. If $\sigma^2 \to 0$ and $\alpha_0 \to 0$ such that they are modulated with $\alpha_0 = \exp\left\{-\lambda/(2\sigma^2)\right\}$ for some $\lambda > 0$, then the negative log of (2.5), multiplied by σ^2, converges to the DP-means objective function (2.6).*

Our initial interest was to incorporate random weights as in (2.3) directly into the DP-means objective (2.6), thinking that where DP-means calculations transform the MAP-calculation problem into a K-means-like problem, then randomly-weighted DP-means might transform the entire DP-posterior into randomized K-means problems. Roughly this program works, as we confirm with large-sample asymptotic calculations in Section 5. Finite-sample performance demands a somewhat more nuanced approach, especially as it pertains to the extraction of relevant loss functions that retain the computational simplicity of (2.6) while providing more useful cluster-inference summaries. Notably, the DP-means loss function does not recognize the empirical distribution of cluster sizes, and it is also not sensitive to the scaling of measurements assignable to a given cluster. These limitations are addressed next.

3 Methodology

3.1 DP-rich: Alternate asymptotics for the DPM

While the small-variance limit in (2.6) 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 $\sigma^2 \to \infty$ 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 σ^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 (2.4), where $\sigma^2 \sim \lambda_2$ for some tuning parameter $\lambda_2 > 0$ to be calibrated by the analyst.

In addition, notice that α_0 is the CRP intensity parameter while ξ_0 acts as the scaling factor between the variance of mixture components and the prior variance of μ_k. Contrary to the SVA setup in Remark 2.1, we further argue that increasing $\alpha_0 \to \infty$ and reducing $\xi_0 \to 0$ must go hand-in-hand from an Empirical-Bayes perspective: if the variance of mixture components stays rather "constant" (i.e., $\sigma^2 \sim \lambda_2$), then larger number of clusters signifies wider data coverage in the Euclidean space. In this case, new centroids must have arisen farther away from μ_0 in order to establish these new "colonies" or clusters. Hence, $\alpha_0 \to \infty$ (indicating higher intensity to create new clusters under the CRP prior) and $\xi_0 \to 0$ (suggesting a noisier prior for μ_k) must happen concurrently. Finally, these limiting behaviors of α_0 and ξ_0 are modulated together with λ_2 via the relationship

$$\lambda_1 = \lambda_2 \log\left[\left(\frac{2\pi\lambda_2}{\xi_0}\right)^{\frac{d}{2}} \frac{1}{\alpha_0}\right], \tag{3.1}$$

where $\lambda_1 > 0$ is another tuning parameter to be calibrated by the analyst (note that this modulating relationship between λ_1 and λ_2 in (3.1) holds with the limiting behavior of ξ_0 and α_0; in practice, we only require the regularization parameters $\lambda_1, \lambda_2 > 0$). These considerations lead to our first main result in Theorem 3.1 – new asymptotics for the DPM which we coin as the **DP-rich** objective function

$$\mathscr{L}^{\text{DP-rich}}_{(\lambda_1,\lambda_2)}(\boldsymbol{\mu}, \boldsymbol{z}, \kappa) := \sum_{k=1}^{\kappa} \sum_{i:z_i=k} \|y_i - \mu_k\|_2^2 + \lambda_1 \cdot \kappa - \lambda_2 \sum_{k=1}^{\kappa} \log\left[\Gamma(n_k)\right]. \tag{3.2}$$

Theorem 3.1 (**DP-rich as alternative asymptotics for the DPM**). *Consider the DPM model (2.4), where $\Sigma = \sigma^2 I_d$ and $h(\Sigma) = \frac{\sigma^2}{\xi_0} I_d$. If $\sigma^2 \sim \lambda_2$ for some $\lambda_2 > 0$, $\alpha_0 \to \infty$ and $\xi_0 \to 0$ such that they are modulated via (3.1) for some $\lambda_1 > 0$, then the negative log of (2.5), multiplied by σ^2, converges to the* ***DP-rich*** *objective function (3.2).*

Proof of Theorem 3.1*.* Given $\Sigma = \sigma^2 I_d$ and $h(\Sigma) = \frac{\sigma^2}{\xi_0} I_d$ and $\sigma^2 \sim \lambda_2$ for some $\lambda_2 > 0$, we have

$$\begin{aligned} -\sigma^2 \log p(\boldsymbol{Y}, \boldsymbol{z}, \kappa, \boldsymbol{\mu}) \sim \frac{1}{2} & \left[\sum_{k=1}^{\kappa} \sum_{i:z_i=k} \|y_i - \mu_k\|_2^2 + \xi_0 \sum_{k=1}^{\kappa} \|\mu_k - \mu_0\|_2^2 \right] \\ & + \kappa \cdot \lambda_2 \cdot \log \left[\left(\frac{2\pi\lambda_2}{\xi_0} \right)^{d/2} \cdot \frac{1}{\alpha_0} \right] - \lambda_2 \sum_{k=1}^{\kappa} \log \left[\Gamma(n_k) \right] \\ & + \frac{nd}{2} \lambda_2 \log(2\pi\lambda_2) - \lambda_2 \log \left[\frac{\Gamma(\alpha_0 + 1)}{\alpha_0 \Gamma(\alpha_0 + n)} \right]. \end{aligned} \tag{3.3}$$

Notice that we have treated Σ as deterministic in this case and so we dropped the term $p(\Sigma)$ in (2.5). Next, the third line of (3.3) does not contain $(\boldsymbol{\mu}, \kappa, \boldsymbol{z})$ and could be dropped. Finally, push $\alpha_0 \to \infty$ and $\xi_0 \to 0$ such that (3.1) is satisfied, and scale the entire equation by 2 to arrive at (3.2). In particular, we also verify that for any finite $n \geq 1$, as $\alpha_0 \to \infty$, $\dfrac{\Gamma(\alpha_0 + 1)}{\alpha_0 \Gamma(\alpha_0 + n)} = \dfrac{\Gamma(\alpha_0)}{\Gamma(\alpha_0 + n)} \to 1$. □

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 2.1. Clearly, setting $\lambda_2 = 0$ in (3.2), i.e. switching off the *rgr* regularization in the DP-rich setup, returns the DP-means objective function.

In addition, notice that, from (3.2), λ_1 allows direct calibration by the analyst to tune the number of clusters κ obtained by the DP-rich algorithm, whereas λ_2 controls the magnitude of the algorithm's *rgr* effect brought about by the term $\log[\Gamma(n_k)]$.

3.2 DP-rich: optimization and illustration

Similar to the DP-means algorithm, the objective function in (3.2) 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 $k \in \{1, \cdots, \kappa\}$ or allocated into a new cluster $\mathcal{C}_{\kappa+1}$, by comparing its "cost" of joining an existing cluster $\mathcal{C}_k$

$$d_{ik} = \|y_i - \mu_k\|_2^2 - \lambda_2 \log(n_{k,-i}) \tag{3.4}$$

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

$$d_{i,\kappa+1} = \lambda_1. \tag{3.5}$$

The term $n_{k,-i}$ in (3.4) 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 $k \in \{1, \cdots, \kappa\} \setminus \{k'\}$. From (3.4), 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 μ_k, which is discounted by $\log(n_{k,-i})$ with a factor of λ_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 1 DP-rich

```
Require: data $\{y_1, \cdots, y_n\}$, 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 $z_i^{\text{old}} = z_i$ do
 3:     $z_i^{\text{old}} \leftarrow z_i$ for all $i$.
 4:     for each data point $y_i$ do
 5:         Compute $d_{ik}$ with (3.4) for $k = 1, \cdots, \kappa$.
 6:         If $\min_{1 \le k \le \kappa} d_{ik} > \lambda_1$, set $\kappa = \kappa + 1$, $z_i = \kappa$ and $\mu_\kappa = y_i$. Otherwise, set $z_i = \arg\min_{1 \le k \le \kappa} d_{ik}$.
 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 $\{\mu_k\}_{1 \le k \le \kappa}$, and cluster assignments $\{z_i\}_{1 \le i \le n}$.
```

After re-assigning all the observations, we move on to the centroid updates. Conditional on the existing partition $(\kappa, \boldsymbol{z})$, the centroid μ_k is updated as the average of $\{y_i : i \in \mathcal{C}_k\}$ for $k = 1, \cdots, \kappa$. Algorithm 1 outlines this DP-rich procedure in detail, while Lemma 3.1 ensures local convergence of Algorithm 1. We refer readers to the supplementary material for other implementation details of the algorithm.

While the K-means procedure is influenced by the choices of initial cluster centroids (Arthur and Vassilvitskii, 2007), the DP-rich procedure depends on the order in which data points are processed (i.e., the order in which $y_i : i \in \{1, \cdots, n\}$ is processed in the for-loop (lines 4–8) of Algorithm 1). This feature is also shared by the DP-means algorithm (Kulis and Jordan, 2012), because both DP-rich and DP-means algorithms involve inserting new cluster(s) and/or deleting empty cluster(s) during their cluster

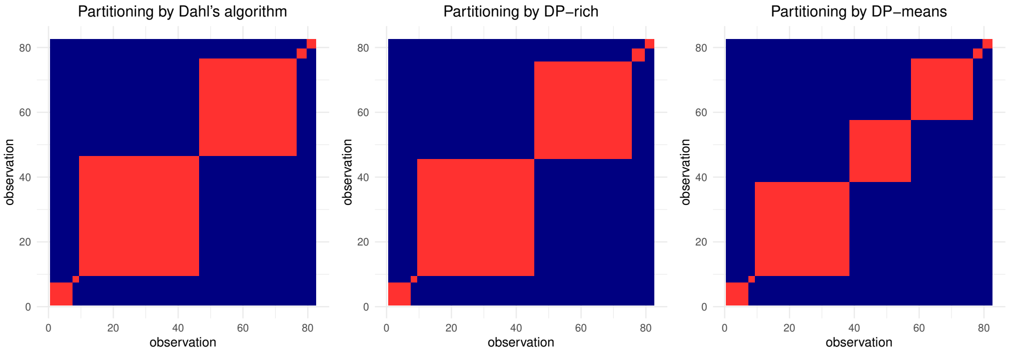


Figure 2: Cluster partitions obtained by Dahl (2009)'s algorithm, DP-rich and DP-means approaches for the 1-dimensional `Galaxy` data set (Roeder, 1990), where red color indicates that the pair of observations is clustered together and navy-blue color otherwise. The observations in the data set are arranged in ascending order.

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 3.1 (**Local Convergence of DP-rich**). *Algorithm 1 monotonically decreases the DP-rich objective function (3.2) until local convergence is achieved.*

Proof of Lemma 3.1. 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 $\lambda_2 \log(n_{k,-i})$. □

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 and 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 $K_{\text{true}} = 6$). 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 (2.4) with known mixture-component variance σ_y^2 and centroids' prior variance $h(\sigma^2) = \sigma_\mu^2$. Specifically, we specify the priors μ_0, σ_y^2 and σ_μ^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 κ is approximately 6. For DP-rich, we specify λ_2 to be the estimated σ_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 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 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 et al., 2010) for DP-rich is 0.916, whereas the NMI for DP-means in this case is 0.700. **scaling and RW...still puzzling if we should split this differently??**

3.3 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 (3.2), we introduce the objective function $\mathscr{L}_\lambda(t, \boldsymbol{W})$ in (2.3) for our main random-weighting countable-mixture model , which we coin as the **random-weighting scaled DP-rich (RW SDP-rich)** approach:

$$\begin{aligned}
&\mathscr{L}^{\text{rwSDP-rich}}_{(\lambda_1,\lambda_2)}(\boldsymbol{z}, \kappa, \boldsymbol{\mu}, \Sigma)\\
:=&\frac{1}{2}\left[\sum_{k=1}^{\kappa}\sum_{i:z_i=k} W_i(y_i-\mu_k)'\Sigma^{-1}(y_i-\mu_k) + \xi_0\sum_{k=1}^{\kappa}(\mu_k-\mu_0)'\Sigma^{-1}(\mu_k-\mu_0) + \operatorname{Tr}\left(\psi_0\Sigma^{-1}\right)\right]\\
&+\left(\sum_{i=1}^{n} W_i + \nu_0 - d - 1\right)\log\left|\Sigma^{1/2}\right| + \lambda_1\cdot\kappa - \lambda_2\sum_{k=1}^{\kappa}\log\left[\Gamma(n_k)\right], \qquad (3.6)
\end{aligned}$$

where the prior components $\mu_0 \in \mathbb{R}^d$ and $\xi_0 > 0$, $\nu_0 > d+1$ and the symmetric positive definite matrix ψ_0 are specified in (2.4), $W_i \overset{iid}{\sim} Exp(1)$, 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 $(\kappa, \boldsymbol{z})$ 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 (3.6). Besides retaining the tuning parameters λ_1 and λ_2 (that allow direct calibration of κ and the magnitude of the *rgr* effect respectively), we incorporate a common covariance term Σ 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 (3.6) is obtained by *modifying* (2.5); see supplementary material for more details about the modification.

3.4 RW SDP-rich: optimization

We repeatedly assign i.i.d. standard Exponential weights $\{W_i\}_{1\leq i\leq n}$ and optimize (3.6) for B times to obtain B random-weighting samples. For any given set of the i.i.d. $(W_1, \cdots, W_n)$, the objective function in (3.6) can be optimized using an algorithm that is similar to Algorithm 1. In particular, the "cost" of the i^{th} data point joining an existing cluster $\mathcal{C}_k$ is updated as

$$d_{ik}^w = \frac{1}{2} W_i (y_i - \mu_k^w)' \Sigma_w^{-1} (y_i - \mu_k^w) - \lambda_2 \log(n_{k,-i}) \tag{3.7}$$

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

$$d_{i,\kappa+1}^w = \frac{1}{2} \frac{\xi_0 W_i}{\xi_0 + W_i} (y_i - \mu_0)' \Sigma_w^{-1} (y_i - \mu_0) + \lambda_1. \tag{3.8}$$

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

$$\mu_k^w = \frac{\sum_{i:z_i=k} W_i y_i + \xi_0 \mu_0}{\sum_{i:z_i=k} W_i + \xi_0} \tag{3.9}$$

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

$$\Sigma_w = \frac{\sum_{k=1}^{\kappa} \sum_{i:z_i=k} W_i (y_i - \mu_k)(y_i - \mu_k)' + \xi_0 \sum_{k=1}^{\kappa} (\mu_k - \mu_0)(\mu_k - \mu_0)' + \psi_0}{(\sum_{i=1}^{n} W_i + \nu_0) - d - 1}. \tag{3.10}$$

These parameter updates (3.9) and (3.10) enable incorporation of prior information in (2.4), which will be superseded by data information as sample size increases.

Algorithm 2 outlines this random-weighting procedure in detail. Notice that this algorithm is trivially parallelizable over $b \in \{1, \cdots, B\}$, which enhances its scalability to large datasets. We refer readers to the supplementary material for other implementation details of the algorithm.

Lemma 3.2. ***(Local Convergence of RW SDP-rich)*** *For any given sets of positive weights $(W_1, \cdots, W_n)$, the while-loop (lines 4–16) of Algorithm 2 monotonically decreases the objective given in (3.6) until local convergence.*

Lemma 3.2 ensures local convergence of the RW SDP-rich algorithm. Its proof is given in the supplementary material. 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 $y_i : i \in \{1, \cdots, n\}$ is processed in the for-loop (lines 6–10) of Algorithm 2). Again, we suggest that for each set of random weights $(W_1, \cdots, W_n)$, we repeat the while-loop (lines 4–16) of Algorithm 2 several times in which we process the data points with different permuted order, and pick the set of solutions with the smallest objective.

3.5 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 3 summarizes these variations of the random-weighting procedures.

Algorithm 2 Random-weighting Scaled DP-rich (RW SDP-rich)

```
Require: data $\{y_1, \cdots, y_n\}$, regularization parameters $\lambda_1$ and $\lambda_2$, prior terms $\{\boldsymbol{\mu}_0, \xi_0, \nu_0, \psi_0\}$, and number of posterior draws $B$
for $b = 1, \cdots, B$ do
    Draw $W_i \overset{iid}{\sim} Exp(1)\ \forall\ i = 1, \cdots, n$.
    Initialize by assigning all observations into a single cluster. In addition, initialize $\Sigma_b^w = \psi_0 / (\nu_0 - d - 1)$.
    while true do
        $z_{i,b}^{w,\text{old}} \leftarrow z_{i,b}^w$ for all $i$.
        for each data point $y_i$ do
            Compute $d_{ik}^w$ with (3.7) for $k = 1, \cdots, \kappa_b^w$, and compute $d_{i,\kappa_b^w+1}^w$ with (3.8).
            If $\min_{1 \le k \le \kappa_b^w} d_{ik}^w > d_{i,\kappa_b^w+1}^w$, set $\kappa_b^w = \kappa_b^w + 1$, $z_{i,b}^w = \kappa_b^w$ and initialize $\mu_{\kappa_b^w,b}^w$ with (3.9). Otherwise, set $z_{i,b}^w = \arg\min_{1 \le k \le \kappa_b^w} d_{ik}^w$.
            Drop empty clusters if they exist.
        end for
        For each cluster $k$, update its cluster centroid $\mu_{k,b}^w$ with (3.9).
        Update $\Sigma_b^w$ with (3.10).
        if $z_{i,b}^{w,\text{old}} = z_{i,b}^w$ for all $i$ then
            Store $\kappa_b^w$, $\Sigma_b^w$, $\mu_{k,b}^w$ for $k = 1, \cdots, \kappa_b^w$ and $z_{i,b}^w$ for $i = 1, \cdots, n$.
        end if
    end while
end for
Ensure: $B$ samples of number of clusters $\{\kappa_b^w\}_{1 \le b \le B}$, covariance term $\{\Sigma_b^w\}_{1 \le b \le B}$, cluster centroids $\left\{\mu_{k,b}^w\right\}_{1 \le k \le \kappa_b^w; 1 \le b \le B}$, and cluster assignments $\left\{z_{i,b}^w\right\}_{1 \le i \le n; 1 \le b \le B}$.
```

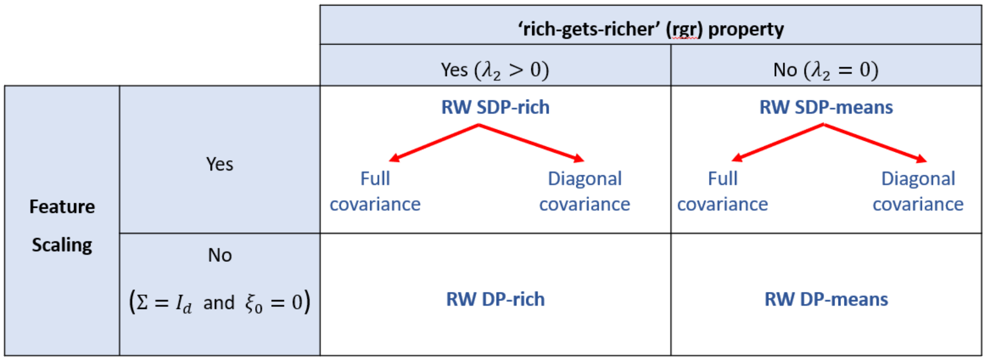
Diagonal Covariance Structure

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 $\{y_1, \cdots, y_n\}$, 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:

$$\begin{aligned}
&\mathscr{L}_{(\lambda_1,\lambda_2)}^{\text{rwSDP-rich}}(\boldsymbol{z}, \kappa, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \\
&:= \sum_{j=1}^{d} \frac{1}{2\sigma_j^2} \left[\sum_{k=1}^{\kappa} \sum_{i: z_i = k} W_i (y_{ij} - \mu_{kj})^2 + \xi_{0j} \sum_{k=1}^{\kappa} (\mu_{kj} - \mu_{0j})^2 + 2b_{0j} \right]
\end{aligned} \tag{3.11}$$

$$+\frac{1}{2}\sum_{j=1}^{d}\left(\sum_{i=1}^{n}W_i+2a_{0j}-2\right)\log\left(\sigma_j^2\right)+\lambda_1\cdot\kappa-\lambda_2\sum_{k=1}^{\kappa}\log\left[\Gamma(n_k)\right],$$

where $W_i \overset{iid}{\sim} Exp(1)$ and $b_{0,j}, \xi_{0,j} > 0$ and $a_{0,j} > 1$ for all $j = 1, \cdots, d$. In fact, this objective function (3.11) is derived by *modifying* (2.5) where a common *diagonal* covariance structure is adopted, i.e. $\Sigma = diag\left(\sigma_1^2, \cdots, \sigma_d^2\right)$, $h(\Sigma) = diag\left(\frac{\sigma_1^2}{\xi_{0,1}}, \cdots, \frac{\sigma_d^2}{\xi_{0,d}}\right)$, and $\sigma_j^2 \sim IG(a_{0,j}, b_{0,j})$ for $j = 1, \cdots, d$. Similar procedure (Algorithm 2) could be used to optimize (3.11), with slightly different formulae for parameter updates and costs of contribution to the objective. We refer readers to the supplementary material for these formulae. Notice that this algorithm runs faster than its full-covariance counterpart since matrix inversion of cluster covariance Σ is avoided in this case.



		'rich-gets-richer' (rgr) property	
		Yes ($\lambda_2 > 0$)	No ($\lambda_2 = 0$)
Feature Scaling	Yes	RW SDP-rich Full covariance Diagonal covariance	RW SDP-means Full covariance Diagonal covariance
	No ($\Sigma = I_d$ and $\xi_0 = 0$)	RW DP-rich	RW DP-means

Figure 3: Schematic depicting different variations of the random-weighting models **maybe it's better to say different variations of loss functions??**.

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 and Liu, 2008), then the *rgr* penalty in the RW SDP-rich could be discarded by setting $\lambda_2 = 0$ in (3.6) or (3.11). 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 (2.4). In addition, the analyst may also choose to specify $h(\Sigma) = {}^{I_d}/_{\xi_0}$, where $\xi_0 \to 0$ signifying a very noisy or uninformative prior for μ_k. In this case, the objective functions

(3.6) and (3.11) could be simplified into the **random-weighting DP-rich (RW DP-rich)** procedure

$$\mathscr{L}^{\text{rwDP-rich}}_{(\lambda_1,\lambda_2)}(\boldsymbol{z},\kappa,\boldsymbol{\mu}) := \sum_{k=1}^{\kappa}\sum_{i:z_i=k} W_i\|y_i-\mu_k\|_2^2 + \lambda_1\cdot\kappa - \lambda_2\sum_{k=1}^{\kappa}\log\left[\Gamma(n_k)\right]. \tag{3.12}$$

(3.12) is indeed the random-weighting version of (3.2). Again, Algorithm 2 could be used to optimize (3.12), 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. Σ is required here for (3.12) , thus leading to faster computation as compared to the RW SDP-rich setup. Note that setting $\lambda_2 = 0$ in (3.12) reduces the objective function to the **random-weighting DP-means (RW DP-means)**:

$$\mathscr{L}^{\text{rwDPmeans}}_{\lambda_1}(\boldsymbol{z},\kappa,\boldsymbol{\mu}) := \sum_{k=1}^{\kappa}\sum_{i:z_i=k} W_i\|y_i-\mu_k\|_2^2 + \lambda_1\cdot\kappa. \tag{3.13}$$

Again, (3.13) is the random-weighting version of (2.6).

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

$$\mathscr{L}^{\text{rwKmeans}}_{K}(\boldsymbol{\mu},\boldsymbol{z}) := \sum_{k=1}^{K}\sum_{i:z_i=k} W_i\left\|y_i-\mu_k\right\|_2^2. \tag{3.14}$$

We refer readers to the supplementary material 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.

3.6 RW SDP-rich: computational complexity

For every set of random weights $(W_1,\cdots,W_n)$, 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 $\mathcal{O}\left(B\cdot t_{\max}\cdot K\cdot n\cdot d\right)$, 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 $\mathcal{O}\left(B\cdot t_{\max}\cdot\bar{\kappa}_{\text{rwDP-rich}}\cdot n\cdot d\right)$, 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 (3.6), the order of complexity is given by

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

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 Σ_w.

RW SDP-rich (diagonal covariance structure). Since the RW Scaled DP-rich approach with diagonal covariance structure (3.11) does not involve calculation of Mahalanobis distance or inversion of covariance matrix, its computational complexity is reduced to $\mathcal{O}\left(B \cdot t_{\max} \cdot \bar{\kappa}_{\text{rwSDP-rich (diag)}} \cdot n \cdot d\right)$, where $\bar{\kappa}_{\text{rwSDP-rich (diag)}}$ denotes the average estimated number of clusters by the algorithm.

3.7 RW SDP-rich: calibrating regularization parameters

We first focus on the *rich-gets-richer (rgr)* tuning parameter λ_2 for both RW SDP-rich and RW DP-rich approaches. Based on the construction of DP-rich in Section 2 as well as Equation (3.7), 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

$$\begin{aligned} d_{ik}^{\text{rwSDP-rich}} &= \frac{1}{2}\left[W_i(y_i - \mu_k^w)'\Sigma_w^{-1}(y_i - \mu_k^w) - \log(n_{k,-i})\right] \\ d_{ik}^{\text{rwDP-rich}} &= W_i \left\|y_i - \mu_k^w\right\|_2^2 - \hat{\sigma}^2 \log(n_{k,-i}). \end{aligned} \tag{3.15}$$

Notice that if $\Sigma_w = I_d$ and $\hat{\sigma}^2 = 1$ in (3.15), then the weighted squared Mahalanobis distance or weighted squared Euclidean distance of a data point y_i from a centroid μ_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 to compare the performance of these two approaches using different *rgr* tuning parameters.

From (3.15), 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 Σ_w.

After determining λ_2, we now turn our attention to the tuning parameter λ_1 that directly regulates κ for all the random-weighting approaches. We opine that calibration of λ_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 λ_1 such that the average of $\{\kappa_b^w\}_{1 \leq b \leq B}$ mimics the MAP estimate of κ. 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 λ_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 for a detailed algorithm that outlines the specific steps for calibrating λ_1 after the analyst has decided on the benchmark measurement to be used for tuning this regularization parameter.

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/ngtunlee/random-weighting-mixture`. In particular, 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).

- Standard MCMC procedure for the DPM of Normals with full covariance structure (i.e., $h(\Sigma_k) = {}^{\Sigma_k}/_{\xi_0}$ and $\Sigma_k|(\boldsymbol{z},\kappa) \sim IW(\nu_0,\psi_0)$ in (2.4)) is implemented with R package `DPpackage` (Jara et al., 2011), and is compared with its variational inference counterpart (formulae provided in the supplementary material) as well as RW SDP-rich in (3.6).
- Standard MCMC procedure for the DPM of Normals with diagonal covariance structure, (i.e., $\Sigma_k = diag\left(\sigma^2_{k,1},\cdots,\sigma^2_{k,d}\right)$, $h(\Sigma_k) = diag\left(\frac{\sigma^2_{k,1}}{\xi_{0,1}},\cdots,\frac{\sigma^2_{k,d}}{\xi_{0,d}}\right)$, and $\sigma^2_{k,j}\Big|(\boldsymbol{z},\kappa) \sim IG(a_{0,j},b_{0,j})$ for $j = 1,\cdots,d$ in (2.4)), is implemented with R package `BNPmix` (Corradin et al., 2021), and is compared with its variational inference counterpart (see supplementary material) as well as RW SDP-rich in (3.11). Again, note that ξ_0 is a $d-$dimensional vector here, whereas ξ_0 under the full covariance structure is a scalar.

Notice that, even though our DPM working model in (2.4) considers a common covariance term acrosss the mixture components, the aforementioned existing software packages implement standard MCMC schemes that involve a more general form of DPM that allows cluster-specific caovriance 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 λ_1 for all random-weighting methods to mimic the posterior mean of κ 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 for more details). Computational times for all these methods in all of our numerical experiments are provided in the supplementary material.

4.1 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 $K_{\text{true}} = 16$ and $\Sigma_{\text{true}} = I_2$. 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 κ under the CRP is approximately 16. Corresponding to the diagonals of Σ_{true}, we specify $\lambda_2^{\text{rwDP-rich}} = 1$.

Simulation Setting II. The simulation setting is similar to Simulation Setting I, except that now $\Sigma_{\text{true}} = \left(\begin{smallmatrix} 1 & .5 \\ .5 & 1 \end{smallmatrix}\right)$, 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 $\psi_0 = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$, such that the inverse Wishart prior has a mean that equals Σ_{true}. Again, corresponding to the diagonals of Σ_{true}, we specify $\lambda_2^{\text{rwDP-rich}} = 1$.

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 $\Sigma_{\text{true}} = \left(\begin{smallmatrix} 1 & .5 \\ .5 & 1 \end{smallmatrix}\right)$, 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 Σ_{true}, we specify $\lambda_2^{\text{rwDP-rich}} = 1$.

For each simulated data set, we draw $B = 5000$ posterior (or random-weighting) 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

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

and

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

which represent the sampled/bootstrapped κ'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 for more detailed formulae.) In particular, the CRP prior of (2.4) could be reformulated as a stick-breaking prior (Sethuraman, 1994): for $i = 1, \cdots, n$,

$$z_i|\pi(\boldsymbol{v}) \sim Mult(1; \pi(\boldsymbol{v})) \quad \text{where} \quad \pi(\boldsymbol{v})|\alpha_0 \sim GEM(\alpha_0). \tag{4.1}$$

The VI method approximates the multinomial components in (4.1) with variational multinomial probabilities $\{\hat{\pi}_{i,k}\}_{1\le i\le n, 1\le k\le K_{\max}}$. 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

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

during the b^{th} iteration (draw), and obtain $\kappa^{(b,t)}_{(\text{VI})}$ as the number of unique values of $\left\{z^{(b,t)}_{i(\text{VI})}\right\}_{1\le i\le n}$.

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 $\boldsymbol{z}^{(b,t)}_{(\cdot)}$, which tells us about the number of clusters $\kappa^{(b,t)}_{(\cdot)}$ obtained by the method, as well as the cluster sizes $\left\{n^{(b,t)}_{k,(\cdot)}\right\}_{1\le k\le \kappa^{(b,t)}_{(\cdot)}}$.
   We keep track of the coefficient of variation (CoV) of these cluster sizes

$$\phi^{(b,t)}_{(\cdot)} := \text{CoV of } \left\{n^{(b,t)}_{k,(\cdot)}\right\}_{1\le k\le \kappa^{(b,t)}_{(\cdot)}}$$

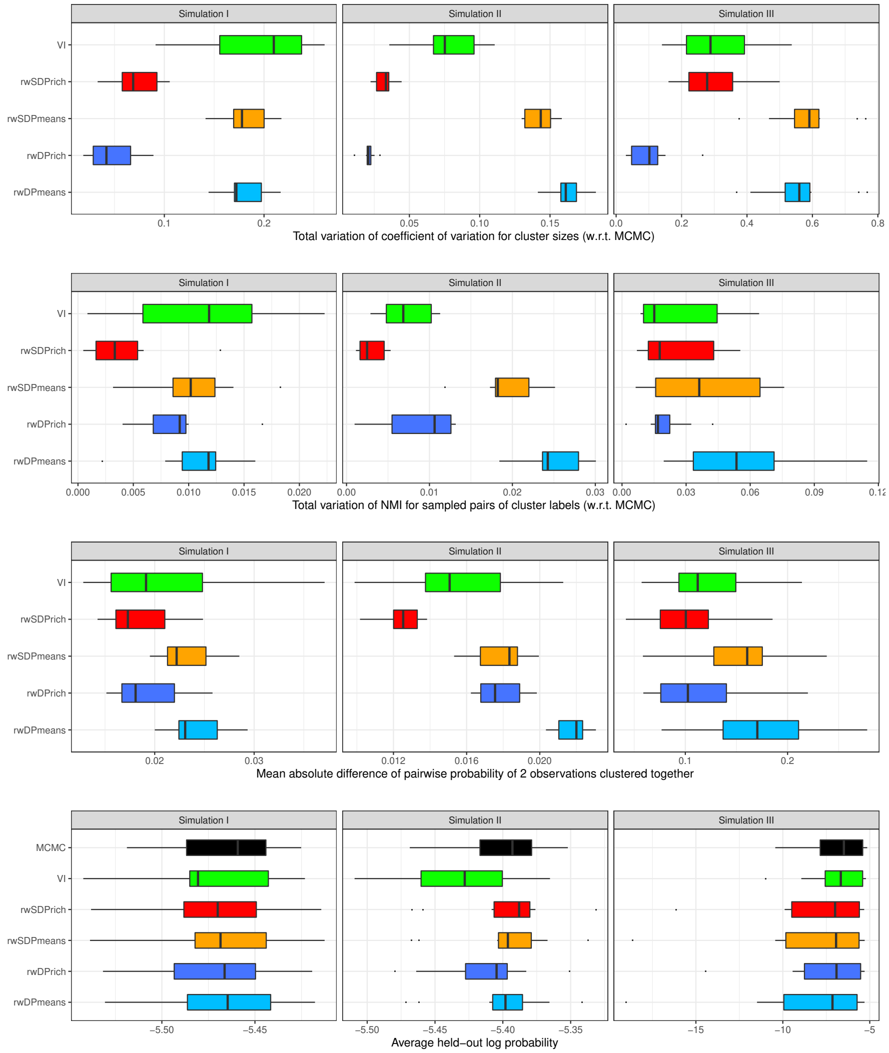


Figure 4: Sampling distribution for 4 comparison measurements among $T = 10$ simulated data sets in 3 simulation settings: total variation distance $TV_{\phi(\cdot)}^{(t)}$ (in comparison with MCMC) of CoV of cluster sizes (Criterion (1)), the total variation distance (in comparison with MCMC) of NMI for randomly-sampled pairs cluster assignments $TV_{\acute{\eta}(\cdot)}^{(t)}$ (Criterion (4)), mean absolute difference (in comparison with MCMC) of pairwise probabilities of clustering any two observations together $\check{p}_{(\cdot)}^{(t)}$ (Criterion (3)), and average held-out log probability $\tilde{p}_{(\cdot)}^{(t)}$ for all 6 methods including MCMC (Criterion (2)).

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

$$\hat{F}^{(t)}_{\phi(\cdot)} = \text{ecdf of } \left\{\phi^{(b,t)}_{(\cdot)}\right\}_{1\le b\le B}. \tag{4.2}$$

$\hat{F}^{(t)}_{\phi(\text{MCMC})}$ is treated as the "benchmark curve", and is used to compare with $\hat{F}^{(t)}_{\phi(\cdot)}$ from the other 5 methods, by keeping track of the total variation $TV^{(t)}_{\phi(\cdot)}$ between $\hat{F}^{(t)}_{\phi(\text{MCMC})}$ and $\hat{F}^{(t)}_{\phi(\cdot)}$.

2. **Average held-out log-probability**
Denote $\tilde{y}^{(t)}_{\tilde{i}}$ as the held-out (test) data for $\tilde{i} = 1, \cdots, m$, 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 held-out log-probability as follows:

$$\tilde{p}^{(t)}_{(\cdot)} := \frac{1}{m}\sum_{\tilde{i}=1}^{m} \log\left[\frac{1}{B}\sum_{b=1}^{B} \phi\left(\tilde{y}^{(t)}_{\tilde{i}} \middle| \mu^{(b,t)}_{(\cdot)}, \Sigma^{(b,t)}_{(\cdot)}\right)\right], \tag{4.3}$$

where $\phi(\cdot)$ denotes multivariate normal density, $\mu^{(b,t)}_{(\cdot)}$ and $\Sigma^{(b,t)}_{(\cdot)}$ denote the b^{th} posterior (or random-weighting) samples of μ and Σ obtained for the t^{th} simulated training dataset. Notice that RW DP-means and RW DP-rich algorithms do not produce samples of Σ (recall Equations (3.13) and (3.12)). In this case, the "pseudo" samples of Σ for RW DP-means and RW DP-rich are computed using (3.10) (or its diagonal-covariance-structure counterpart as shown in the supplementary material, depending on which covariance structure is adopted for MCMC, VI and RW SDP-rich), by substituting cluster centroids and cluster assignments that are obtained from RW DP-means or RW DP-rich respectively. Meanwhile, the average held-out log-probability for VI is computed based on its variational densities and its corresponding variational parameters. Detailed formulae are provided in the supplementary material.

3. **Pairwise probability of any two observations clustered together**
We keep track of the probability of clustering the i^{th} and j^{th} observations together by each of the 6 methods in the t^{th} simulated training dataset

$$\breve{p}^{(t)}_{ij(\cdot)} := \frac{1}{B}\sum_{b=1}^{B} \mathbb{1}_{\left\{z^{(b,t)}_{i(\cdot)} = z^{(b,t)}_{j(\cdot)}\right\}}$$

for all $i, j \in \{1, \cdots, n\}$ and $i \neq j$, and then compare the other 5 methods against MCMC by computing the average absolute difference of these pairwise probabilities for the t^{th} dataset

$$\breve{p}^{(t)}_{(\cdot)} := \frac{2}{n(n-1)}\sum_{i<j} \left|\breve{p}^{(t)}_{ij(\cdot)} - \breve{p}^{(t)}_{ij(\text{MCMC})}\right|. \tag{4.4}$$

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, $\acute{B}$ pairs of cluster assignments and compute their NMI's. Specifically, let $\left\{ \acute{\boldsymbol{z}}_{(\cdot)}^{(\acute{b},t)} \right\}_{1 \leq \acute{b} \leq \acute{B}}$ be the random samples from the (standard or approximate) posterior cluster assignments $\left\{ \boldsymbol{z}_{(\cdot)}^{(b,t)} \right\}_{1 \leq b \leq B}$ obtained by one of the 6 aforementioned methods for the t^{th} simulated training dataset. Next, we compute NMI for the $\acute{b}^{th}$ randomly-sampled pair of cluster assignments (where one of them is from MCMC) with
$$\acute{\eta}_{(\cdot)}^{(\acute{b},t)} := \mathrm{NMI}\left(\acute{\boldsymbol{z}}_{(\cdot)}^{(\acute{b},t)} \ , \ \acute{\boldsymbol{z}}_{(\mathrm{MCMC})}^{(\acute{b},t)} \right),$$
and then obtain the empirical distribution function (ecdf) of these NMI values
$$\hat{F}_{\acute{\eta}(\cdot)}^{(t)} = \text{ecdf of } \left\{ \acute{\eta}_{(\cdot)}^{(\acute{b},t)} \right\}_{1 \leq \acute{b} \leq \acute{B}} . \tag{4.5}$$
In particular, $\hat{F}_{\acute{\eta}(\mathrm{MCMC})}^{(t)}$ is treated as the "benchmark curve", and is used to compare with $\hat{F}_{\acute{\eta}(\cdot)}^{(t)}$ from the other 5 methods, by keeping track of the total variation $TV_{\acute{\eta}(\cdot)}^{(t)}$ between $\hat{F}_{\acute{\eta}(\mathrm{MCMC})}^{(t)}$ and $\hat{F}_{\acute{\eta}(\cdot)}^{(t)}$. Note that $\acute{\eta}_{(\mathrm{MCMC})}^{(\acute{b},t)}$ is computed as $NMI\left(\grave{\boldsymbol{z}}_{(\cdot)}^{(\acute{b},t)} \ , \ \acute{\boldsymbol{z}}_{(\mathrm{MCMC})}^{(\acute{b},t)} \right)$, where $\grave{\boldsymbol{z}}_{(\cdot)}^{(\acute{b},t)}$ 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 held-out log probability indicates "better fit 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 $TV_{\acute{\eta}(\cdot)}^{(t)}$.

The simulation results are presented in Figure 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 $\left\{TV_{\hat{\eta}(\cdot)}^{(t)}\right\}_{1\leq t\leq 10}$ 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 Σ 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 held-out log probability in all simulation settings (with VI and RW DP-rich register slightly lower values in Simulation Setting II).

4.2 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 $K_{\text{true}} = 3$). 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 $K_{\text{true}} = 3$).

We refer readers to the supplementary material for details about specifying the priors for MCMC, VI, RW SDP-rich and RW SDP-means. λ_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) 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 (4.2)) and ecdf curves for NMI $\hat{F}_{\hat{\eta}(\cdot)}$ computed based on randomly-sampled pairs of posterior cluster assignments (see, Equation (4.5)). 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 (4.4)).

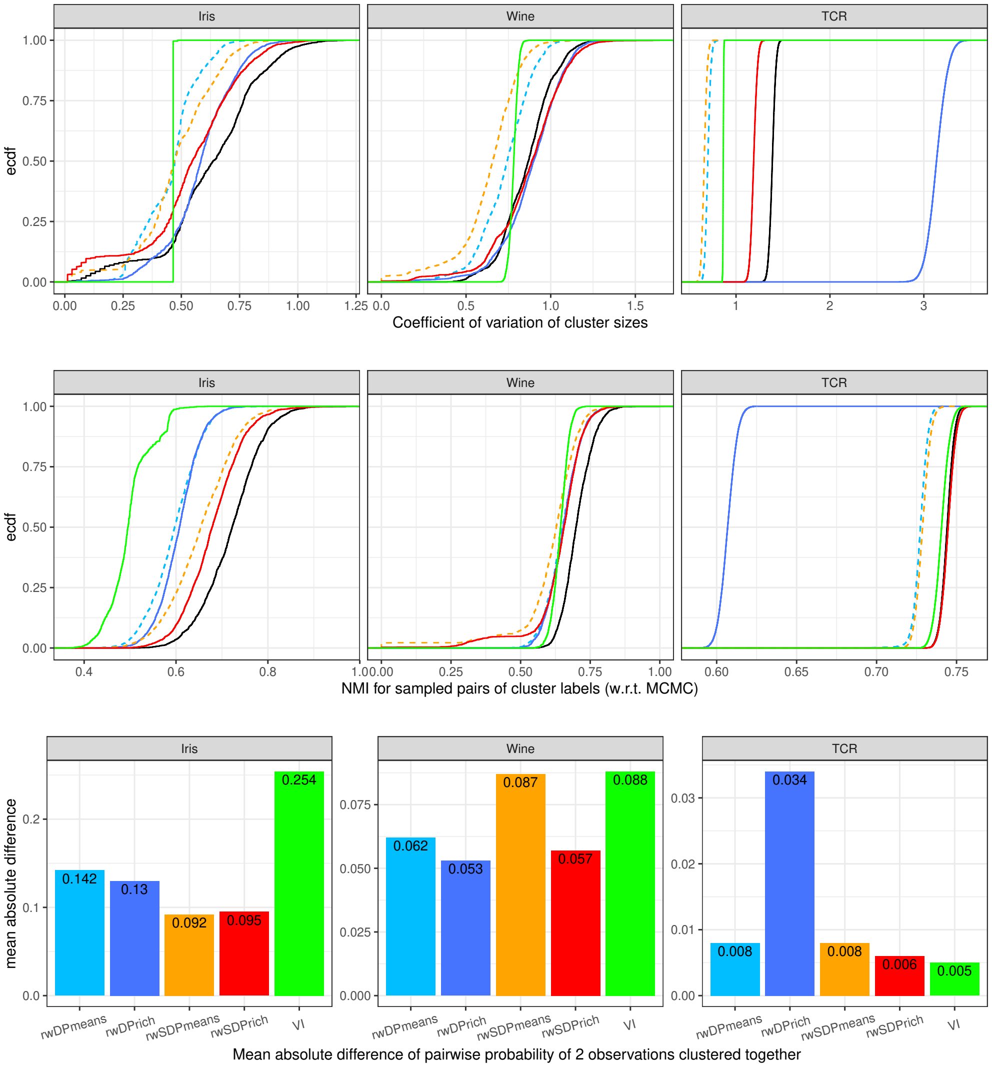


Figure 5: The ecdf curves of CoV of cluster sizes (see, Equation (4.2)) and the ecdf curves of NMI (see, Equation (4.5)) comparing randomly-sampled pairs of cluster assignments for all 6 methods – MCMC (solid black), VI (solid green), RW DP-means (dashed light-blue), RW DP-rich (solid dark-blue), RW 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 (4.4)) for the other 5 methods, among the 3 benchmark and motivating data examples.

The results are presented in Figure 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 $\left\{\boldsymbol{z}_{(\mathrm{VI})}^{(b)}\right\}_{1\leq b\leq B}$ 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. 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 Motivating Example: T-cell Receptor Data

Now we consider our motivating T-cell Receptor (TCR) data example. Specifically, Zahm et al. (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 $d = 3$. 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. Again, λ_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 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 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 $\check{p}_{(\mathrm{VI})}$ is the smallest, which is closely followed by $\check{p}_{(\mathrm{rwSDP-rich})}$ (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 $\lambda_2^{\text{rwDP-rich}}$: if $\lambda_2^{\text{rwDP-rich}}$ is too small, then its performance is no different from a RW DP-means implementation; on the other hand, if $\lambda_2^{\text{rwDP-rich}}$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.

5 Large Sample Asymptotics

We first furnish additional theoretical details relating the Bayesian NPL framework (Lyddon et al., 2018) from Section 2 to the clustering parameters in our random-weighting approach for mixture models from Section 3. Then, we present our asymptotic results under this framework.

Specifically, from Section 2, we are interested in expected loss $\mathscr{L}(t,F)$ where posterior sampling of F is approximated with Bayesian bootstrap F_w, which leads to

$$\mathscr{L}(t,F_w) = \int_\Omega \tilde{l}(t,y)\,dF_w(y) = \int_\Omega l(t,y)\,dF_w(y) + \lambda_0 l_0(t) = \sum_{i=1}^n w_i l(t,y_i) + \lambda_0 l_0(t), \tag{5.1}$$

where $(w_1,\cdots,w_n) \sim Dir(1,\cdots,1)$ and $\lambda_0 > 0$ is supplied by the analyst. Then, we arrive at (2.3) 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 λ in (2.3). We refer interested readers to the supplementary material for detailed derivation of this Bayesian NPL approach. The following subsections will instead focus on the setup in (5.1).

5.1 Connection to Random Weighting for Mixture Models

RW K-means

First, let $A_K = \{a_1, \cdots, a_K\}$ be a set of K points on $\mathbb{R}^d$, and we want to find A_K that minimizes

$$\inf_{A_K} \mathscr{L}(A_K, F_w) = \inf_{A_K} \left\{ \int_\Omega \min_{a \in A_K} \|y - a\|_2^2 \, dF_w(y) \right\} = \min_{(\boldsymbol{\mu}, \boldsymbol{z})} \left\{ \sum_{k=1}^{K} \sum_{i: z_i = k} w_i \|y_i - \mu_k\|_2^2 \right\}, \tag{5.2}$$

where $(w_1, \cdots, w_n) \sim Dir(1, \cdots, 1)$, and F_w is the Bayesian bootstrap defined in (5.1). From the discussion in Section 2, it is evident that the RHS of (5.2) is related to (3.14). 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

$$V_k := \left\{ y_i \in \Omega : \|y_i - a_k\|_2^2 < \|y_i - a_{k'}\|_2^2 \quad \text{for all } k' \neq k \right\}. \tag{5.3}$$

Then, $\bigcup_k V_k$ is the Voronoi tessellation (e.g., Urschel, 2017) of Ω, and the set $\Omega \backslash (\bigcup_k V_k)$ 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 $\Omega \backslash (\bigcup_k V_k)$ as the ***Voronoi partition*** (denoted as $\mathcal{P}$).

Let $A_{n,K}^w := \arg\min_{A_K} \mathscr{L}(A_K, F_w)$ be the minimizer of (5.2), where the subscript n indicates that the set of centroids changes with dataset. Then, the NPL posterior distribution $\Pi_n\left(A_K \middle| y\right)$ has a corresponding (approximate) posterior density

$$\pi\left(A_K \middle| y\right) = \int \pi\left(A_K \middle| F_w\right) d\pi(F_w), \tag{5.4}$$

where the approximation comes from the fact that the integral in (5.4) is performed w.r.t. the Bayesian bootstrap approximation F_w, and

$$\pi\left(A_K \middle| F_w\right) = \delta_{A_{n,K}^w(F_w)}\left(A_K\right). \tag{5.5}$$

The delta arises because A_K is a deterministic functional of F_w from (5.2). Notice that $A_{n,K}^w$ depends on F_w, i.e. $A_{n,K}^w$ depends on the independent dirichlet weights $(w_1, \cdots, w_n)$. See also, Section 2.3 of Fong et al. (2019) for a discussion of Bayesian NPL posterior. In addition, for the delta in (5.5) 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 (5.2) 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 $\Pi_n\left(\mathcal{P}_K \middle| y\right)$ has a corresponding (approximate) posterior density

$$\pi\left(\mathcal{P}_K \middle| y\right) = \int \pi\left(\mathcal{P}_K \middle| F_w\right) d\pi(F_w), \tag{5.6}$$

where $\pi\left(\mathcal{P}_K \middle| F_w\right) = \delta_{\mathcal{P}_{n,K}^w(F_w)}\left(\mathcal{P}_K\right)$.

RW DP-means

To derive the RW DP-means objective function from the Bayesian NPL perspective, let $A_{\lambda_0} = \{a_1, \cdots, a_\kappa\}$ be a set of κ points on $\mathbb{R}^d$ for $\kappa = |A_{\lambda_0}| \in \mathbb{N}$, and we want to find A_{λ_0} that minimizes

$$\begin{aligned} \inf_{A_{\lambda_0}} \mathscr{L}(A_{\lambda_0}, F_w) &= \inf_{A_{\lambda_0}} \left\{ \int_\Omega \min_{a \in A_{\lambda_0}} \|y - a\|_2^2 \, dF_w(y) + \lambda_0 \cdot |A_{\lambda_0}| \right\} \\ &= \min_{(\boldsymbol{\mu}, \boldsymbol{z}, \kappa)} \left\{ \sum_{k=1}^{\kappa} \sum_{i: z_i = k} w_i \|y_i - \mu_k\|_2^2 + \lambda_0 \kappa \right\}, \end{aligned} \tag{5.7}$$

where $\lambda_0 > 0$ is a tuning parameter, F_w is the Bayesian bootstrap defined in (5.1), and $(w_1, \cdots, w_n) \sim Dir(1, \cdots, 1)$. In addition, denote $\mathcal{P}_{\lambda_0}$ as the *Voronoi partition* associated with A_{λ_0}. The subscript λ_0 in A_{λ_0} and $\mathcal{P}_{\lambda_0}$ highlights the fact that the variables depend on the tuning parameter λ_0. We note that (5.7) is also in line with the concept of Loss NPL introduced in Section 2.6 of Fong et al. (2019). Again, from Section 2, it is evident that (5.7) is related to (3.13).

Let $A^w_{n,\lambda_0} := \arg\min_{A_{\lambda_0}} \mathscr{L}(A_{\lambda_0}, F_w)$ be the minimizer of (5.7), and let $\mathcal{P}^w_{n,\lambda_0}$ be the *Voronoi partition* associated with A^w_{n,λ_0}. Again, the subscript n indicates that the solutions change with dataset. Then, the NPL posterior distribution $\Pi_n \left(A_{\lambda_0} \middle| y \right)$ has a corresponding (approximate) posterior density

$$\pi \left(A_{\lambda_0} \middle| y \right) = \int \pi \left(A_{\lambda_0} \middle| F_w \right) d\pi(F_w), \tag{5.8}$$

where $\pi \left(A_{\lambda_0} \middle| F_w \right) = \delta_{A^w_{n,\lambda_0}(F_w)} \left(A_{\lambda_0} \right)$, whereas the NPL posterior distribution $\Pi_n \left(\mathcal{P}_{\lambda_0} \middle| y \right)$ has a corresponding (approximate) posterior density

$$\pi \left(\mathcal{P}_{\lambda_0} \middle| y \right) = \int \pi \left(\mathcal{P}_{\lambda_0} \middle| F_w \right) d\pi(F_w), \tag{5.9}$$

where $\pi \left(\mathcal{P}_{\lambda_0} \middle| F_w \right) = \delta_{\mathcal{P}^w_{n,\lambda_0}(F_w)} \left(\mathcal{P}_{\lambda_0} \right)$.

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 Σ is pre-specified with a symmetric positive-definite matrix Σ_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 2, by recognizing $(2\lambda_1)$ to be $(\lambda_0 \sum_{i=1}^n W_i)$ for $W_i \overset{iid}{\sim} Exp(1)$, we can re-specify the RW SDP-means setup into the form of

$$\inf_{A_{(\lambda_0, \Sigma_0)}} \mathscr{L}(A_{(\lambda_0, \Sigma_0)}, F_w)$$

$$= \inf_{A_{(\lambda_0,\Sigma_0)}} \left\{ \int_\Omega \min_{a \in A_{(\lambda_0,\Sigma_0)}} (y-a)'\Sigma_0^{-1}(y-a)\, dF_w(y) + \lambda_0 \cdot \left| A_{(\lambda_0,\Sigma_0)} \right| \right\} \tag{5.10}$$
$$= \min_{(\boldsymbol{\mu},\boldsymbol{z},\kappa)} \left\{ \sum_{k=1}^{\kappa} \sum_{i:z_i=k} w_i (y_i - \mu_k)'\Sigma_0^{-1}(y_i-\mu_k) + \lambda_0 \kappa \right\},$$

where $\kappa = \left|A_{(\lambda_0,\Sigma_0)}\right|$, and $(w_1, \cdots, w_n) \sim Dir(1, \cdots, 1)$. We also denote $\mathcal{P}_{(\lambda_0,\Sigma_0)}$ as the *Voronoi partition* associated with $A_{(\lambda_0,\Sigma_0)}$. The subscript (λ_0, Σ_0) highlights the fact that the variables depend on the choices of λ_0 and Σ_0. Basically, the setup in (5.7) is a special case of (5.10) with $\Sigma_0 = I_d$.

Again, let $A^w_{n,(\lambda_0,\Sigma_0)} := \arg\min_{A_{(\lambda_0,\Sigma_0)}} \mathscr{L}(A_{(\lambda_0,\Sigma_0)}, F_w)$ be the minimizer of (5.10), and let $\mathcal{P}^w_{n,(\lambda_0,\Sigma_0)}$ be the *Voronoi partition* associated with $A^w_{n,(\lambda_0,\Sigma_0)}$. Then, the NPL posterior distribution $\Pi_n \left(A_{(\lambda_0,\Sigma_0)} \middle| y\right)$ has a corresponding (approximate) posterior density

$$\pi\left(A_{(\lambda_0,\Sigma_0)} \middle| y\right) = \int \pi\left(A_{(\lambda_0,\Sigma_0)} \middle| F_w\right) d\pi(F_w), \tag{5.11}$$

where $\pi\left(A_{(\lambda_0,\Sigma_0)} \middle| F_w\right) = \delta_{A^w_{n,(\lambda_0,\Sigma_0)}(F_w)}\left(A_{(\lambda_0,\Sigma_0)}\right)$, whereas the NPL posterior distribution $\Pi_n\left(\mathcal{P}_{(\lambda_0,\Sigma_0)} \middle| y\right)$ has a corresponding (approximate) posterior density

$$\pi\left(\mathcal{P}_{(\lambda_0,\Sigma_0)} \middle| y\right) = \int \pi\left(\mathcal{P}_{(\lambda_0,\Sigma_0)} \middle| F_w\right) d\pi(F_w), \tag{5.12}$$

where $\pi\left(\mathcal{P}_{(\lambda_0,\Sigma_0)} \middle| F_w\right) = \delta_{\mathcal{P}^w_{n,(\lambda_0,\Sigma_0)}(F_w)}\left(\mathcal{P}_{(\lambda_0,\Sigma_0)}\right)$.

5.2 Asymptotic Results

Lyddon et al. (2018) and Fong et al. (2019) mentioned about the Bayesian NPL strong consistency property of the solutions/samples $\theta_w := \arg\min_{t\in\Theta} \mathscr{L}(t, F_w)$ in (5.1), which relies on the strong consistency property of the Bayesian bootstrap, i.e.

$$F_w \longrightarrow F_* \quad a.s.\ P_{F_*}^{(\infty)}, \tag{5.13}$$

where the convergence of random measure in (5.13) 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).

Here, we present a rigorous discussion on how the solutions/samples obtained by our random-weighting mixture models satisfy the Bayesian NPL strong consistency property under certain regularity conditions.

First, we consider the Hausdorff metric $\mathcal{D}_\mathcal{H}$ to metrize the solution space of the sets of centroids.

Theorem 5.1. ***(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*

$$A_{*,K} = \arg\min_{A_K} \mathscr{L}(A_K, F_*) = \arg\min_{A_K} \left\{ \int_\Omega \min_{a \in A_K} \|y - a\|_2^2 \, dF_*(y) \right\}. \tag{5.14}$$

Then, for every $\epsilon > 0$, as $n \to \infty$,

$$\Pi_n \left(A_K : \mathcal{D}_{\mathcal{H}} \left(A_K, A_{*,K} \right) > \epsilon \middle| y \right) \to 0 \quad a.s. \ P_{F_*}^{(\infty)}, \tag{5.15}$$

where the posterior distribution Π_n of A_K is defined in (5.4).

(b) ***(RW DP-means)*** *suppose that under F_* and the choice of $\lambda_0 > 0$, there exists a unique set A_{*,λ_0} of $\kappa = |A_{*,\lambda_0}|$ points on $\mathbb{R}^d$ such that*

$$A_{*,\lambda_0} = \arg\min_{A_{\lambda_0}} \mathscr{L}(A_{\lambda_0}, F_*) = \arg\min_{A_{\lambda_0}} \left\{ \int_\Omega \min_{a \in A_{\lambda_0}} \|y - a\|_2^2 \, dF_*(y) + \lambda_0 \kappa \right\}. \tag{5.16}$$

Then, for every $\epsilon > 0$, as $n \to \infty$,

$$\Pi_n \left(A_{\lambda_0} : \mathcal{D}_{\mathcal{H}} \left(A_{\lambda_0}, A_{*,\lambda_0} \right) > \epsilon \middle| y \right) \to 0 \quad a.s. \ P_{F_*}^{(\infty)}, \tag{5.17}$$

where the posterior distribution Π_n of A_{λ_0} is defined in (5.8).

(c) ***(RW SDP-means)*** *suppose that under F_* and the choices of $\lambda_0 > 0$ and symmetric positive-definite Σ_0, there exists a unique set $A_{*,(\lambda_0,\Sigma_0)}$ of $\kappa = \left|A_{*,(\lambda_0,\Sigma_0)}\right|$ points on $\mathbb{R}^d$ such that*

$$A_{*,(\lambda_0,\Sigma_0)} = \arg\min_{A_{(\lambda_0,\Sigma_0)}} \left\{ \int_\Omega \min_{a \in A_{(\lambda_0,\Sigma_0)}} \left[(y-a)'\Sigma_0^{-1}(y-a) \right] dF_*(y) + \lambda_0 \kappa \right\}. \tag{5.18}$$

Then, for every $\epsilon > 0$, as $n \to \infty$,

$$\Pi_n \left(A_{(\lambda_0,\Sigma_0)} : \mathcal{D}_{\mathcal{H}} \left(A_{(\lambda_0,\Sigma_0)}, A_{*,(\lambda_0,\Sigma_0)} \right) > \epsilon \middle| y \right) \to 0 \quad a.s. \ P_{F_*}^{(\infty)}, \tag{5.19}$$

where the posterior distribution Π_n of $A_{(\lambda_0,\Sigma_0)}$ is defined in (5.11).

Comments on Assumptions of Theorem 5.1. 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 Π_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 κ_{*,λ_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_{λ_0} or $A_{(\lambda_0,\Sigma_0)}$) is not unique. For instance, consider the case where Ω 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 5.2.

Next, we consider Leonardi and Tamanini (2002)'s metric (denoted as $\mathcal{D}_{\mathcal{L}}$) to metrize the solution space of the *Voronoi partitions*. 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.

Theorem 5.2. ***(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 5.1. Let $\mathcal{P}_{*,K}$ be the Voronoi partition corresponding to $A_{*,K}$. Then, for every $\epsilon > 0$, as $n \to \infty$,

$$\Pi_n \left(\mathcal{P}_K : \mathcal{D}_{\mathcal{L}} \left(\mathcal{P}_K, \mathcal{P}_{*,K}\right) > \epsilon \middle| y\right) \to 0 \quad a.s.\ P_{F_*}^{(\infty)}, \tag{5.20}$$

where the posterior distribution Π_n of $\mathcal{P}_K$ is defined in (5.6).

(b) adopt the assumptions in part (b) of Theorem 5.1. Let $\mathcal{P}_{*,\lambda_0}$ be the Voronoi partition corresponding to A_{*,λ_0}. Then, for every $\epsilon > 0$, as $n \to \infty$,

$$\Pi_n \left(\mathcal{P}_{\lambda_0} : \mathcal{D}_{\mathcal{L}} \left(\mathcal{P}_{\lambda_0}, \mathcal{P}_{*,\lambda_0}\right) > \epsilon \middle| y\right) \to 0 \quad a.s.\ P_{F_*}^{(\infty)}, \tag{5.21}$$

where the posterior distribution Π_n of $\mathcal{P}_{\lambda_0}$ is defined in (5.9).

(c) adopt the assumptions in part (c) of Theorem 5.1. Let $\mathcal{P}_{*,(\lambda_0,\Sigma_0)}$ be the Voronoi partition corresponding to $A_{*,(\lambda_0,\Sigma_0)}$. Then, for every $\epsilon > 0$, as $n \to \infty$,

$$\Pi_n \left(\mathcal{P}_{(\lambda_0,\Sigma_0)} : \mathcal{D}_{\mathcal{L}} \left(\mathcal{P}_{(\lambda_0,\Sigma_0)}, \mathcal{P}_{*,(\lambda_0,\Sigma_0)}\right) > \epsilon \middle| y\right) \to 0 \quad a.s.\ P_{F_*}^{(\infty)}, \tag{5.22}$$

where the posterior distribution Π_n of $\mathcal{P}_{(\lambda_0,\Sigma_0)}$ is defined in (5.12).

Remark 5.1. *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 5.1 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*

$$\sum_{k=1}^{\kappa} \log \Gamma(n_k) \le \log \Gamma(n) = \mathcal{O}(n \log n)$$

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.

The following result is a direct consequence of the assumptions adopted in Theorem 5.2.

Lemma 5.1 **(Zero-measure of decision boundaries).** *Adopt assumptions in Theorem 5.2. Then, the decision-boundary set* $\Omega \backslash \bigcup_k V_k$ *(i.e., the set of points which are equidistant from more than one centroid) of a Voronoi partition has measure zero.*

Proof of Lemma 5.1. 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 Σ_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_*. □

Comments on Assumptions of Theorem 5.2. 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 5.1. 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 5.1. 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 $\mathcal{D}_{\mathcal{L}}\left(\{[0, {}^1\!/\!_2], ({}^1\!/\!_2, 1]\}, \{[0, {}^1\!/\!_2), [{}^1\!/\!_2, 1]\}\right) = 0$.

Connection to Centroidal Voronoi Tessellation We also want to point out that the objective function

$$\min_{a \in A_K} \left\{ \int_{\mathcal{P}_K} g(y-a) \; dF_*(y) \right\}, \tag{5.23}$$

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 Σ_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 5.1 and 5.2 is exactly (5.23) with squared Euclidean distance, whereas for RW DP-means, its asymptotic limit in Theorems 5.1 and 5.2 could be thought of as applying (5.23) 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 $K = 1, 2, \cdots$). Similar argument is also applicable to the asymptotic limit of RW SDP-means (with a fixed Σ_0) in Theorems 5.1 and 5.2, 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 (5.23) that apply to every possible scenario. We refer interested readers to the aforementioned references on the characterization of (5.23) in certain specific settings, which is beyond the scope of this paper.

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 Ω consisting of $K_0 \geq 1$ disjoint clusters $\left\{\mathcal{C}_1^0, \cdots, \mathcal{C}_{K_0}^0\right\}$. Conditional on this partition $\mathcal{P}_0$, the RW K-means (3.14), RW DP-means(3.13) and RW DP-rich (3.12) setups are reduced to obtaining random-weighting centroids

$$\mu_{n,k}^w = \frac{\sum_{i \in \mathcal{C}_k^0} W_i y_i}{\sum_{i \in \mathcal{C}_k^0} W_i} \tag{5.24}$$

for $k = 1, \cdots, K_0$ and $W_i \overset{iid}{\sim} Exp(1)$, 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 Σ_w and) random-weighting centroids

$$\mu_{n,k}^w = \frac{\sum_{i \in \mathcal{C}_k^0} W_i y_i + \xi_0 \mu_0}{\sum_{i \in \mathcal{C}_k^0} W_i + \xi_0} \tag{5.25}$$

for $k = 1, \cdots, K_0$ and $W_i \overset{iid}{\sim} Exp(1)$. Conditional on data with a fixed partition $\mathcal{P}_0$ of Ω, we prove that these random-weighting centroids, which are centered on their corresponding sample mean of the cluster

$$\hat{\mu}_{n,k} = \frac{1}{n_k} \sum_{i \in \mathcal{C}_k^0} y_i \tag{5.26}$$

for $n_k = \left|\mathcal{C}_k^0\right|$, are asymptotically normal as $n \to \infty$. 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 5.3 (Asymptotic Normality). *Assume that F_* has finite second moments. Suppose Ω 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 (5.26) and the random-weighting centroid $\mu_{n,k}^w$ defined in (5.24) or (5.25). Then, for $k = 1, \cdots, K_0$ and for any Borel set $B \subset \mathbb{R}^d$, as $n_k \to \infty$,*

$$P\left(\sqrt{n_k}\left(\mu_{n,k}^w - \hat{\mu}_{n,k}\right) \in B \middle| y\right) \to P(Z \in B) \quad a.s.\ P_{F_*}^{(\infty)},$$

where $Z \sim N_d\left(0\ ,\ V_{*,k}^{\mathcal{P}_0}\right)$.

The proof for Theorem 5.3 is provided in the supplementary material.

Supplementary Material

Supplementary material: Random Weighting in Finite or Countable Mixture Models. **The supplementary material has many sections....write more ...**

References

Anderson, E. (1935). "The irises of the Gaspe Peninsula." *Bulletin of the American Iris Society*, 59: 2–5. 23

Arthur, D. and Vassilvitskii, S. (2007). "k-means++: the advantages of careful seeding." In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 1027–1035. 9

Bissiri, P. G., Holmes, C. C., and 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. 2

Blackwell, D. and MacQueen, J. B. (1973). "Ferguson Distribution via Polya Urn Schemes." *The Annals of Statistics*, 1(2): 353–355. 5

Blei, D. M. and Jordan, M. I. (2006). "Variational inference for dirichlet process mixtures." *Bayesian Analysis*, 1(1): 121–144. 2, 17

Broderick, T., Kulis, B., and 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. 6

Corradin, R., Canale, A., and Nipoti, B. (2021). *BNPmix: Bayesian Nonparametric Mixture Models*. R package version 0.2.8.
URL https://CRAN.R-project.org/package=BNPmix 17

Dahl, D. B. (2009). "Modal clustering in a class of product partition models." *Bayesian Analysis*, 4(2): 243–264. 10, 11

Dash, P., Fiore-Gartland, A. J., Hertz, T., Wang, G. C., Sharma, S., Souquette, A., Crawford, J. C., Clemens, E. B., Nguyen, T. H. O., Kedzierska, K., Gruta, N. L. L., Bradley, P., and Thomas, P. G. (2017). "Quantifiable predictive features define epitope-specific T cell receptor repertoires." *Nature*, 547: 89–93. 25

Fang, Y. and Wang, J. (2012). "Selection of the number of clusters via the bootstrap method." *Computational Statistics and Data Analysis*, 56: 468–477. 17

Fong, E., Lyddon, S., and Holmes, C. (2019). "Scalable nonparametric sampling from multimodal posteriors with the posterior bootstrap." *Proceedings of the 36th International Conference on Machine Learning*. 2, 3, 4, 5, 15, 25, 27, 28, 29

Ghosal, S. and van der Vaart, A. (2017). *Fundamentals of Nonparametric Bayesian Inference*. Cambridge University Press. 29

Hartigan, J. A. and 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. 3

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, second edition. 3, 13, 25

Jara, A., Hanson, T., Quintana, F., Mueller, P., and Rosner, G. (2011). "Bayesian Semi-

and Nonparametric Modeling in R." *Journal of Statistical Software*, 40(5): 1–30. URL http://www.jstatsoft.org/v40/i05/ 17

Jensen, S. T. and Liu, J. S. (2008). "Bayesian Clustering of Transcription Factor Binding Motifs." *Journal of the American Statistical Association*, 103(481): 188–200. 14

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. 2, 17

Kulis, B. and Jordan, M. I. (2012). "Revisiting k-means: New Algorithms via Bayesian Nonparametrics." *Proceedings of the 29 th International Conference on Machine Learning*. 6, 9, 10

Leonardi, G. P. and Tamanini, I. (2002). "Metric spaces of partitions, and Caccioppoli partitions." *Advances in Mathematical Sciences and Applications*, 12(2): 725–753. 31, 32

Lyddon, S., Holmes, C., and Walker, S. (2019). "General Bayesian updating and the loss-likelihood bootstrap." *Biometrika*, 106(2): 465–478. 2

Lyddon, S., Walker, S., and Holmes, C. (2018). "Nonparametric learning from Bayesian models with randomized objective functions." In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, 2075–2085. 2, 26, 29

Ma, X., Korthauer, K., Kendziorski, C., and 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. 1

McLachlan, G. J., Lee, S. X., and Rathnayake, S. I. (2019). "Finite mixture models." *Annual review of statistics and its application*, 6: 355–378. 2

Mossel, E. and Vigoda, E. (2006). "Limitations of Markov Chain Monte Carlo Algorithms for Bayesian Inference of Phylogeny." *The Annals of Applied Probability*, 16(4): 2215–2234. 2

Müller, P., Quintana, F. A., Jara, A., and Hanson, T. (2015). *Bayesian nonparametric data analysis*. Springer. 1, 2, 4

Nemeth, C. and Fearnhead, P. (2021). "Stochastic Gradient Markov Chain Monte Carlo." *Journal of the American Statistical Association*, 116(533): 433–450. 2

Ng, T. L. and Newton, M. A. (2022). "Random weighting in LASSO regression." *Electronic Journal of Statistics*, 16(1): 3430 – 3481. URL https://doi.org/10.1214/22-EJS2020 5

Paul, D. and Das, S. (2020). "A Bayesian non-parametric approach for automatic clustering with feature weighting." *Stat*, 9(1). 15, 17

Pollard, D. (1981). "Strong consistency of K-means clustering." *The Annals of Statistics*, 9(1): 135–140. 4, 30

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R

Foundation for Statistical Computing, Vienna, Austria.
URL https://www.R-project.org/ 17

Richter, R. and Alexa, M. (2015). "Mahalanobis centroidal Voronoi tessellations." *Computers and Graphics*, 46: 48–54. 32

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. 10

Scrucca, L., Fop, M., Murphy, T. B., and Raftery, A. E. (2016). "mclust 5: clustering, classification and density estimation using Gaussian finite mixture models." *The R Journal*, 8(1): 289–317. 1, 13

Sethuraman, J. (1994). "A constructive definition of dirichlet priors." *Statistica Sinica*, 4: 639–650. 19

Stephens, M. (2000). "Dealing with label switching in mixture models." *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 62(4): 795–809. 22, 31

Tseng, P. (2001). "Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization." *Journal of Optimization Theory and Applications*, 109(3): 475–494. 8

Urschel, J. C. (2017). "On the characterization and uniqueness of centroidal Voronoi tessellations." *SIAM Journal on Numerical Analysis*, 55(3): 1525–1547. 27, 31, 32

Vinh, N. X., Epps, J., and Bailey, J. (2010). "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance." *Journal of Machine Learning Research*, 11: 2837–2854. 11, 17, 22

Vujovic, M., Degn, K. F., Marin, F. I., Schaap-Johansen, A.-L., Chain, B., Andresen, T. L., Kaplinsky, J., and Marcatili, P. (2020). "T cell receptor sequence clustering and antigen specificity." *Computational and Structural Biotechnology Journal*, 18: 2166–2173. 25

Wade, S. and Ghahramani, Z. (2018). "Bayesian cluster analysis: point estimation and credible balls (with discussion)." *Bayesian Analysis*, 13(2): 559–626. 1

Welling, M. and Teh, Y. W. (2011). "Bayesian Learning via Stochastic Gradient Langevin Dynamics." *Proceedings of International Conference on Machine Learning*. 2

Williams, G. J. (2011). *Data Mining with Rattle and R: The art of excavating data for knowledge discovery*. Use R! Springer.
URL 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 23

Zahm, C., Ng, T. L., Newton, M. A., and McNeel, D. (2022). "Antigen specificity of T-cell receptors." *In preparation*. 25

Zuanetti, D. A., Muller, P., Zhu, Y., Yang, S., and Ji, Y. (2019). "Bayesian nonparametric clustering for large data sets." *Statistics and Computing*, 29: 203–215. 2, 17

Acknowledgments

The authors are supported by ....