Supplementary Material: Random Weighting in Finite or Countable Mixture Models

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

1 Implementation details of DP-rich

1.1 Singleton clusters

Suppose we are now in the cluster reassignment step of the DP-rich and we are considering the data point y_i that is sitting alone in cluster $\mathcal{C}_{k'}$, i.e. $i \in \mathcal{C}_{k'}$ where $n_{k'} = 1$. Then, problems will arise in the DP-rich algorithm. First, $\log(n_{k',-i}) = \log(n_{k'} - 1)$ would be undefined. Furthermore, we should not consider $d_{i,\kappa+1}$ in this case, since taking y_i out of $\mathcal{C}_{k'}$ (i.e. $\mathcal{C}_{k'}$ is emptied and subsequently dropped) and putting it into a brand new cluster does not increase the total number of clusters κ in the objective function.

Therefore, if $i \in \mathcal{C}_{k'}$ where $n_{k'} = 1$, we should do the following instead: first, update

$$\mu_{k'} = y_i, \tag{1.1}$$

which is in fact the centroid that we would initialize had we created a new cluster. Then, update the cost to join an existing cluster to be

$$d_{ik} = \|y_i - \mu_k\|_2^2 - \lambda_2 \log(n_{k',-i}) - \lambda_1 \tag{1.2}$$

for all $k \in \{1, \cdots, \kappa\} \backslash \{k'\}$, whereas

$$d_{ik'} = 0, \tag{1.3}$$

and finally, set $d_{i,\kappa+1} = \infty$ since we do not consider creating another new cluster in this case. If $\arg\min_{k \in \{1,\cdots,\kappa\}} d_{ik} \neq k'$, then cluster $C_{k'}$ is dropped and no observation will ever be allocated to this cluster in subsequent steps. Again, all these modified formulae still ensure that the objective function never increase, and the local convergence property of the DP-rich algorithm is still ensured.

1.2 Initialization of algorithm

Here are some details that we need to consider when we initialize the DP-rich algorithm.

Rich-gets-richer penalty

Recall that the cost to join an existing cluster is given by

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

[*]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

When we first initialize the algorithm where all data points are grouped together, the term $\lambda_2 \log(n_{k,-i})$ may be too overwhelming, and the algorithm may fail to break up this single initial cluster.

To overcome this problem, we suggest to set $\lambda_2 = 0$ in the first (few) epoch(s) of the DP-rich algorithm. One epoch here is defined as one iteration of cluster reassignment through all data points.

However, care has to be taken here, because too many epochs with $\lambda_2 = 0$ before allowing $\lambda_2 > 0$ may lead to the DP-rich algorithm to saddle at a local solution that is too similar to the DP-means algorithm. From our numerical experiments, we find that one epoch with $\lambda_2 = 0$ (before allowing $\lambda_2 > 0$) leads to reasonable performance by the algorithm. We report that Raykov et al. (2016) faced similar problem with their own algorithm and suggested similar workarounds.

Initial cluster labels

We note that the DP-rich algorithm could also be initialized with more than one cluster. However, we caution the readers against attempts to game or "improve" initialization by using other methods such as the standard K-means, as this might lead to the DP-rich algorithm saddling at suboptimal local solution, i.e. the DP-rich algorithm might produce a solution with a smaller objective had we initialized the algorithm by randomly assigning the observations. Therefore, in this paper, we follow the convention of Kulis and Jordan (2012) as well as Paul and Das (2020), where all observations are grouped together when we initialize the algorithm.

2 Additional details of RW SDP-rich

2.1 Modifying DPM's negative log-posterior

Here, we provide further details about the modification of the negative log-posterior of the DPM of Normals to arrive at the objective function of RW SDP-rich. Specifically, we begin with $h(\Sigma) = \Sigma/\xi_0$ and $p(\Sigma)$ is inverse-Wishart with ν_0 degrees of freedom and a symmetric positive-definite scale matrix ψ_0. Following the Bayesian NPL framework where we assign i.i.d. standard Exponential random weights $(W_1, \cdots, W_n)$ on the

likelihood component of the DPM, we have:

$$
\begin{aligned}
&p_w\left(\boldsymbol{Y}, \boldsymbol{z}, \kappa, \{\mu_k\}_{k=1}^{\kappa}, \Sigma\right) \\
&:= p_w\left(\boldsymbol{Y} \middle| \boldsymbol{z}, \kappa, \{\mu_k\}_{k=1}^{\kappa}, \Sigma\right) \times p\left(\{\mu_k\}_{k=1}^{\kappa} \middle| \Sigma, \boldsymbol{z}, \kappa\right) \times p\left(\Sigma\right) \times p(\boldsymbol{z}, \kappa) \\
&\propto (2\pi)^{-\frac{d}{2}\sum_{i=1}^{n} W_i} |\Sigma|^{-\frac{1}{2}\sum_{i=1}^{n} W_i} \exp\left\{-\frac{1}{2}\sum_{k=1}^{\kappa}\sum_{i:z_i=k} W_i (y_i-\mu_k)'\Sigma^{-1}(y_i-\mu_k)\right\} \\
&\times (2\pi)^{-\frac{d\kappa}{2}} \xi_0^{\frac{d\kappa}{2}} \mid \Sigma|^{-\frac{\kappa}{2}} \exp\left\{-\frac{\xi_0}{2}\sum_{k=1}^{\kappa}(\mu_k-\mu_0)'\Sigma^{-1}(\mu_k-\mu_0)\right\} \\
&\times |\Sigma|^{-(\nu_0+d+1)/2} \exp\left\{-\frac{1}{2}\operatorname{Tr}\left(\psi_0\Sigma^{-1}\right)\right\} \times \alpha_0^{\kappa-1}\frac{\Gamma(\alpha_0+1)}{\Gamma(\alpha_0+n)}\prod_{k=1}^{\kappa}\Gamma(n_k).
\end{aligned}
$$

Then, taking negative log,

$$
\begin{aligned}
&-\log p_w(\boldsymbol{Y}, \boldsymbol{z}, \kappa, \boldsymbol{\mu}, \Sigma) \\
&= \frac{1}{2}\left[\sum_{k=1}^{\kappa}\sum_{i:z_i=k} W_i (y_i-\mu_k)'\Sigma^{-1}(y_i-\mu_k) + \xi_0\sum_{k=1}^{\kappa}(\mu_k-\mu_0)'\Sigma^{-1}(\mu_k-\mu_0) + \operatorname{Tr}\left(\psi_0\Sigma^{-1}\right)\right] \\
&+ \left(\sum_{i=1}^{n} W_i + \nu_0 + d + 1 + \kappa\right)\log\left|\Sigma^{1/2}\right| \qquad (2.1) \\
&+ \kappa\log\left(\left(\frac{2\pi}{\xi_0}\right)^{d/2}\cdot\frac{1}{\alpha_0}\right) - \sum_{k=1}^{\kappa}\log\left[\Gamma(n_k)\right] + \text{other terms.}
\end{aligned}
$$

Borrowing the idea of the DP-rich algorithm, we replace the coefficient of κ in (2.1) with a tuning parameter $\lambda_1 > 0$, and introduce another tuning parameter $\lambda_2 > 0$ for the *rgr* term in (2.1). Recall that λ_1 allows direct calibration by the analyst to tune the number of clusters obtained by the algorithm, whereas λ_2 controls the magnitude of the algorithm's *rgr* effect. Then, we are left with

$$
\begin{aligned}
&\frac{1}{2}\left[\sum_{k=1}^{\kappa}\sum_{i:z_i=k} W_i (y_i-\mu_k)'\Sigma^{-1}(y_i-\mu_k) + \xi_0\sum_{k=1}^{\kappa}(\mu_k-\mu_0)'\Sigma^{-1}(\mu_k-\mu_0) + \operatorname{Tr}\left(\psi_0\Sigma^{-1}\right)\right] \\
&+ \left(\sum_{i=1}^{n} W_i + \nu_0 + d + 1 + \kappa\right)\log\left|\Sigma^{1/2}\right| + \lambda_1\kappa - \lambda_2\sum_{k=1}^{\kappa}\log\left[\Gamma(n_k)\right].
\end{aligned} \qquad (2.2)
$$

Notice how (2.2) looks very similar to our RW SDP-rich objective function, except for the coefficient of $\log\left|\Sigma^{1/2}\right|$. Now, if we had adopted (2.2) as our objective function, then solving for Σ (while holding $\kappa, \boldsymbol{z}$ and $\{\mu_k\}_{1\le k\le\kappa}$ constant) would have yielded

$$
\frac{\sum_{k=1}^{\kappa}\sum_{i:z_i=k} W_i (y_i-\mu_k)(y_i-\mu_k)' + \xi_0\sum_{k=1}^{\kappa}(\mu_k-\mu_0)(\mu_k-\mu_0)' + \psi_0}{\sum_{i=1}^{n} W_i + \nu_0 + d + 1 + \kappa}. \qquad (2.3)
$$

The presence of the term κ in the denominator of (2.3) is problematic. First, recall from the construction of DP-rich, ξ_0 would be small if there is prior belief for larger number of clusters. Thus, the term $\sum_{k=1}^{\kappa}(\mu_k-\mu_0)(\mu_k-\mu_0)'$ is moderated by ξ_0, and as sample size increases, the term $\sum_{k=1}^{\kappa}\sum_{i:z_i=k} W_i(y_i-\mu_k)(y_i-\mu_k)'$ would dominate the other two terms in the numerator of (2.3). However, in the denominator of (2.3), if κ also increases with $\sum_{i=1}^{n} W_i$ as sample size increases, $|\Sigma|^{1/2}$ becomes smaller. This problem becomes evident when we consider the cost to create a new cluster:

$$d^{w}_{i,\kappa+1} = \frac{1}{2}\frac{\xi_0 W_i}{\xi_0 + W_i}(y_i-\mu_0)'\Sigma^{-1}(y_i-\mu_0) + \lambda_1.$$

The larger κ, the smaller $|\Sigma|^{1/2}$, the smaller the cost $d^{w}_{i,\kappa+1}$ to create a new cluster, which leads to a cascade of more clusters getting created and so on.

To break this vicious cycle, we modify (2.2) by replacing the coefficient of $\log\left|\Sigma^{1/2}\right|$ in (2.2) with

$$\sum_{i=1}^{n} W_i + \nu_0 - d - 1, \tag{2.4}$$

such that when sample size is small (and thus number of clusters κ is not huge), Σ is approximately equal to its inverse-Wishart prior mean $\psi_0/(\nu_0-d-1)$. This helps to ensure stability of the variable Σ (and thus the stability of the algorithm itself) especially when sample size is small, and also justifies initialization of Σ_w with $\psi_0/(\nu_0-d-1)$ in the beginning of the RW SDP-rich algorithm. As sample size increases, the denominator of Σ is heavily influenced by the term $\sum_{i=1}^{n} W_i$, and so Σ will be approximately

$$\frac{\sum_{k=1}^{\kappa}\sum_{i:z_i=k} W_i(y_i-\mu_k)(y_i-\mu_k)'}{\sum_{i=1}^{n} W_i}.$$

2.2 Proof of Lemma (Local convergence of RW SDP-rich)

This proof is an extension of the proof for Kulis and Jordan (2012)'s Theorem 3.1.

Proof. The reassignment step results in a non-increasing objective since the weighted Mahalanobis distance between a point and its newly-assigned weighted cluster centroid (discounted by the corresponding *rgr* "gravitational pull") is smaller than that before the re-allocation occurs. If an observation is assigned to a new cluster, the cost of creating the new cluster is cheaper than to assign the observation to any one of the existing clusters, which results in a reduction in objective. Dropping empty cluster(s) – for example, dropping cluster $\mathcal{C}_k$ – decreases the objective by $\lambda_1+\xi_0(\mu_k-\mu_0)'\Sigma^{-1}(\mu_k-\mu_0)$. Similarly, the cluster parameter updates lead to a non-increasing objective since the objective function of the RW SDP-rich is convex in $\boldsymbol{\mu}$ and Σ conditional on $(\kappa, \boldsymbol{z})$. The algorithm will converge locally because the objective function cannot increase, and that there are only a finite number of possible clusterings of the data. □

2.3 Singleton clusters

The issue discussed in Section 1.1 is also applicable to the RW SDP-rich algorithm, except that now we have to replace Equation (1.1) with

$$\mu_{k'}^{w} = \frac{W_i y_i + \xi_0 \mu_0}{W_i + \xi_0},$$

and replace Equation (1.2) with

$$d_{ik}^{w} = \frac{1}{2} W_i (y_i - \mu_k)' \Sigma^{-1} (y_i - \mu_k) - \lambda_2 \log(n_{k',-i}) - \lambda_1,$$

and replace Equation (1.3) with

$$d_{ik'}^{w} = \frac{1}{2} \frac{\xi_0 W_i}{\xi_0 + W_i} (y_i - \mu_0)' \Sigma^{-1} (y_i - \mu_0).$$

2.4 Initialization of algorithm

Again, the issues about the *rgr* penalty and initial cluster assignments, which are discussed in Section 1.2, are also relevant when we initialize the RW SDP-rich algorithm.

Furthermore, here we also need to consider about the issue regarding initialization of Σ. Recall that Σ is updated with the formula

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

When we first initialize the algorithm where all data points are grouped together, the corresponding initialized Σ will be approximately the overall weighted sum-of-squares

$$\frac{\sum_{i=1}^{n} W_i (y_i - \bar{\mu}_w)(y_i - \bar{\mu}_w)'}{\sum_{i=1}^{n} W_i},$$

where $\bar{\mu}_w = {}^{(\sum_{i=1}^{n} W_i y_i + \xi_0 \mu_0)}/_{(\sum_{i=1}^{n} W_i + \xi_0)}$ represents the weighted grand centroid. We may be "overestimating" the dispersion among data points from the same cluster in this case. Hence, we suggest fixing $\Sigma = {}^{\psi_0}/_{(\nu_0 - d - 1)}$ (or, for the case of diagonal covariance structure, $\sigma_j^2 = {}^{b_{0,j}}/_{(a_{0,j} - 1)}$ for all j) during the first epoch of the RW SDP-rich algorithm to ensure more stable performance by the algorithm, based on our experience in the numerical experiments.

2.5 Formulae for diagonal covariance structure

Conditional on an existing partition $(\kappa, \boldsymbol{z})$, for $j = 1 \cdots, d$,

$$\mu_{kj}^{w} = \frac{\sum_{i:z_i=k} W_i y_{ij} + \xi_{0j} \mu_{0j}}{\sum_{i:z_i=k} W_i + \xi_{0j}},$$

and

$$(\sigma_w^2)_j = \frac{\sum_{k=1}^{\kappa}\sum_{i:z_i=k} W_i(y_{ij}-\mu_{kj})^2 + \xi_{0j}\sum_{k=1}^{\kappa}(\mu_{kj}-\mu_{0j})^2 + 2b_{0j}}{(\sum_{i=1}^{n} w_i + 2a_{0j}) - 2},$$

and in the beginning of the algorithm, we initialize $(\sigma_w^2)_j$ with ${}^{b_{0j}}/_{(a_{0j}-1)}$. In the cluster reassignment step, the cost d_{ik}^w of assigning observation y_i to an existing cluster $\mathcal{C}_k$ is now

$$d_{ik}^w = \sum_{j=1}^{d} \frac{W_i(y_{ij}-\mu_{kj})^2}{2\sigma_j^2} - \lambda_{n,2}\log(n_k)$$

for $k = 1, \cdots, \kappa$, whereas the cost to create a new cluster for observation y_i is

$$d_{i,\kappa+1}^w = \sum_{j=1}^{d} \frac{\xi_{0j}W_i}{\xi_{0j}+W_i}\frac{(y_{ij}-\mu_{0j})^2}{2\sigma_j^2} + \lambda_{n,1}.$$

Similarly, if $i \in \mathcal{C}_{k'}$ where $n_{k'} = 1$, update $\boldsymbol{\mu}_{k'}^w$ as we have discussed in Section 2.3, and update

$$d_{ik}^w = \sum_{j=1}^{d} \frac{W_i(y_{ij}-\mu_{kj})^2}{2\sigma_j^2} - \lambda_{n,2}\log(n_{k',-i}) - \lambda_{n,1} \quad \text{for} \quad k \in \{1,\cdots,\kappa\}\backslash\{k'\},$$

$$d_{ik'}^w = \sum_{j=1}^{d} \frac{\xi_{0j}W_i}{\xi_{0j}+W_i}\frac{(y_{ij}-\mu_{0j})^2}{2\sigma_j^2},$$

$$d_{i,\kappa+1}^w = \infty.$$

2.6 Regularization parameters

Choice of λ_2

We now compare the performances of RW DP-rich and RW SDP-rich specified with different values of $\lambda_2^{\text{rwDP-rich}}$ and $\lambda_2^{\text{rwSDP-rich}}$ using a set of simulations. Specifically, we adopt the "*full-covariance higher-correlation*" Simulation Setting as well as its corresponding MCMC prior specifications, which we already outlined in the Main Text.

Here, we compare $\lambda_2^{\text{rwDP-rich}} \in \{0, 0.5, 1, 2\}$ and $\lambda_2^{\text{rwSDP-rich}} \in \{0, 0.5, 1, 2\}$, using the different comparison criteria that we described in the Main Text. Recall that setting $\lambda_2^{\text{rwDP-rich}} = 0$ corresponds to RW DP-means whereas specifying $\lambda_2^{\text{rwSDP-rich}} = 0$ corresponds to RW SDP-means.

Figure 1 shows the performances of these different methods. Ideally, we want all performance criteria for these methods that involve total variation to be as close to zero as possible, which indicates higher degree of "similarity" to MCMC samples. Meanwhile, the average held-out log probability for these methods should be as high as possible, and

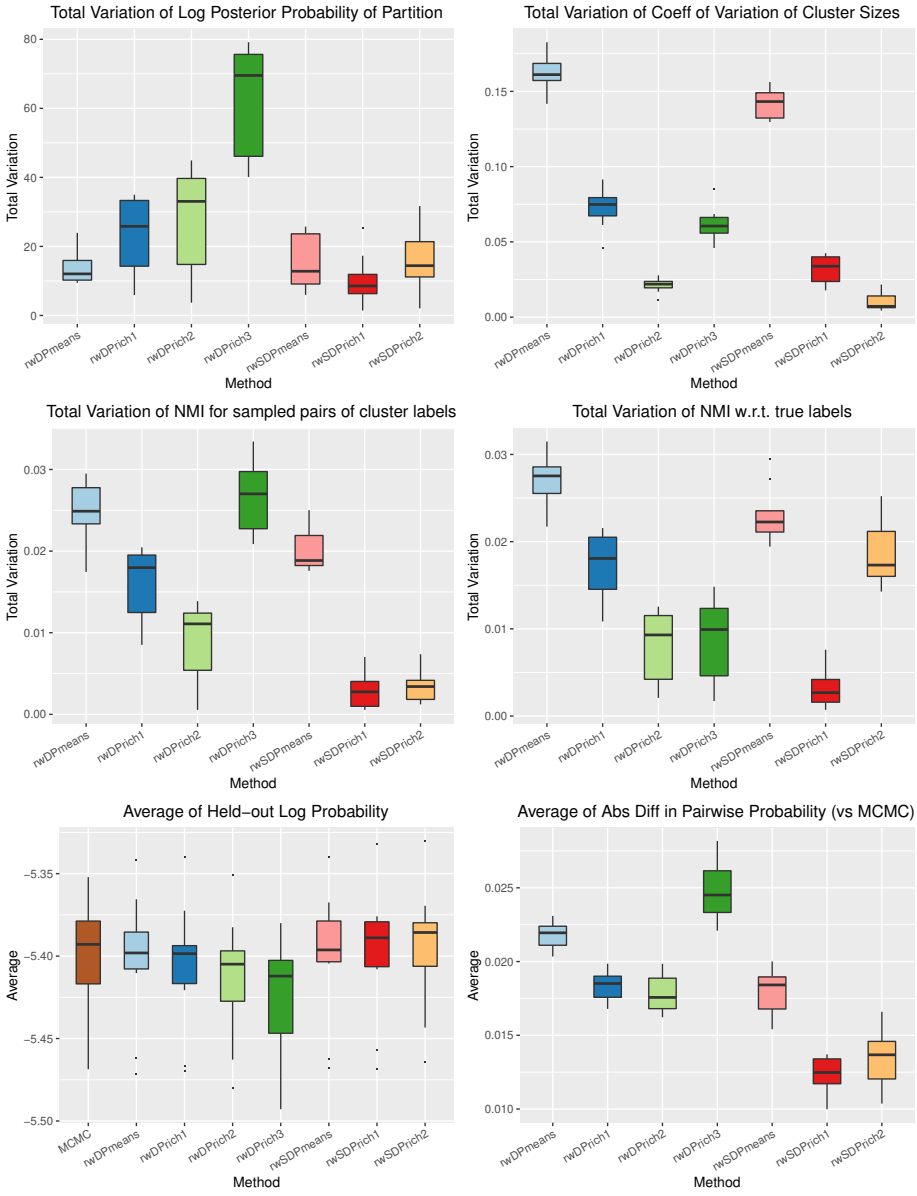

Figure 1: Comparing performances of RW DP-rich and RW SDP-rich using different *rgr* tuning parameters. For RW DP-rich, we specify $\lambda_2^{\text{rwDP-rich}}$ to be 0 (denoted `rwDPmeans`), 0.5 (denoted `rwDPrich1`), 1 (denoted `rwDPrich2`) and 2 (denoted `rwDPrich3`). For RW SDP-rich, we specify $\lambda_2^{\text{rwSDP-rich}}$ to be 0 (denoted `rwSDPmeans`), 0.5 (denoted `rwSDPrich1`) and 1 (denoted `rwSDPrich2`).

the average of absolute difference in pairwise probability (of two observations clustered together) as compared to MCMC samples should be as close to zero as possible.

From Figure 1, it appears that setting $\lambda_2^{\text{rwDP-rich}} = 1$ (denoted as `rwDPrich2`) leads to the best performance in most of the comparison criteria among the RW DP-rich contenders (which is unsurprising, since the true variance of the mixture components is indeed equal to 1), whereas specifying $\lambda_2^{\text{rwSDP-rich}} = 0.5$ (denoted as `rwSDPrich1`) leads to the best performance in most of the comparison criteria among the RW SDP-rich candidates. In particular, the boxplots for RW SDP-rich with $\lambda_2 = 2$ (denoted as `rwSDPrich3`) are not shown in Figure 1 because their performances are the worst (way worse than all the other methods).

Calibrating λ_1

Here, we use a Binary Search procedure (e.g. Raykov et al., 2016) to tune λ_1 such that the (average of) random-weighting samples of κ "matches" a targeted number of clusters K_{targ} for any one of the four random-weighting algorithms: RW DP-means, RW DP-rich, RW SDP-means or RW SDP-rich. In our numerical experiments, we specify K_{targ} to be the posterior mean of κ obtained from standard MCMC method, which (in most cases of our numerical experiments) are also close to the MAP of κ.

Briefly, we start off by picking a starting value of $\lambda_1^{(0)}$ via, say, a farthest-first approach (e.g. Kulis and Jordan, 2012). Next, we generate and store B' sets of i.i.d. standard Exponential random weights $\{\boldsymbol{W}_b^{1:n}\}_{1\leq b\leq B'} := \{(W_{1,b}, \cdots, W_{n,b})\}_{1\leq b\leq B'}$. For calibration purpose, we could use a "cheaper" random-weighting scheme by using a smaller number of draws, say, $B' = 1000$ in order to save computational time, and yet perform reasonably well in our numerical experiments.

Each $b = 1, \cdots, B'$ set of these random weights $\boldsymbol{W}_b^{1:n}$ is then fed into the random-weighting procedure specified with $\lambda_1 = \lambda_1^{(0)}$ to obtain $\{\kappa_b^w\}_{1\leq b\leq B}$. Next, we compute the average of $\kappa_b^w - K_{\text{targ}}$

$$d_{\lambda_1^{(0)}} := \frac{1}{B'} \sum_{b=1}^{B'} (\kappa_b^w - K_{\text{targ}}) \,. \tag{2.5}$$

If $d_{\lambda_1^{(0)}} > 0$, this indicates that on average, there are more clusters obtained by the random-weighting procedure than K_{targ}, i.e. we want *less* clusters, and thus we need to scale up λ_1 to inflict a heavier penalty on κ in the objective function. Therefore, we set $\underline{\lambda}_1 = \lambda_1^{(0)}$ to be the lower bound of λ_1, and specify the next potential tuning parameter value $\lambda_1^{(1)} = 2 \times \lambda_1^{(0)}$.

On the other hand, if $d_{\lambda_1^{(0)}} < 0$, we want *more* clusters, and thus we need to reduce λ_1 to inflict less penalty on κ in the objective function. Therefore, we set $\bar{\lambda}_1 = \lambda_1^{(0)}$ to be the upper bound of λ_1, and specify the next potential tuning parameter value $\lambda_1^{(1)} = \frac{\lambda_1^{(0)}}{2}$. We substitute $\lambda_1^{(1)}$ into the random-weighting procedure and repeat the steps above to obtain $d_{\lambda_1^{(1)}}$ and so on.

Suppose after the $(t-1)^{th}$ step, we have our lower bound $\underline{\lambda}_1$ and upper bound $\bar{\lambda}_1$. Then, we could specify

$$\lambda_1^{(t)} = \frac{\underline{\lambda}_1 + \bar{\lambda}_1}{2} \tag{2.6}$$

and subsequently obtain $d_{\lambda_1^{(t)}}$. If $d_{\lambda_1^{(t)}} < 0$, update the lower bound $\underline{\lambda}_1 = \lambda_1^{(t)}$ and set $\lambda_1^{(t+1)}$ via the same formula (2.6) using this new lower bound. On the other hand, if $d_{\lambda_1^{(t)}} > 0$, update the upper bound $\bar{\lambda}_1 = \lambda_1^{(t)}$, and compute $\lambda_1^{(t+1)}$ with (2.6) using this new upper bound. Repeat this process until $|d_{\lambda_1}| \leq \epsilon_\lambda$, where ϵ_λ is some minute tolerance level determined by the analyst.

Alternatively, we can also use this Binary Search procedure to nail down a reasonable range of $\left[\underline{\lambda}_1, \bar{\lambda}_1\right]$, so that we can then use a grid search approach to find the value of λ_1 that produces the smallest $|d_{\lambda_1}|$.

3 Additional details of RW K-means

The RW K-means procedure mentioned in the main text is outlined in Algorithm 1. In particular, the standard K-means optimization procedures, such as that by Hartigan and Wong (1979), could still be used to optimize the RW K-means objective function, except that now, weighted Euclidean distance is considered in the cluster reassignment step and weighted centroids are updated instead. Arthur and Vassilvitskii (2007)'s discussion about careful seeding is also relevant here to improve the local solutions obtained by the RW K-means procedure.

Algorithm 1 Random-weighting K-means

Require: data $\{y_1, \cdots, y_n\}$, number of clusters K, number of posterior draws B.

1: **for** $b = 1, \cdots, B$ **do**
2: Initialize with K centroids.
3: Draw $W_i \overset{iid}{\sim} Exp(1)\ \forall\ i = 1, \cdots, n$.
4: Optimize the RW K-means objective function, and store $\mu_{k,b}^w$ for $k = 1, \cdots, K$, and $z_{i,b}^w$ for $i = 1, \cdots, n$.
5: **end for**

Ensure: B samples of cluster centroids $\left\{\mu_{k,b}^w\right\}_{1 \leq k \leq K; 1 \leq b \leq B}$, and B samples of cluster assignments $\left\{z_{i,b}^w\right\}_{1 \leq i \leq n; 1 \leq b \leq B}$.

While the regular K-means has long been known to be the small-variance asymptotics (SVA) of the Gaussian finite-mixture model (GMM) (e.g., Hastie et al., 2009), we verify in Lemma 3.1 that this SVA property remains applicable to their random-weighting counterparts.

Lemma 3.1 (**RW K-means as the SVA of RW GMM**). *For a Gaussian finite mixture with common variance* $\Sigma_k = \sigma^2 I_d\ \forall\ k = 1, \cdots, K$, *the negative loglikelihood of*

its random-weighting counterpart (Fong et al., 2019)

$$\prod_{k=1}^{K}\prod_{i:z_i=k}\left[p_k f_k\left(y_i|\mu_k,\Sigma_k\right)\right]^{W_i}, \tag{3.1}$$

multiplied with $2\sigma^2$*, converges to the objective function of random-weighting K-means*

$$\mathscr{L}_K^{rwKmeans}(\boldsymbol{\mu},\boldsymbol{z}) := \sum_{k=1}^{K}\sum_{i:z_i=k} W_i \left\|y_i-\mu_k\right\|_2^2 \tag{3.2}$$

when we push $\sigma^2 \to 0$*.*

Proof of Lemma 3.1. Taking negative log of (3.1) where $\Sigma_k = \sigma^2 I_d \;\forall\; k = 1,\cdots,K$, we have

$$-\sum_{k=1}^{K}(\log p_k)\sum_{i:z_i=k} W_i + \sum_{k=1}^{K}\sum_{i:z_i=k} W_i\left[\frac{d}{2}\log\sigma^2 + \frac{\left\|y_i-\mu_k\right\|_2^2}{2\sigma^2}\right] + \frac{d\log(2\pi)}{2}\sum_{i=1}^{n} W_i. \tag{3.3}$$

Multiply (3.3) with $2\sigma^2$, then push $\sigma^2 \to 0$ to obtain the objective function in (3.2). □

4 More details for theoretical properties

4.1 Probability space

There are two sources of variation in the random-weighting setup under the Bayesian NPL framework, namely the data $\{y_1, y_2, \cdots\}$ and the random weights $\{w_1, w_2, \cdots\}$. Consequently, we consider a common probability space with the common probability measure $P = P_{F_*}^{(\infty)} \times P_{\tilde{F}_w}$, where $P_{F_*}^{(\infty)}$ is the probability measure of the observed data, and $P_{\tilde{F}_w}$ is the probability measure of the triangular array of random weights (Mason and Newton, 1992) that arises from Bayesian bootstrap $\tilde{F}_W$. The use of product measure reflects the independence of data and random weights. The study of asymptotic properties under the random-weighting framework is not new; see, for example, Mason and Newton (1992), Lyddon et al. (2019) and Ng and Newton (2020).

4.2 Derivation for Bayesian NPL framework

Under the Bayesian NPL framework (Fong et al., 2019), since F_* is unknown, we place a Dirichlet Process (DP) prior on the sampling distribution

$$F|(\alpha_0, F_0) \sim DP(\alpha_0, F_0), \tag{4.1}$$

where α_0 is the concentration parameter and F_0 is the prior centering measure. We want to remind readers that F_* is not required to be in some "neighborhood" of F_0,

and $DP(\alpha_0, F_0)$ in (4.1) is NOT related to the DPM working model that we mentioned in the main text.

From the conjugacy of the DP (e.g., Ghosal and van der Vaart, 2017), the posterior of F becomes

$$F|y := \tilde{F} \sim DP\left(\alpha_0 + n \ , \ \frac{\alpha_0}{\alpha_0 + n} F_0 + \frac{1}{\alpha_0 + n} \sum_{i=1}^{n} \delta_{y_i}\right), \tag{4.2}$$

where δ denotes the dirac measure. Based on the stick-breaking construction (Sethuraman, 1994) of the DP, we have

$$\arg\min_{t\in\Theta} \mathscr{L}(t, \tilde{F}) = \arg\min_{t\in\Theta} \int l(t, y) d\tilde{F}(y) = \arg\min_{t\in\Theta} \left\{ \sum_{j=1}^{\infty} \breve{w}_j l(t, \breve{y}_j) \right\}, \tag{4.3}$$

where $\{\breve{w}_j\}_{j=1}^{\infty} \sim GEM(\alpha_0 + n)$ and

$$\breve{y}_j \overset{iid}{\sim} \left(\frac{\alpha_0}{\alpha_0 + n} F_0 + \frac{1}{\alpha_0 + n} \sum_{i=1}^{n} \delta_{y_i} \right)$$

for all j (Ishwaran and Zarepour, 2002). Exact posterior calculation of (4.3) requires infinite sampling, but could be approximated with

$$\arg\min_{t\in\Theta} \left\{ \sum_{i=1}^{n} w_i l(t, y_i) + \sum_{j=1}^{T} \tilde{w}_j l(t, \tilde{y}_j) \right\} \tag{4.4}$$

for large truncation limit T, where $\tilde{y}_j \overset{iid}{\sim} F_0$ and

$$(w_1, \cdots, w_n, \tilde{w}_1, \cdots, \tilde{w}_T) \sim Dir\left(1, \cdots, 1, {}^{\alpha_0}/_{T}, \cdots, {}^{\alpha_0}/_{T}\right).$$

As $n \to \infty$ such that $n \gg T$, data realizations overwhelm prior information, which motivates Rubin (1981)'s Bayesian bootstrap approximation of $\tilde{F}$ (henceforth denoted as F_w) by setting $\alpha_0 = 0$ in (4.4):

$$\mathscr{L}(t, F_w) = \int_{\Omega} \tilde{l}(t, y)\, dF_w(y) = \int_{\Omega} l(t, y)\, dF_w(y) + \lambda_0 l_0(t) = \sum_{i=1}^{n} w_i l(t, y_i) + \lambda_0 l_0(t), \tag{4.5}$$

where $(w_1, \cdots, w_n) \sim Dir(1, \cdots, 1)$. See also Muliere and Secchi (1996) on further interpretations of the Bayesian bootstrap. Since

$$(w_1, \cdots, w_n) \overset{d}{=} \left(\frac{W_1}{\sum_{i=1}^{n} W_i}, \cdots, \frac{W_n}{\sum_{i=1}^{n} W_i} \right)$$

where $W_i \overset{iid}{\sim} Exp(1)$, solving $\min_{t\in\Theta} \mathscr{L}(t, F_w)$ in (4.5) is equivalent to optimizing

$$\min_{t\in\Theta} \left\{ \sum_{i=1}^{n} \Big[W_i \cdot l(t, y_i) \Big] + \left(\lambda_0 \sum_{i=1}^{n} W_i \right) \cdot l_0(t) \right\}.$$

In practice, we replace the $(\lambda_0 \sum_{i=1}^n W_i)$ term with a unifying regularization parameter $\lambda > 0$ to be calibrated by the analyst, and finally we arrive at

$$\mathscr{L}_\lambda(t, \boldsymbol{W}) := \sum_{i=1}^{n} \Big[W_i l(t, y_i) \Big] + \lambda l_0(t). \tag{4.6}$$

4.3 Additional proofs

First, recall the strong consistency property of the Bayesian bootstrap, i.e.

$$F_w \longrightarrow F_* \quad a.s.\ P_{F_*}^{(\infty)}, \tag{4.7}$$

where the convergence of random measure in (4.7) 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).

To prove Theorems 5.1 (Bayesian NPL strong consistency for the set of centroids), we need the following two lemmas.

Lemma 4.1 (**Finiteness of asymptotic limits**). *Adopt assumptions in Theorem 5.1. Then,*

(a) $\kappa_{*,\lambda_0} < \infty$ *and* $\kappa_{*,(\lambda_0,\Sigma_0)} < \infty$.

(b) all centroids in $A_{*,K}$, A_{*,λ_0} *and* $A_{*,(\lambda_0,\Sigma_0)}$ *are finite.*

Proof of Lemma 4.1. The finite second moment requirement on F_* leads to

$$\int_\Omega \|y\|_2^2 dF_*(y) < \infty \quad \text{and} \quad \int_\Omega y\Sigma_0^{-1} y \; dF_*(y) < \infty$$

for any symmetric positive definite Σ_0. Then, for any point $r \in \mathbb{R}^d$,

$$\int_\Omega \min_{a \in A_K} \|y - a\|_2^2 \; dF_*(y) \leq \int_\Omega \|y - r\|_2^2 dF_*(y) \leq 4\|r\|_2^2 + 4 \int_\Omega \|y\|_2^2 dF_*(y), \tag{4.8}$$

where the RHS of (4.8) is finite only if point r is finite. Thus, all centroids in $A_{*,K}$ have to be finite, otherwise contradiction occurs. Similarly, for any point $r \in \mathbb{R}^d$ and for any symmetric positive-definite Σ_0,

$$\begin{aligned} \int_\Omega \min_{a \in A_K} \left[(y-a)'\Sigma_0^{-1}(y-a) \right] dF_*(y) &\leq \int_\Omega (y-r)'\Sigma_0^{-1}(y-r) dF_*(y) \\ &\leq 4r'\Sigma_0^{-1} r + 4 \int_\Omega y'\Sigma_0^{-1} y \; dF_*(y), \end{aligned} \tag{4.9}$$

where the second line of (4.9) is finite only if point r is finite. We remind the readers that for RW DP-means and RW SDP-means, κ is data-driven instead of being pre-specified by the analyst, but we need (4.8) and (4.9) to prove part (a) of the lemma.

For RW DP-means, we have, from (4.8),

$$\int_\Omega \min_{a\in A} \|y-a\|_2^2 dF_*(y) = \mathcal{O}(1)$$

for any partition $\mathcal{P}$ of Ω that is associated with the set of centroids $A = \{a_1, \cdots, a_\kappa\}$ where $\kappa = |A|$. Increasing the number of clusters indefinitely shrinks the integral to 0 but increases $\kappa \to \infty$, which in turn pushes the objective to ∞. Thus, a minimizer must arrive at finite κ_{*,λ_0}.

Next, since $\kappa_{*,\lambda_0} < \infty$, we could think of minimizing

$$\left\{ \int_\Omega \min_{a\in A_{\lambda_0}} \|y-a\|_2^2 \; dF_*(y) + \lambda_0 \kappa \right\}$$

as minimizing the LHS of (4.8) on the grid of positive integers $\mathbb{N}$, and evaluate the corresponding objectives penalized with $\lambda_0 K$ for $K = 1, \cdots, \kappa_{*,\lambda_0}^{\max}$ where $\kappa_{*,\lambda_0} \le \kappa_{*,\lambda_0}^{\max}$. Finally, pick the clustering that has the lowest objective. Using similar argument in (4.8), we ensure that all the centroids in A_{*,λ_0} are finite.

Finally, For RW SDP-means with a fixed symmetric positive-definite Σ_0, we could use (4.9) and similar arguments to establish that $\kappa_{*,(\lambda_0,\Sigma_0)} < \infty$, and that all centroids in $A_{*,(\lambda_0,\Sigma_0)}$ are finite. □

Recall, from the definitions in Theorem 5.1, that

$$A_{*,K} = \arg\min_{A_K} \mathscr{L}(A_K, F_*) = \arg\min_{A_K} \left\{ \int_\Omega \min_{a\in A_K} \|y-a\|_2^2 \; dF_*(y) \right\}. \tag{4.10}$$

$$A_{*,\lambda_0} = \arg\min_{A_{\lambda_0}} \mathscr{L}(A_{\lambda_0}, F_*) = \arg\min_{A_{\lambda_0}} \left\{ \int_\Omega \min_{a\in A_{\lambda_0}} \|y-a\|_2^2 \; dF_*(y) + \lambda_0 \kappa \right\}. \tag{4.11}$$

$$A_{*,(\lambda_0,\Sigma_0)} = \arg\min_{A_{(\lambda_0,\Sigma_0)}} \left\{ \int_\Omega \min_{a\in A_{(\lambda_0,\Sigma_0)}} \left[(y-a)'\Sigma_0^{-1}(y-a)\right] dF_*(y) + \lambda_0 \kappa \right\}. \tag{4.12}$$

Lemma 4.2 (Continuity for sets of centroids). *Adopt assumptions in Theorem 5.1. Then,*

(a) $\mathscr{L}(A_K, F_*)$ *is a continuous function of* A_K *in (4.10).*

(b) $\mathscr{L}(A_{\lambda_0}, F_*)$ *is a continuous function of* A_{λ_0} *in (4.11).*

(c) $\mathscr{L}(A_{(\lambda_0,\Sigma_0)}, F_*)$ *is a continuous function of* $A_{(\lambda_0,\Sigma_0)}$ *in (4.12).*

Proof of Lemma 4.2. To prove continuity, we need to first invoke Lemma 4.1 to ensure that $\kappa_{*,\lambda_0} < \infty$ and $\kappa_{*,(\lambda_0,\Sigma_0)} < \infty$, and all centroids in $A_{*,K}$, A_{*,λ_0} and $A_{*,(\lambda_0,\Sigma_0)}$ are finite. Then, we immediately obtain part (a) of the Lemma by invoking the established result in Pollard (1981) about the continuity of $\mathscr{L}(A_K, F_*)$ as a function of A_K in (4.10).

To obtain part (b), we still need to construct a similar $\epsilon-\zeta$ proof; i.e., for every $\epsilon > 0$, there exists $\zeta_\epsilon > 0$ such that $\mathcal{D}_{\mathcal{H}}\left(A_{\lambda_0}, A'_{\lambda_0}\right) < \zeta_\epsilon$ implies $\left|\mathscr{L}(A_{\lambda_0}, F_*) - \mathscr{L}(A'_{\lambda_0}, F_*)\right| < \epsilon$. First, note that if $|A_{\lambda_0}| \neq \left|A'_{\lambda_0}\right|$, there is always another set of centroids A''_{λ_0} such that $|A_{\lambda_0}| = \left|A''_{\lambda_0}\right|$ and $\left|\mathscr{L}(A_{\lambda_0}, F_*) - \mathscr{L}(A''_{\lambda_0}, F_*)\right| < \left|\mathscr{L}(A_{\lambda_0}, F_*) - \mathscr{L}(A'_{\lambda_0}, F_*)\right|$, because A''_{λ_0} can be exactly the same as A_{λ_0} except for one of its centroids, where its coordinates are slightly perturbed such that the resulting change in the sum of squares of its cluster is smaller than $\left|\mathscr{L}(A_{\lambda_0}, F_*) - \mathscr{L}(A'_{\lambda_0}, F_*)\right|$. Hence, we only need to consider the case where $|A_{\lambda_0}| = \left|A'_{\lambda_0}\right|$ in the $\epsilon-\zeta$ proof, and Pollard (1981)'s continuity result immediately follows.

Similar arguments can also be applied to prove part (c) by replacing the Euclidean distance with the Mahalanobis distance. □

Proof of Theorem 5.1. We first invoke Lemma 4.2 to establish that $\mathscr{L}(A, F_*)$ is a continuous function of A, where A stands for A_K, A_{λ_0} and $A_{(\lambda_0,\Sigma_0)}$ in (4.10), (4.11) and (4.12) respectively. Then, by noticing that A_K, A_{λ_0} and $A_{(\lambda_0,\Sigma_0)}$ are deterministic functionals of F_w, whereas $A_{*,K}$, A_{*,λ_0} and $A_{*,(\lambda_0,\Sigma_0)}$ are deterministic functionals of F_*, the convergence of the posterior distribution Π_n of A_K, A_{λ_0} and $A_{(\lambda_0,\Sigma_0)}$ follows immediately from (4.7). □

Before we prove Theorem 5.2 (Bayesian NPL strong consistency for partition), we need the following results.

Lemma 4.3 (**Continuity for partitions**). *Adopt assumptions in Theorem 5.2. Then,*

(a) $\mathscr{L}\left(\mathcal{P}_K\left(A_K\right), F_*\right)$ *is a continuous function of* $\mathcal{P}_K$*, where* $\mathcal{P}_K$ *is the Voronoi partition associated with* A_K *in (4.10).*

(b) $\mathscr{L}\left(\mathcal{P}_{\lambda_0}\left(A_{\lambda_0}\right), F_*\right)$ *is a continuous function of* $\mathcal{P}_{\lambda_0}$*, where* $\mathcal{P}_{\lambda_0}$ *is the Voronoi partition associated with* A_{λ_0} *in (4.11).*

(c) $\mathscr{L}\left(\mathcal{P}_{(\lambda_0,\Sigma_0)}\left(A_{(\lambda_0,\Sigma_0)}\right), F_*\right)$ *is a continuous function of* $\mathcal{P}_{(\lambda_0,\Sigma_0)}$*, where* $\mathcal{P}_{(\lambda_0,\Sigma_0)}$ *is the Voronoi partition associated with* $A_{(\lambda_0,\Sigma_0)}$ *in (4.12).*

Proof of Lemma 4.3. Again, due to Lemma 4.1, we ensure that $\kappa_{*,\lambda_0} < \infty$ and $\kappa_{*,(\lambda_0,\Sigma_0)} < \infty$, and all centroids in $A_{*,K}$, A_{*,λ_0} and $A_{*,(\lambda_0,\Sigma_0)}$ are finite. Then, we need to construct an $\epsilon - \zeta$ proof for continuity; i.e., for every $\epsilon > 0$, there exists $\zeta_\epsilon > 0$ such that $\mathcal{D}_{\mathcal{L}}\left(\mathcal{P}, \mathcal{P}'\right) < \zeta_\epsilon$ implies $|\mathscr{L}(\mathcal{P}(A), F_*) - \mathscr{L}(\mathcal{P}'(A'), F_*)| < \epsilon$, where $\mathcal{P}$ and $\mathcal{P}'$ have their respective subscripts stipulated in parts (a), (b) and (c) of the Lemma. However, note that $\mathcal{P}$ and $\mathcal{P}'$ are *Voronoi partitions* that are associated with specific sets of

centroids A and A'. That is, based on $\mathcal{P}$ and $\mathcal{P}'$, we could compute both $\mathcal{D}_{\mathcal{L}}(\mathcal{P}, \mathcal{P}')$ and $\mathcal{D}_{\mathcal{H}}(A, A')$, and that we actually compute $|\mathscr{L}(A, F_*) - \mathscr{L}(A', F_*)|$.

To understand the following arguments, we require readers to understand Section 3.1 of Leonardi and Tamanini (2002). Now, for part (a) of the Lemma, it is clear that two partitions $\mathcal{P}_K$ and $\mathcal{P}'_K$ are "close" to each other (i.e. $\mathcal{D}_{\mathcal{L}}(\mathcal{P}_K, \mathcal{P}'_K)$ is small) only if every cluster V_k in $\mathcal{P}_K$ largely overlaps its counterpart V'_k in $\mathcal{P}'_K$ (the notion of "counterpart" makes sense here, because Leonardi and Tamanini (2002)'s metric $\mathcal{D}_{\mathcal{L}}$ actually considers the minimum of the Lebesgue measure of non-overlapping regions for every permutation of the clusters in the two partitions). High degree of overlapping occurs when the cluster centroid a_k of V_k is close to the centroid a'_k of V'_k for all $k = 1, \cdots, K$; i.e., $\mathcal{D}_{\mathcal{H}}(A_K, A'_K)$ is small when $\mathcal{D}_{\mathcal{L}}(\mathcal{P}_K, \mathcal{P}'_K)$ is small. Hence, there is only an additional layer to be inserted in the $\epsilon - \zeta$ proof: for every $\epsilon > 0$, pick partition $\mathcal{P}'_K$ that has $\mathcal{D}_{\mathcal{L}}(\mathcal{P}_K, \mathcal{P}'_K) < \zeta_{\mathcal{P}}(\zeta_\epsilon)$ such that their corresponding sets of centroids have $\mathcal{D}_{\mathcal{H}}(A_K, A'_K) < \zeta_\epsilon$, then plug in Pollard (1981)'s proof for continuity to ensure that $|\mathscr{L}(\mathcal{P}_K(A_K), F_*) - \mathscr{L}(\mathcal{P}'_K(A'_K), F_*)| = |\mathscr{L}(A_K, F_*) - \mathscr{L}(A'_K, F_*)| < \epsilon$.

For part (b), we can deploy a similar argument in the proof for part (b) of Lemma 4.2 to restrict our consideration to partitions $\mathcal{P}_{\lambda_0}$ and $\mathcal{P}'_{\lambda_0}$ with the same number of clusters. Then, the result for part (b) immediately follows from part (a). Part (c) is also the same, except that now Mahalanobis distance is involved. □

Proof of Theorem 5.2. We first invoke Lemma 4.3 to establish that $\mathscr{L}(\mathcal{P}(A), F_*)$ is a continuous function of $\mathcal{P}$, where $\mathcal{P}$ stands for the *Voronoi paritions* $\mathcal{P}_K$, $\mathcal{P}_{\lambda_0}$ and $\mathcal{P}_{(\lambda_0,\Sigma_0)}$ that are associated with A_K, A_{λ_0} and $A_{(\lambda_0,\Sigma_0)}$ in (4.10), (4.11) and (4.12) respectively. Then, by noticing that $\mathcal{P}_K$, $\mathcal{P}_{\lambda_0}$ and $\mathcal{P}_{(\lambda_0,\Sigma_0)}$ are deterministic functionals of F_w, whereas $\mathcal{P}_{*,K}$, $\mathcal{P}_{*,\lambda_0}$ and $\mathcal{P}_{*,(\lambda_0,\Sigma_0)}$ are deterministic functionals of F_*, the convergence of the posterior distribution Π_n of $\mathcal{P}_K$, $\mathcal{P}_{\lambda_0}$ and $\mathcal{P}_{(\lambda_0,\Sigma_0)}$ follows immediately from (4.7). □

Finally, we present the proof for Theorem 5.3.

Proof of Theorem 5.3 (Asymptotic Normality). We first consider the cases for RW K-means, RW DP-means and RW DP-rich. Notice that

$$\begin{aligned}\sqrt{n_k}\left(\mu^w_{n,k} - \hat{\mu}_{n,k}\right) &= \frac{\sqrt{n_k}}{\sum_{i\in\mathcal{C}^0_k} W_i}\left[\sum_{i\in\mathcal{C}^0_k} W_i y_i - \left(\sum_{i\in\mathcal{C}^0_k} W_i\right)\hat{\mu}_{n,k}\right] \\ &= \frac{1}{\overline{W}_{n,k}} \cdot \frac{1}{\sqrt{n_k}} \sum_{i\in\mathcal{C}^0_k} W_i(y_i - \hat{\mu}_{n,k})\end{aligned}$$

where $\overline{W}_{n,k} = \frac{1}{n_k}\sum_{i\in\mathcal{C}^0_k} W_i \xrightarrow{a.s.} 1$ as $n_k \to \infty$. Then, for every $z \in \mathbb{R}^d$,

$$z'\left[\sqrt{n_k}\left(\mu^w_{n,k} - \hat{\mu}_{n,k}\right)\right] = \frac{1}{\overline{W}_{n,k}} \cdot \frac{1}{\sqrt{n_k}} \sum_{i\in\mathcal{C}^0_k} W_i\left[z'(y_i - \hat{\mu}_{n,k})\right]$$

$$= \frac{1}{\overline{W}_{n,k}} \cdot \sqrt{\frac{1}{n_k} \sum_{i \in \mathcal{C}_k^0} \left[z'(y_i - \hat{\mu}_{n,k})\right]^2} \cdot \frac{\sum_{i \in \mathcal{C}_k^0} \left[z'(y_i - \hat{\mu}_{n,k})\right] W_i}{\sqrt{\sum_{i \in \mathcal{C}_k^0} \left[z'(y_i - \hat{\mu}_{n,k})\right]^2}},$$

where

$$\begin{aligned} \frac{1}{n_k} \sum_{i \in \mathcal{C}_k^0} \left[z'(y_i - \hat{\mu}_{n,k})\right]^2 &= z' \left(\frac{1}{n_k} \sum_{i \in \mathcal{C}_k^0} [y_i - \hat{\mu}_{n,k}][y_i - \hat{\mu}_{n,k}]' \right) z \\ &\to z' \left(V_{*,k}^{\mathcal{P}_0} \right) z \quad a.s. \quad P_{F_*}^{(\infty)}, \end{aligned}$$

and

$$\begin{aligned} \mathbb{E}\left(\sum_{i \in \mathcal{C}_k^0} \left[z'(y_i - \hat{\mu}_{n,k})\right] W_i \middle| y \right) &= \sum_{i \in \mathcal{C}_k^0} \left[z'(y_i - \hat{\mu}_{n,k})\right] = 0, \\ Var\left(\sum_{i \in \mathcal{C}_k^0} \left[z'(y_i - \hat{\mu}_{n,k})\right] W_i \middle| y \right) &= \sum_{i \in \mathcal{C}_k^0} \left[z'(y_i - \hat{\mu}_{n,k})\right]^2 = \mathcal{O}(n), \\ \mathbb{E}\left(\sum_{i \in \mathcal{C}_k^0} \left[z'(y_i - \hat{\mu}_{n,k})\right]^4 W_i^4 \middle| y \right) &= \sum_{i \in \mathcal{C}_k^0} \left[z'(y_i - \hat{\mu}_{n,k})\right]^4 \mathbb{E}\left(W_i^4\right) = \mathcal{O}(n), \end{aligned}$$

where the last two lines follow from the finite (second and) fourth moment(s) property of the standard-Exponentially-distributed random weights. Hence, the Liapounov's sufficient condition is satisfied to apply the Lindeberg's Central Limit Theorem (coupled with Slutsky's Theorem) to obtain

$$P\left(z' \left[\sqrt{n_k}\left(\mu_{n,k}^w - \hat{\mu}_{n,k}\right)\right] \in B \middle| y\right) \to P(Z_1 \in B) \quad a.s.\ P_{F_*}^{(\infty)}$$

for any Borel set $B \subset \mathbb{R}$ and $Z_1 \sim N\left(0, z'\left(V_{*,k}^{\mathcal{P}_0}\right) z\right)$. Finally, apply the Cramer-Wold device to obtain the result.

For RW SDP-means and RW SDP-rich, notice that

$$\begin{aligned} \sqrt{n_k}\left(\mu_{n,k}^w - \hat{\mu}_{n,k}\right) &= \frac{\sqrt{n_k}}{\sum_{i \in \mathcal{C}_k^0} W_i + \xi_0} \left[\sum_{i \in \mathcal{C}_k^0} W_i y_i - \left(\sum_{i \in \mathcal{C}_k^0} W_i + \xi_0 \right) \hat{\mu}_{n,k} + \xi_0 \mu_0 \right] \\ &= \frac{\sum_{i \in \mathcal{C}_k^0} W_i}{\sum_{i \in \mathcal{C}_k^0} W_i + \xi_0} \cdot \frac{1}{\overline{W}_{n,k}} \cdot \frac{1}{\sqrt{n_k}} \left[\sum_{i \in \mathcal{C}_k^0} W_i (y_i - \hat{\mu}_{n,k}) + \xi_0 \left(\mu_0 - \hat{\mu}_{n,k}\right) \right]. \end{aligned}$$

The first term converges to one almost surely as n_k increases, whereas the extra term

$$\frac{\xi_0 \left(\mu_0 - \hat{\mu}_{n,k}\right)}{\sqrt{n_k}} \to 0 \quad a.s. \quad P_{F_*}^{(\infty)}.$$

The rest of the terms are the same from before, and the result follows from applying Slutsky's theorem to deal with these extra terms. □

5 Additional information for numerical experiments

5.1 Additional comparison for simulations

We consider an additional comparison criterion for our simulation results: **NMI w.r.t. ground truth cluster labels**. Specifically, we compute the NMI (Vinh et al., 2010) value that compares the b^{th} posterior draw of cluster assignments and the ground-truth cluster labels for the t^{th} simulated training data set

$$\eta_{(\cdot)}^{(b,t)} := \text{NMI}\left(\boldsymbol{z}_{(\cdot)}^{(b,t)}\ ,\ \boldsymbol{z}_{(\text{truth})}^{(t)}\right),$$

and then plot the boxplots for the mean of these NMI's from each of these 6 methods for $t = 1, \cdots, T$ datasets

$$\bar{\eta}_{(\cdot)}^{(t)} = \frac{1}{B}\sum_{b=1}^{B} \eta_{(\cdot)}^{(b,t)}. \tag{5.1}$$

Basically, we want to compare how well these methods in "recovering" the true cluster partition (one for perfect recovery of true partition, and zero otherwise). We note that this comparison criterion is popular in existing classification and/or clustering literature.

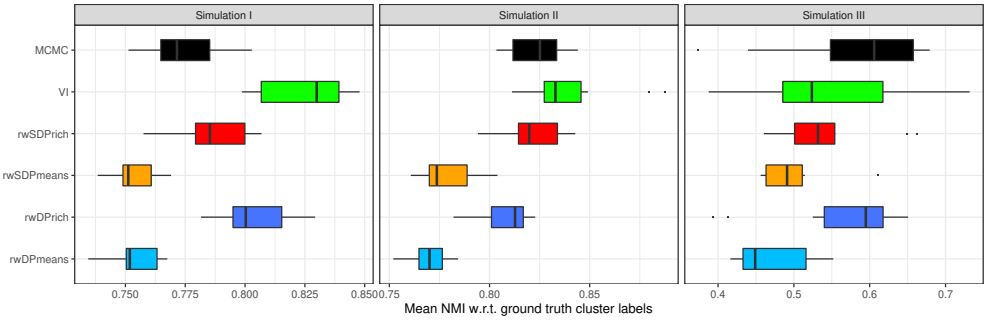

Figure 2: Sampling distribution of average NMI $\eta_{(\cdot)}^{(b,t)}$ in comparison with ground-truth cluster assignments (Equation (5.1)) among $T = 10$ simulated data sets in 3 simulation settings for each of the 6 methods: MCMC, VI and the 4 random-weighting setups.

Results. Overall the RW DP-rich and RW SDP-rich have average NMI values that are comparable to those of MCMC and VI, and they are also higher than those of RW DP-means and RW SDP-means. This could be attributable to the presence of *rgr* regularization in the RW DP-rich and RW SDP-rich setups.

5.2 Specifying priors for benchmark and motivating data examples

Briefly, the priors for these data sets are specified with an Empirical Bayes approach (i.e., priors are estimated using some information from the data itself), and then the

same set of priors are adopted for MCMC, VI and RW SDP-rich (where applicable) to facilitate meaningful comparison among all these methods.

Benchmark data examples

For `iris` data set, the full-covariance structure is adopted for MCMC, VI and RW SDP-rich. Based on the ground-truth cluster labels, we find their corresponding cluster-specific centroids and covariances. Let $\tilde{\mu}_0$ and $\tilde{\Sigma}_0$ be the corresponding weighted (by true mixing proportion) average of these centroids and covariances. Then, we specify $\mu_0 = \tilde{\mu}_0$, $\nu_0 = d + 3 = 7$, and $\psi_0 = 2 \times \tilde{\Sigma}_0$, such that the inverse-Wishart prior mean of Σ is equal to $\tilde{\Sigma}_0$ whereas the prior variance of Σ is huge. ξ_0 is estimated to be the average of element-wise ratios between the diagonals of $\tilde{\Sigma}_0$ and the diagonals of the covariance of all data. Finally, α_0 is fixed at 0.4 such that the CRP prior mean of κ (e.g., Teh, 2010) is equal to $K_{\text{true}} = 3$.

Similar method of prior specification is also used for the `wine` data set, except that we adopt the diagonal covariance structure here, since we are analyzing the transformed data set via PCA. Again, let $\tilde{\mu}_0$ and $\left\{\tilde{\sigma}^2_{0,j}\right\}_{1 \leq j \leq d}$ be the corresponding weighted (by true mixing proportion) average of the cluster-specific centroids and variances for each dimension $j = 1, \cdots, d$. Then, we fix $\mu_0 = \tilde{\mu}_0$, $a_{0,j} = 2$, and $b_{0,j} = 2 \times \tilde{\sigma}^2_{0,j}$ for all $1 \leq j \leq d$, such that the inverse-Gamma prior mean of σ^2_j is equal to $\tilde{\sigma}^2_{0,j}$ whereas the prior variance of σ^2_j is huge. Similarly, for each dimension j, $\xi_{0,j}$ is taken to be ratio between $\tilde{\sigma}^2_{0,j}$ and variance of $\{y_{ij}\}_{1 \leq i \leq n}$. Again, we also fix $\alpha_0 = 0.4$ to equate the CRP prior mean of κ to $K_{\text{true}} = 3$.

TCR Data Example

We first perform an agglomerative hierarchical clustering (HC) with average linkage on the 3-dimensional TCR data set ($n = 13387$ TCR sequences) to obtain a solution path of partitions starting from singleton/atomic clusters to all observations lumped together in a degenerate cluster.

Next, we need to determine a suitable cutoff on the HC dendrogram which would give us a corresponding partition to help us specify our priors. Now, we could obtain Shannon's entropy for each partition along the hierarchical clustering (HC) solution path. Intuitively, a "good" partition is one that clusters homogeneous observations together and separates non-homogeneous ones apart, which leads to a significant drop in Shannon's entropy. Consequently, for each partition along the HC solution path, we repeatedly permute the cluster labels and recalculate the corresponding (permuted) entropy. By keeping track of the percentage of permuted entropies that are smaller (more extreme) than the observed entropies, we are able to obtain a series of permutation p-values associated with the HC solution path. This permutation exercise reveals that the partition consisting of 1477 clusters is the finest partition we could get before the permutation p-values rise up sharply, i.e. subsequent agglomeration of the clusters no longer leads to any significant drop in Shannon's entropy.

Based on these 1477 clusters obtained from hierarchical clustering, we specify our priors for the DPM under the diagonal-covariance structure: we fix $\alpha_0 = 420$ so that the CRP prior mean of κ is approximately 1470, and that the VI stick-breaking threshold is fixed at $K_{\max} = 2000$. Subsequently, similar to our preceding benchmark data analyses, based on these 1477 HC clusters, we compute the weighted (by the mixing proportion as indicated by the 1477 HC clusters) average $\tilde{\mu}_0$ of the cluster centroids, weighted variance (denoted with $\breve{\mu}_0$) of the cluster centroids, weighted average $\tilde{\sigma}^2_{0,j}$ of the cluster variances as well as the weighted variance (denoted with $\breve{\sigma}^2_{0,j}$) of cluster variances for $j = 1, \cdots, d$. Finally, we specify the priors $\mu_{0,j}$, $\xi_{0,j}$, $a_{0,j}$ and $b_{0,j}$ via method-of-moments:

$$\begin{aligned}
\mu_{0,j} &= \tilde{\mu}_{0,j}, \\
\xi_{0,j} &= \frac{\tilde{\sigma}^2_{0,j}}{\breve{\mu}_{0,j}}, \\
a_{0,j} &= \frac{\left[\tilde{\sigma}^2_{0,j}\right]^2}{\breve{\sigma}^2_{0,j}} + 2, \\
b_{0,j} &= \tilde{\sigma}^2_{0,j} \times (a_{0,j} - 1)\,.
\end{aligned}$$

5.3 Additional plots and tables

This subsection serves as a placeholder for additional plots and tables for the numerical experiments in this paper.

Computational times for the numerical experiments are tabulated in Table 1 for comparison.

Methods	`Simul I`	`Simul II`	`Simul III`	`Iris` Data	`Wine` Data	`TCR` Data
MCMC						∼10 days
RW DP-means						
RW DP-rich						
RW SDP-means						
RW SDP-rich						
VI						

Table 1: Computational times for various methods in our numerical experiments. Simulation is abbreviated as `Simul`.

Meanwhile, Figure 3 shows the trace plots for posterior number of clusters obtained by MCMC for all benchmark and motivating data sets. In particular, the trace plot corresponding to the MCMC scheme deployed on the original `wine` data set (original number of features $d = 13$) using R package `DPpackage` shows poor mixing of the MCMC chain. This MCMC implementation involves a full covariance structure where the priors are specified using similar approach that we described in the preceding subsection. A closer inspection of the covariance among the features reveals a highly-correlated data structure, which justifies our approach of first transforming the data set with PCA and use only the first 5 principal components which explain more than 80% of the variation in the data as illustrated in Figure 4. Subsequent MCMC implementation based on this transformed data set demonstrates reasonable mixing of the MCMC chain.

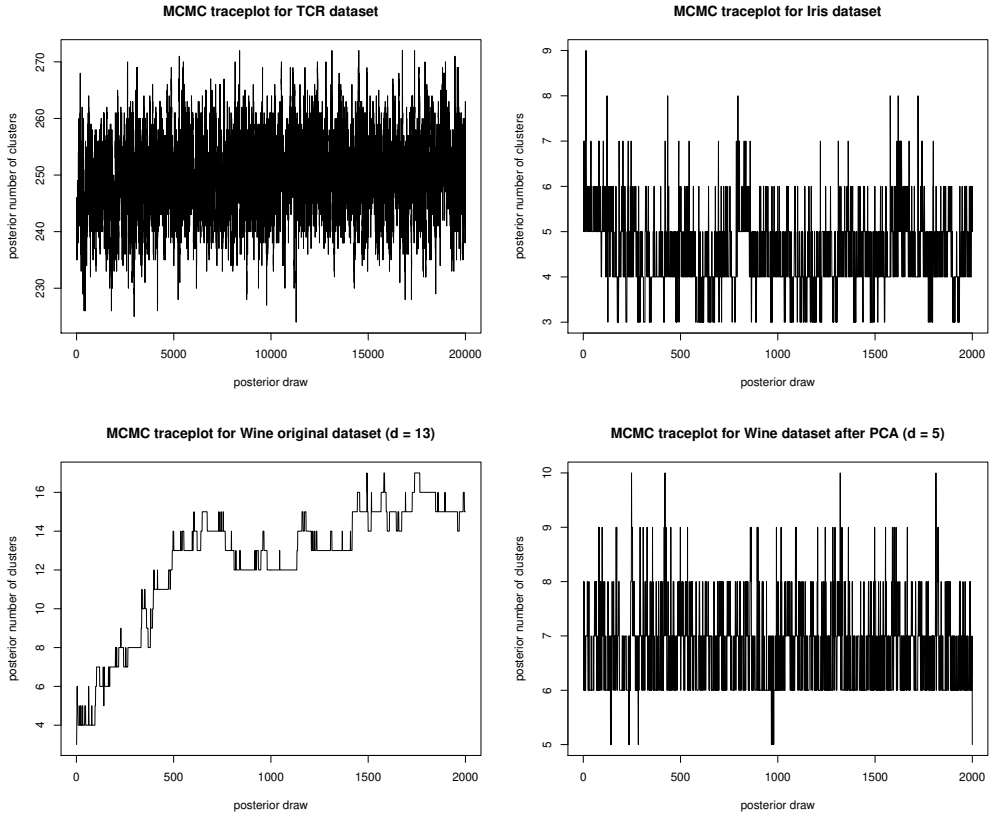

Figure 3: Trace plots for posterior number of clusters obtained by MCMC for all benchmark and motivating data sets.

6 Variational Inference

The DPM of Normals (with Normal-inverse-Wishart prior) can be expressed with a stick-breaking prior (Sethuraman, 1994):

$$\begin{aligned} y_i|(z_i = k, \mu_k, \Sigma_k) &\sim N_d(\mu_k, \Sigma_k) \\ \mu_k\big|\Sigma_k &\sim N_d\left(\mu_0, h(\Sigma_k)\right) \\ \Sigma_k &\sim p(\Sigma_k) \\ z_i|\pi(\boldsymbol{v}) &\sim Mult\left(\pi(\boldsymbol{v})\right) \\ \pi(\boldsymbol{v})|\alpha_0 \sim GEM(\alpha_0) \Longleftrightarrow \pi_k &= v_k \prod_{l=1}^{k-1}(1 - v_l) \quad \text{for} \quad v_k \sim Beta(1, \alpha_0). \end{aligned} \tag{6.1}$$

Note that in the main text, we have considered a DPM working model which shares a common mixture-component covariance term. However, since the MCMC schemes

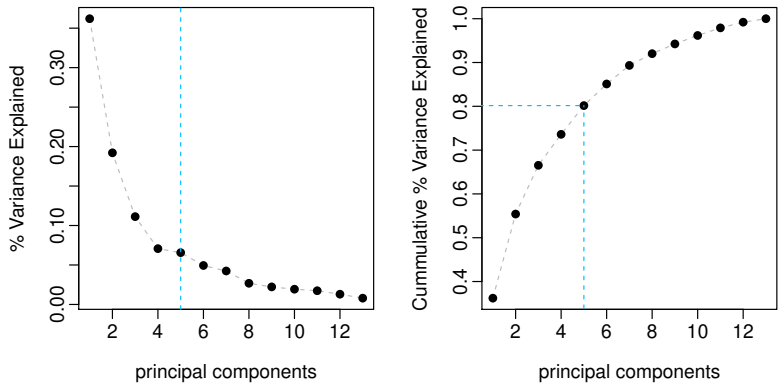

Figure 4: PCA scree plots for `wine` data set depicting percentage of variance explained in the data across the principal components. The blue dashed lines represent linear mapping of the original data set to a 5-dimensional subspace. The gray dashed lines only serve as interpolation between the points to ease visual inspection.

deployed in the numerical experiments adopt the more general form of DPM that allows cluster-specific covariance terms, we also adopt the same generalized DPM model for variational inference (VI) so that the results obtained by VI are more closely aligned to the MCMC samples.

6.1 Full-covariance structure

Under the full-covariance structure, $h(\Sigma_k) = {}^{\Sigma_k}/_{\xi_0}$, and here we specify a Wishart prior on the precision

$$\Sigma_k^{-1} \sim Wishart_d(\nu_0, \psi_0) \tag{6.2}$$

Note that the parameterization used for Wishart distribution is such that $\mathbb{E}\left(\Sigma_k^{-1}\right) = \nu_0\psi_0$. Applying the mean-field variational inference at a truncation level $K_{\max}$, we approximate

$$p\left(\mu, \Sigma, z, \boldsymbol{v} \middle| \boldsymbol{y}\right)$$

with

$$\prod_{k=1}^{K_{\max}} \left[q\left(\mu_k \middle| \Sigma_k\right) q\left(\Sigma_k\right) q(v_k)\right] \times \prod_{i=1}^{n} q(z_i),$$

where the variational densities q are specified as below:

$$\begin{aligned} \mu_k \big| \Sigma_k &\sim N_d\left(\hat{\mu}_k, \frac{1}{\hat{\xi}_k}\Sigma_k\right) \\ \Sigma_k^{-1} &\sim Wishart_d(\hat{\nu}_k, \widehat{\psi}_k) \\ z_i &\sim Mult\left(\hat{\boldsymbol{\pi}}_i\right) \\ v_k &\sim Beta\left(\hat{\alpha}_{k1}, \hat{\alpha}_{k2}\right). \end{aligned} \tag{6.3}$$

We need to solve for

$$\{\hat{\alpha}_{k1}, \hat{\alpha}_{k2}\}_{1 \leq k \leq K_{\max}} \quad \text{and} \quad \left\{\hat{\mu}_k, \hat{\xi}_k, \hat{\nu}_k, \widehat{\psi}_k\right\}_{1 \leq k \leq K_{\max}} \quad \text{and} \quad \{\hat{\pi}_{ik}\}_{1 \leq i \leq n, 1 \leq k \leq K_{\max}}.$$

Using techniques outlined in Section 4.1 of Nakajima et al. (2019), we obtain

$$\begin{aligned}
\hat{\mu}_k &= \frac{\xi_0 \mu_0 + \sum_{i=1}^{n} \hat{\pi}_{ik} y_i}{\xi_0 + \sum_{i=1}^{n} \hat{\pi}_{ik}} \\
\hat{\xi}_{kj} &= \xi_{0j} + \sum_{i=1}^{n} \hat{\pi}_{ik} \\
\hat{\nu}_k &= \nu_0 + \sum_{i=1}^{n} \hat{\pi}_{ik} \\
\widehat{\psi}_k^{-1} &= \sum_{i=1}^{n} \hat{\pi}_{ik} y_i y_i' + \xi_0 \mu_0 \mu_0' - \hat{\xi}_k \hat{\mu}_k \hat{\mu}_k' + \psi_0^{-1} \\
\hat{\alpha}_{k1} &= 1 + \sum_{i=1}^{n} \hat{\pi}_{ik} \\
\hat{\alpha}_{k2} &= \alpha_0 + \sum_{l=k+1}^{K_{\max}} \sum_{i=1}^{n} \hat{\pi}_{il},
\end{aligned} \tag{6.4}$$

and

$$\begin{aligned}
\bar{\pi}_{ik} = \exp \Bigg\{ & \left[\Psi\left(\hat{\alpha}_{k1}\right) - \Psi\left(\hat{\alpha}_{k1} + \hat{\alpha}_{k2}\right)\right] + \sum_{l=1}^{k-1} \left[\Psi\left(\hat{\alpha}_{k2}\right) - \Psi\left(\hat{\alpha}_{k1} + \hat{\alpha}_{k2}\right)\right] \\
& + \frac{1}{2} \left[\sum_{j=1}^{d} \Psi\left(\frac{\hat{\nu}_k + 1 - j}{2}\right) + d \log 2 + \log \left|\widehat{\psi}_k\right| - \frac{d}{\hat{\xi}_k} - \hat{\nu}_k (y_i - \hat{\mu}_k)' \widehat{\psi}_k (y_i - \hat{\mu}_k) \right] \Bigg\}
\end{aligned} \tag{6.5}$$

where $\Psi(\cdot)$ denotes the digamma function, such that

$$\hat{\pi}_{ik} = \frac{\bar{\pi}_{ik}}{\sum_{l=1}^{K_{\max}} \bar{\pi}_{il}}. \tag{6.6}$$

The coordinate ascent algorithm is used to iteratively update the parameters of the variational distributions until the **evidence lower bound (ELBO)** converges:

$$\begin{aligned}
\mathcal{L}_q = & - \sum_{k=1}^{K_{\max}} \sum_{i=1}^{n} \hat{\pi}_{ik} \log \hat{\pi}_{ik} - \sum_{k=1}^{K_{\max}} \log \frac{\Gamma\left(\hat{\alpha}_{k1} + \hat{\alpha}_{k2}\right)}{\Gamma\left(\hat{\alpha}_{k1}\right) \Gamma\left(\hat{\alpha}_{k2}\right)} \\
& + \sum_{k=1}^{K_{\max}} \sum_{j=1}^{d} \log \Gamma\left(\frac{\hat{\nu}_k + 1 - j}{2}\right) + \frac{1}{2} \sum_{k=1}^{K_{\max}} \hat{\nu}_k \log \left|\widehat{\psi}_k\right| + \frac{d \log 2}{2} \sum_{k=1}^{K_{\max}} \hat{\nu}_k - \frac{d}{2} \sum_{k=1}^{K_{\max}} \log\left(\hat{\xi}_k\right).
\end{aligned} \tag{6.7}$$

The predictive distribution $p(y_{n+1}|\boldsymbol{y}_{1:n})$ is approximated by

$$\sum_{k=1}^{K_{\max}} \left\{ \frac{\hat{\alpha}_{k1}}{\hat{\alpha}_{k1}+\hat{\alpha}_{k2}} \times \prod_{j=1}^{k-1} \frac{\hat{\alpha}_{j2}}{\hat{\alpha}_{j1}+\hat{\alpha}_{j2}} \right. \\ \left. \times \pi^{-d/2} \times \left(\frac{\hat{\xi}_k}{1+\hat{\xi}_k}\right)^{\frac{d}{2}} \times \frac{\Gamma\left(\frac{\hat{\nu}_k+1}{2}\right)}{\Gamma\left(\frac{\hat{\nu}_k+1-d}{2}\right)} \times \frac{\left|\widehat{\psi}_k^{-1}\right|^{\frac{\hat{\nu}_k}{2}}}{\left|\widehat{\psi}_{new,k}^{-1}\right|^{\frac{\hat{\nu}_k+1}{2}}} \right\}, \tag{6.8}$$

where

$$\widehat{\psi}_{new,k}^{-1} = \widehat{\psi}_k^{-1} + \frac{\hat{\xi}_k}{1+\hat{\xi}_k}(y_{n+1}-\hat{\mu}_k)(y_{n+1}-\hat{\mu}_k)'.$$

6.2 Diagonal-covariance structure

For diagonal-covariance structure, $\Sigma_k = diag\left(\sigma_{k1}^2, \cdots, \sigma_{kd}^2\right)$ in (6.1), and

$$h(\Sigma_k) = diag\left(\frac{\sigma_{k1}^2}{\xi_{0,1}}, \cdots, \frac{\sigma_{kd}^2}{\xi_{0,d}}\right).$$

Gamma priors are adopted for the precision terms, i.e. for $j = 1, \cdots, d$,

$$\frac{1}{\sigma_{kj}^2} \overset{ind}{\sim} Gamma(a_{0j}, b_{0j}).$$

Applying the mean-field variational inference at a truncation level $K_{\max}$, we approximate

$$p\left(\mu, \sigma^2, z, \boldsymbol{v}|\boldsymbol{y}\right)$$

with

$$\prod_{k=1}^{K_{\max}} \left\{ q(v_k) \prod_{j=1}^{d} \left[q\left(\mu_{kj}|\sigma_{kj}^2\right) q\left(\sigma_{kj}^2\right) \right] \right\} \times \prod_{i=1}^{n} q(z_i),$$

where the variational densities $q(z_i)$ and $q(v_k)$ are the same as their counterparts in (6.3), and the other variational densities for the component parameters are

$$\mu_{kj}|\sigma_{kj}^2 \sim N\left(\hat{\mu}_{kj}, \frac{1}{\hat{\xi}_{kj}}\sigma_{kj}^2\right)$$
$$\frac{1}{\sigma_{kj}^2} \sim Gamma(\hat{a}_{kj}, \hat{b}_{kj}).$$

We need to solve for

$$\{\hat{\alpha}_{k1}, \hat{\alpha}_{k2}\}_{1\le k\le K_{\max}} \quad \text{and} \quad \left\{\hat{\mu}_{kj}, \hat{\xi}_{kj}, \hat{a}_{kj}, \hat{b}_{kj}\right\}_{1\le k\le K_{\max}\ ;\ 1\le j\le d} \quad \text{and} \quad \{\hat{\pi}_{ik}\}_{1\le k\le K_{\max}\ ;\ 1\le i\le n}.$$

The solutions for $\left\{\hat{\alpha}_{k1}, \hat{\alpha}_{k2}, \hat{\mu}_{kj}, \hat{\xi}_{kj}\right\}$ are the same as their counterparts in (6.4), whereas

$$\hat{a}_{kj} = a_{0j} + \frac{1}{2}\sum_{i=1}^{n} \hat{\pi}_{ik}$$
$$\hat{b}_{kj} = b_{0j} + \frac{1}{2}\sum_{i=1}^{n} \hat{\pi}_{ik} y_{ij}^2 + \frac{1}{2}\xi_{0j}\mu_{0j}^2 - \frac{1}{2}\hat{\xi}_{kj}\hat{\mu}_{kj}^2.$$

We still use (6.6) to calculate $\hat{\pi}_{ik}$, but we need to modify the formula to calculate $\bar{\pi}_{ik}$ by replacing the second line of (6.5) inside the exponent with

$$+\frac{1}{2}\sum_{j=1}^{d}\left[\Psi(\hat{a}_{kj}) - \log\left(\hat{b}_{kj}\right) - \frac{1}{\hat{\xi}_{kj}} - \frac{\hat{a}_{kj}}{\hat{b}_{kj}}\left(y_{ij} - \hat{\mu}_{kj}\right)^2\right].$$

The formula for ELBO also needs to be modified by replacing the second line of (6.7) with

$$+\sum_{k=1}^{K_{\max}}\sum_{j=1}^{d}\log\Gamma\left(\hat{a}_{kj}\right) - \sum_{k=1}^{K_{\max}}\sum_{j=1}^{d}\hat{a}_{kj}\log\left(\hat{b}_{kj}\right) - \frac{1}{2}\sum_{k=1}^{K_{\max}}\sum_{j=1}^{d}\log\left(\hat{\xi}_{kj}\right).$$

Finally, the formula to approximate the predictive distribution should be modified by replacing the second line of (6.8) with

$$(2\pi)^{-d/2} \times \prod_{j=1}^{d}\left(\frac{\hat{\xi}_{kj}}{1+\hat{\xi}_{kj}}\right)^{\frac{1}{2}} \times \prod_{j=1}^{d}\frac{\Gamma\left(\hat{a}_{kj}+\frac{1}{2}\right)}{\Gamma\left(\hat{a}_{kj}\right)} \times \prod_{j=1}^{d}\frac{\hat{b}_{kj}^{\hat{a}_{kj}}}{\check{b}_{kj}^{\hat{a}_{kj}+\frac{1}{2}}},$$

where

$$\check{b}_{kj} = \hat{b}_{kj} + \frac{\hat{\xi}_{kj}}{1+\hat{\xi}_{kj}}\frac{\left(y_{n+1,j} - \hat{\mu}_{kj}\right)^2}{2}.$$

References

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

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*. 10

Ghosal, S. and van der Vaart, A. (2017). *Fundamentals of Nonparametric Bayesian Inference*. Cambridge University Press. 11, 12

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. 9

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, second edition. 9

Ishwaran, H. and Zarepour, M. (2002). "Exact and approximate sum representations for the Dirichlet process." *Canadian Journal of Statistics*, 30(2): 269–283. 11

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*. 2, 4, 8

Leonardi, G. P. and Tamanini, I. (2002). "Metric spaces of partitions, and Caccioppoli partitions." *Advances in Mathematical Sciences and Applications*, 12(2): 725–753. 15

Lyddon, S., Holmes, C., and Walker, S. (2019). "General Bayesian updating and the loss-likelihood bootstrap." *Biometrika*, 106(2): 465–478. 10

Mason, D. M. and Newton, M. A. (1992). "A rank statistics approach to the consistency of a general bootstrap." *The Annals of Statistics*, 20(3): 1611–1624. 10

Muliere, P. and Secchi, P. (1996). "Bayesian nonparametric predictive inference and bootstrap techniques." *Annals of the Institute of Statistical Mathematics*, 48(4): 663–673. 11

Nakajima, S., Watanabe, K., and Sugiyama, M. (2019). *Variational Bayesian Learning Theory*. Cambridge University Press, first edition. 22

Ng, T. L. and Newton, M. A. (2020). "Random Weighting in LASSO Regression." *arXiv: 2002.02629*. In revision at the Electronic Journal of Statistics. 10

Paul, D. and Das, S. (2020). "A Bayesian non-parametric approach for automatic clustering with feature weighting." *Stat*, 9(1). 2

Pollard, D. (1981). "Strong consistency of K-means clustering." *The Annals of Statistics*, 9(1): 135–140. 14, 15

Raykov, Y. P., Boukouvalas, A., and Little, M. A. (2016). "Simple approximate MAP inference for Dirichlet processes mixtures." *Electronic Journal of Statistics*, 10: 3548–3578. 2, 8

Rubin, D. B. (1981). "The Bayesian Bootstrap." *The Annals of Statistics*, 9(1): 130–134. 11

Sethuraman, J. (1994). "A constructive definition of dirichlet priors." *Statistica Sinica*, 4: 639–650. 11, 20

Teh, Y. W. (2010). *Dirichlet Process*. University College London. https://www.stats.ox.ac.uk/~teh/research/npbayes/Teh2010a.pdf. 18

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. 17