Supplementary Materials for Spectral Clustering, Bayesian Spanning Forest, and Forest Process

S 1 Model-based Extensions to Forest Model

S 1.1 Extension to High-dimensional Clustering Model

For clustering high dimensional data, good performances have been demonstrated through finding a low-dimensional sparse representation z_i for each y_i (Vidal, 2011; Wu et al., 2014), and then clustering z_i instead of y_i. To briefly review the idea, for high-dimensional data, a useful assumption is that $y_i \in \mathbb{R}^p$ can be "reconstructed" using a linear combination of a few other y_j's, that is, $y_i \approx \sum_j w_{i,j} y_j$, with $w_{i,i} = 0$ and $w_i = (w_{i,1}, \ldots, w_{i,n})$ contains only a few non-zeros.

Although w_i is obtained as a vector of coefficients, it can be viewed as a low-dimensional *relative* coordinate, that can be used instead of the absolute coordinate $y_i \in \mathbb{R}^p$. The key idea is that if w_i and w_j are in different subspaces ($w_i' w_j = 0$), then y_i and y_j are likely to be in different clusters. Using a similarity function defined on each pair (w_i, w_j), one could obtain a similarity matrix and then apply the spectral clustering algorithm.

We now propose a generative distribution. We use $W = [w_1', \ldots, w_n']$ as the $n \times n$ matrix with the ith row equal to w_i, and Y the $n \times p$ data matrix. We include the reconstruction loss $\|Y - WY\|_F^2$ (with $\|.\|_F$ the Frobenius norm) via a matrix Gaussian distribution:

$$Y \sim \text{Matrix-Gaussian}\big(O, \sigma_y^2[(I_n - W)'(I_n - W)]^{-1}, I_p\big).$$

We note a link between this model and the spatial autoregressive (SAR) model (Ord, 1975), except that the neighborhood information W is not known. We view each w_i as a

transform of another unit-norm vector z_i that satisfies $\|z_i\|_2 = 1$ and $\|z_i\|_0 = d$ (the number of non-zeros is d) via

$$w_{i,k} = \alpha_i z_{i,k}, \text{ for } k \neq i, \quad w_{i,i} = 0, \quad z_{i,i} \in \mathbb{R},$$

with $\alpha_i > 0$ some scale parameter, and $z_{i,i}$ not necessarily zero. And we model $(z_1, \ldots, z_n)$ as from a forest model based on sparse von Mises–Fisher densities:

$$\begin{aligned}
&z_1, \ldots, z_n \sim \text{Forest Model}(\mathcal{T}),\\
&f(z_i \mid z_j; \kappa) \propto \exp(\kappa z_i' z_j) 1(z_i' z_j \neq 0) 1(\|z_i\|_2 = 1, \|z_i\|_0 = d),\\
&r(z_i) \propto 1(\|z_i\|_2 = 1, \|z_i\|_0 = d).
\end{aligned}$$

The leaf $f(z_i \mid z_j; \kappa)$ is supported in those $(d-1)$-dimensional unit spheres $\mathcal{S}^{(d-1)} \subset \mathcal{S}^{(n-1)}$, such that z_i and z_j are not in completely disjoint subspaces. The von Mises–Fisher density in a given $\mathcal{S}^{(d-1)}$ has a tractable normalizing constant that depends on κ only. Further, with $\|z_j\|_0 = d$, we can easily tell the number of those $\mathcal{S}^{(d-1)}$ with $z_i : z_i' z_j \neq 0$ is equal to $\binom{n}{d} - \binom{n-d}{d}$. Similarly, r is a uniform density on all $\mathcal{S}^{(d-1)} \subset \mathcal{S}^{(n-1)}$. Therefore, both of the normalizing constants in f and r are available. We refer to the model for Y as a latent forest model.

One could further assign priors on κ, σ_y^2 and α_i's, and develop a Gibbs sampling algorithm for posterior estimation. In this section, since our main focus is to demonstrate a high-dimensional model extension and compare the point estimates against a few other algorithms, we use a fast posterior approximation algorithm for the above model. Specifically, we first use the lasso algorithm to solve for a sparse $\hat{W} = \arg\min_{W: w_{i,i}=0 \; \forall i} (1/2)\|Y - WY\|_F^2 + \lambda \|W\|_1$ with $\lambda = 1$; then for each $\hat{w}_i$, we take the top $(d-1)$ elements in magnitude, and set the other elements to zero. Then we replace $w_{i,i}$ by 1 and normalize the vector to produce a unit 2-norm vector z_i. Conditioning on the transformed matrix $\hat{Z}$ and κ fixed to 10, we sample the forest $\mathcal{T}$ using the random-walk covering algorithm.

To assess the clustering performance, we use the image data from the Yale face database B (Georghiades et al., 2001). This dataset contains single light source images of 10 subjects. We take the ones corresponding to the forward-facing poses under 64 different illumination conditions (shown in Figure S.1). We resize each image to have 48×42 pixels. We label those images by subject id from 1 to 10. Therefore, we have a clustering task with $n = 640$ and $p = 2,016$.

(a) One subject under illumination condition 1.

(b) One subject under illumination condition 2.

(c) One subject under illumination condition 3.

(d) Another subject under illumination condition 1.

(e) Another subject under illumination condition 2.

(f) Another subject under illumination condition 3.

Figure S.1: a few sample photos from the Yale face database B (Georghiades et al., 2001).

We compare the performance against several popular clustering methods. To produce a point estimate, for the forest model on z_i's, we apply spectral clustering with $K = 10$ on

the posterior co-assignment probability matrix (as described in the main text); for each of the other methods, we use $K = 10$ as the specified parameter. To evaluate the clustering accuracy, we relabel the point estimate $(c_1, \ldots, c_n)$ using the Hungarian matching algorithm (Kuhn, 1955), so that the Hamming distance dist_h between $(c_1, \ldots, c_n)$ and the subject id's is minimized. Then the clustering accuracy is calculated as $(n - \text{dist}_\text{h})/n$. As the accuracy can be sensitive to the initialization of each algorithm, for a fair comparison, we repeat running each algorithm 20 times, and report the mean and the 95% confidence interval.

Method	K-means	Mclust (VII)	Mclust (VEI)	Mclust (EII)
Accuracy	0.18 (0.16, 0.21)	0.24 (0.24, 0.24)	0.26 (0.26, 0.26)	0.23 (0.23, 0.23)
Method	HDDC (AkjBkQkDk)	HDDC (AkjBQkDk)	SpecC on $e^{-\lambda_s \|y_i - y_j\|_2^2}$	SpecC on $y_i' y_j$
Accuracy	0.324	0.296	0.35 (0.25, 0.43)	0.30 (0.28, 0.34)
Method	K-means on w_i	SpecC on $(w_i' w_j)_+$	SpecC on $\|w_{i,j}\| + \|w_{j,i}\|$	Forest on z_i
Accuracy	0.25 (0.18, 0.39)	0.64 (0.52, 0.69)	0.59 (0.46, 0.68)	0.82 (0.71, 0.93)

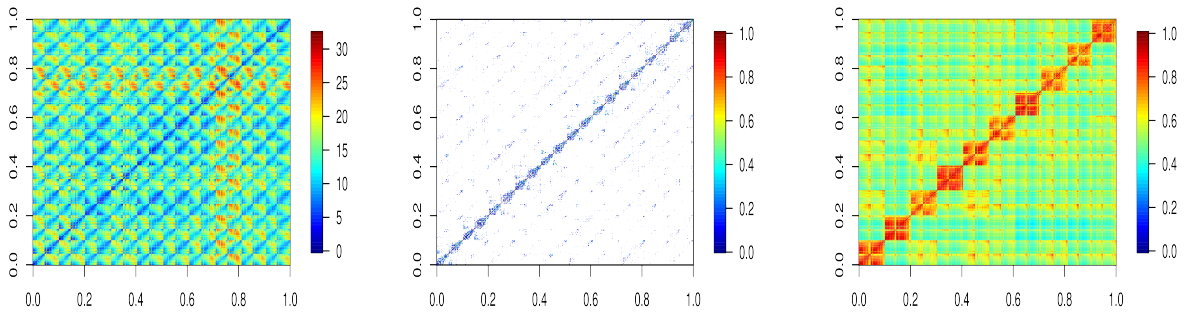
Table S.1: Clustering 640 face photos collected from 10 subjects.

Tabel S.1 shows the results. We use the K-means function from the native R library, the Mclust function in the MCLUST package (Scrucca et al., 2016) for various Gaussian mixture models, the hddc function in HDclassif (Bergé et al., 2012) package for Gaussian mixture models with near low-rank covariance matrices, and the specc function from the kernlab package (Karatzoglou et al., 2004) for the spectral clustering algorithm. For those spectral clustering algorithms, we use w_i's as the sparse representation estimated from the lasso regression, without imposing a low-cardinality constraint. For the latent forest model, we use z_i's with cardinality constraint at $d = 4$.

Clearly, for this high-dimensional dataset, clustering the sparse representation z_i's (or w_i's) instead of y_i's has a significantly improved accuracy. Interestingly, we found that K-

means on those w_i's produce much worse results than all the spectral clustering algorithms. This suggests that forest models (as a generative model for spectral clustering) give a better fit to those w_i's, compared to the Gaussian mixture models (as a generative model for K-means). Lastly, compared to the existing spectral clustering algorithms using similarity $|w_{i,j}| + |w_{j,i}|$ (Vidal, 2011) or $(w_i' w_j)_+$ (Wu et al., 2014), imposing a cardinality constraint seemed to further improve the signal that is helpful for clustering. To verify this, we also conducted spectral clustering using similarity $(z_i' z_j)_+$ and obtained almost the same clustering accuracy as the one from the latent forest model.

As shown in Figure S.2, for this high-dimensional dataset, the pairwise Euclidean distance $\|y_i - y_j\|_2$'s are too noisy to be used for clustering, the inner product on the sparse z_i has much less noise, and the pairwise co-assignment probability matrix produces a very clear partition of 10 clusters.



(a) Euclidean distances between y_i's.

(b) $(z_i' z_j)$ between the latent z_i's.

(c) $\Pr(c_i = c_j \mid y)$ from the latent forest model.

Figure S.2: Pairwise information between the observed y_i's, and sparse latent z_i's, and posterior co-assignment probability matrix in the latent forest model.

S 1.2 Extension to Covariate-dependent Forest Clustering

Following Müller et al. (2011), we now illustrate an extension where the clustering is dependent on external covariates x_i's (each x_i is an m-dimension vector). Müller et al. (2011) proposed the following covariate-dependent product partition model (PPMx):

$$\Pi_0(V_1, \ldots, V_K \mid K, x) \propto \prod_{k=1}^{K} C(V_k) G(\{x_i\}_{i \in V_k}),$$

where C and G together form a modified cohesion function, with G positive-valued and quantifying the overall similarity among those $x_i : i \in V_k$. To specify G, Müller et al. (2011) proposed to use

$$G(\{x_i\}_{i \in V_k}) = \int [\prod_{i \in V_k} \tilde{g}_1(x_i; \xi_k)] \tilde{g}_2(\xi_k) d\xi_k$$

with $\tilde{g}_1$ and $\tilde{g}_2$ some probability density/mass functions with conjugacy, such as $\tilde{g}_1$ as multivariate Gaussian $\mathrm{N}(\cdot \mid \mu_k, \Sigma_1)$ and $\tilde{g}_2$ as Gaussian for $\mathrm{N}(\mu_k \mid 0, \Sigma_2)$, with Σ_1 and Σ_2 some fixed parameters. Importantly, the purpose of G is to form a density-based cohesion function as a priori, hence G is not interpreted as the generative distribution for x_i's.

We note that the above $G(\{x_i\}_{i \in V_k})$ effectively treats $x_i : i \in V_k$ as conditionally independent. Now suppose there is a tree T_k, we can equivalently form a joint distribution by starting from a $x_{k*} \sim \tilde{g}_1(\cdot; \xi_k)$, and then for any $(i, j) \in T_k$, $(x_j - x_i) \sim \tilde{g}_1^*(\cdot; \xi_k)$, with $\tilde{g}_1^*$ the transformed distribution on the difference. Therefore, we have a tree-based similarity function:

$$G(\{x_i\}_{i \in V_k}; T_k) = \int \left[\tilde{g}_1(x_{k^*}; \xi_k) \prod_{(i,j) \in T_k} \tilde{g}_1^*(x_j - x_i; \xi_k) \right] \tilde{g}_2(\xi_k) d\xi_k.$$

In this section, we use Gaussian $\tilde{g}_1$ and $\tilde{g}_2$ as mentioned above. We have $\tilde{g}_1^*$ as $\mathrm{N}(\cdot \mid 0, 2\Sigma_1)$.

After integration, we have

$$G(\{x_i\}_{i\in V_k}; T_k) = \prod_{(i,j)\in T_k} \underbrace{|2\pi(2\Sigma_1)|^{-1/2} \exp\Big[-(x_i - x_j)'(4\Sigma_1)^{-1}(x_i - x_j)\Big]}_{f_0(x_i;x_j)} \times$$
$$\underbrace{|2\pi\Sigma_1\Sigma_2|^{-1/2}|\Sigma_1^{-1} + \Sigma_2^{-1}|^{-1/2} \exp\Big[-\frac{1}{2}x'_{k^*}\Sigma_1^{-1}x_{k^*} + \frac{1}{2}x'_{k^*}\Sigma_1^{-1}(\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}\Sigma_1^{-1}x_{k^*}\Big]}_{r_0(x_k^*)},$$

where we use f_0 and r_0 to simplify notation. Therefore, we can achieve similar effects of PPMx, using an x-informative tree distribution:

$$\Pi(E_k \mid V_k)\Pi(k^* \mid E_k, V_k) = \frac{r_0(x_{k^*}) \prod_{(i,j)\in T_k} f_0(x_i; x_j)}{[\sum_{k\in V_k} r_0(x_k)][\sum_{T'_k} \prod_{(i,j)\in T'_k} f_0(x_i; x_j)]},$$
$$\Pi_0(V_1, \ldots, V_K, K) \propto \lambda^K \Big\{ \prod_{k=1}^{K} [\sum_{k\in V_k} r_0(x_k)][\sum_{T'_k} \prod_{(i,j)\in T'_k} f_0(x_i; x_j)] \Big\}.$$

Note that if $f_0(x_i; x_j) \propto 1$ for any (x_i, x_j), and $r_0(x_i) \propto 1$ for any x_i, then the above would be $\Pi_0(V_1, \ldots, V_K, K) = \lambda^K n_k^{n_k - 1}$, the same as the distribution we describe in the main text.

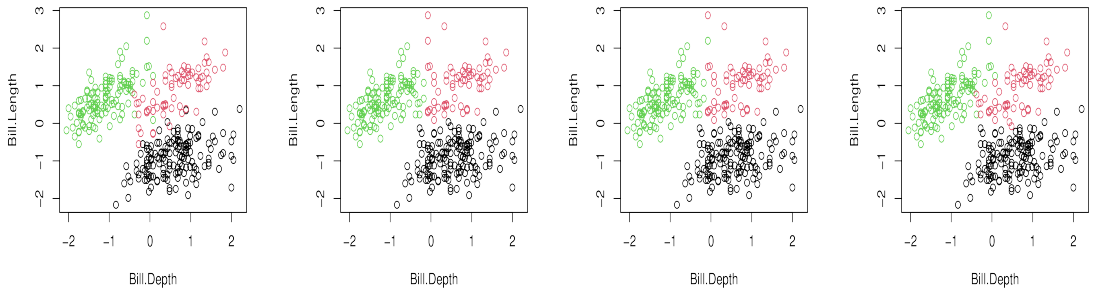
Compared to directly clustering (y_i, x_i) as the joint observation together, a strength of the above approach (and PPMx methods in general) is that as a priori, we can directly control the influence from x_i to clustering, by adjusting the parameters in G. For example, we use $\Sigma_1 = \Sigma_2 = \eta S_n$ with S_n the empirical covariance of x_i's and $\eta > 0$ an adjustable hyper-parameter. This leads to $f_0(x_i; x_j) = |2\pi(2\Sigma_1)|^{-1/2} \exp[-(x_i - x_j)'(4\Sigma_1)^{-1}(x_i - x_j)]$ and $r_0(x_{k^*}) = |2\pi(2\Sigma_1)|^{-1/2} \exp[-x'_{k^*}(4\Sigma_1)^{-1}x_{k^*}]$. As η increases, the influence of x_i becomes weaker. Note that if we were to use x_i in a likelihood, we would not have such flexibility.

To illustrate this model, we use the Palmer Penguins dataset provided in the "palmerpenguins" package (Horst et al., 2020). To clarify, for such a clean dataset, existing approaches such as the Gaussian mixture model can also produce a similarly good accuracy; our goal here is to illustrate the high extensibility of the forest model via distribution

specification.

We remove the duplicated data entries and have a sample of size $n = 334$. The dataset has observations about three species of Antarctic penguins, containing the length and depth measurements of each penguin's bill (in mm). These two variables contain strong signals for distinguishing between species, and we denote each record of (length, depth) by y_i. In addition, the dataset also has measurements of flipper length (in mm) and body mass (in grams), and we denote each record by x_i.

As shown in Figure S.3, the forest model without using covariates correctly estimates most species labels. On the other hand, including the external information from x_i further increases the accuracy.



(a) Penguin data, colored by species.

(b) Forest model estimate without using covariates.

(c) Forest model estimate using x_i as external covariates with $\eta = 2$.

(d) Forest model estimate using x_i as external covariates with $\eta = 1$.

Figure S.3: Clustering the penguin data that contain records of bill length and depth. The forest model alone (Panel b) leads to a good estimate (accuracy 94.6%). Nevertheless, using external covariates (flipper length and body mass) gives more accurate estimates (Panel c: accuracy 95.8%, Panel d: accuracy 97.3%).

S 1.3 Algorithm for Estimating Multi-view Clustering

We use $k_i^{(s)} \in \{1, \ldots, \tilde{K}\}$ to denote the latent assignment $k_i^{(s)} = l$, for $\eta_i^{(s)} = \eta_l^*$. We use the following Gibbs sampling algorithm:

- Using $L_{\mathcal{T}^{(s)}_{1:n,1:n}}$ to denote the Laplacian matrix of the forest graph without auxiliary node 0, we have

$$z^{(s)} \mid \eta^{(s)}, \mathcal{T}^{(s)} \sim \mathrm{N}\Big\{(L_{\mathcal{T}^{(s)}_{1:n,1:n}}/\rho + I/\sigma_z^2)^{-1}(\eta^{(s)}/\sigma_z^2), (L_{\mathcal{T}^{(s)}_{1:n,1:n}}/\rho + I/\sigma_z^2)^{-1}\Big\}.$$

- Sample

$$\Pr(k_i^{(s)} = l \mid \cdot) \propto v_{i,l} \exp\big(-\frac{1}{2}\|z_j^{(s)} - \eta_l^*\|_2^2/\sigma_z^2\big).$$

- Sample $v_i \sim \mathrm{Dir}(\{1/\tilde{\kappa} + \sum_s 1(k_i^{(s)} = l)\}_{l=1,\ldots,\tilde{\kappa}})$.

- Sample $\mathcal{T}^{(s)}$ and $\theta^{(s)}$ for all s, according to the algorithm in Section 3.1 of the main text.

S 2 Proof of Theorems

S 2.1 Proof of Theorem 1

Proof: For ease of notation, in this proof, we use $p = (n + 1)$.

1. Obtain closed-form of the marginal connecting probability.

Since L correspond to a connected graph with weight $A_{i,j} = \exp(W_{i,j})$ for $i \neq j$ and $A_{i,i} = 0$, hence only has one eigenvalue equal to 0 and with eigenvector $\vec{1}/\sqrt{p}$, therefore we

have:

$$p^{-1}\prod_{i=2}^{n+1}\lambda_{(i)}(L) = |L + J/p^2|,$$

where $J = \vec{1}\vec{1}^{\mathrm{T}}$. Let $\tilde{L} = L + J/p^2$, differentiating $\log|\tilde{L}|$ with respect to $W_{i,j}$ yields:

$$M_{i,j} = (\Omega_{i,i} + \Omega_{j,j} - 2\Omega_{i,j})A_{i,j},$$

where $\Omega = \tilde{L}^{-1}$, and $M_{i,i} = 0$.

2. Obtain M as a perturbation form

Let $L = \sum_{l=1}^{p}\lambda_l\psi_l\psi_l^{\mathrm{T}}$ be the eigendecomposition of L, and $N = D^{-1/2}LD^{-1/2}$. Note that,

$$\begin{aligned}
M_{i,j} =& (\Omega_{i,i} + \Omega_{j,j} - 2\Omega_{i,j})A_{i,j} \\
=& (\Omega_{i,i} + \Omega_{j,j} - 2\Omega_{i,j})\{-L_{i,j}1(j \neq i)\} \\
\overset{(a)}{=}& \vec{b}_{i,j}^{\mathrm{T}}(L + J/p^2)^{-1}\vec{b}_{i,j}(-L_{i,j}) \\
=& D_i^{1/2}\vec{b}_{i,j}^{\mathrm{T}}(L + J/p^2)^{-1}\vec{b}_{i,j}D_j^{1/2}(-N_{i,j})
\end{aligned}$$

where (a) is due to $\Omega_{i,i} + \Omega_{j,j} - 2\Omega_{i,j} = 0$ if $1(j \neq i) = 0$, hence $1(k \neq i)$ can be omitted; $\vec{b}_{i,j}$ is a binary vector with the ith element 1 and the kth element -1, and all other elements 0.

Let $\alpha_{i,j} := D_i^{1/2}\vec{b}_{i,j}^{\mathrm{T}}(L + J/p^2)^{-1}\vec{b}_{i,j}D_j^{1/2}$ for $i \neq j$. Since $N_{i,i} = 1$, and $x(I - N)$ has the same eigenvectors as N for any scalar $x > 0$, we see that $M = -\alpha \circ (I - N)$ is an element-wise perturbation of $x(I - N)$. Therefore, our next task is to show α is close to a simple xJ for some $x > 0$.

3. Bound the difference between the K leading eigenvectors.

Slightly changing the above,

$$\begin{aligned}
\alpha_{i,j} &= D_i^{1/2}\vec{b}_{i,j}^{\mathrm{T}}(L + J/p^2)^{-1}\vec{b}_{i,j}D_j^{1/2} \\
&= D_i^{1/2}\vec{b}_{i,j}^{\mathrm{T}}D^{-1/2}(N + D^{-1/2}JD^{-1/2}/p^2)^{-1}D^{-1/2}\vec{b}_{i,j}D_j^{1/2}.
\end{aligned}$$

Using $N = I - D^{-1/2}AD^{-1/2}$, we have

$$\begin{aligned}(N + D^{-1/2}JD^{-1/2}/p^2)^{-1} &= \{I - D^{-1/2}(A - J/p^2)D^{-1/2}\}^{-1} \\ &\overset{(a)}{=} I + \sum_{k=1}^{\infty}\{D^{-1/2}(A - J/p^2)D^{-1/2}\}^k \\ &= I + E.\end{aligned}$$

where (a) uses the Neumann expansion, as E is not divergent:

$$E = D^{1/2}(L + J/p^2)^{-1}D^{1/2} - I = D^{1/2}(\sum_{l=2}^{p}\lambda_l^{-1}\psi_l\psi_l^{\mathrm{T}} + J)D^{1/2} - I,$$

as $\lambda_2 > 0$, E is bounded element-wise for any D with finite-value elements. Further, note that $D_i^{-1/2}(A_{i,j} - 1/p^2)D_j^{-1/2} \to 0$ and monotonically decreasing for fixed $A_{i,j}$ and increasing D_i or D_j; hence E is always bounded elementwise even as all $D_i \to \infty$. We denote the bound constant by $\max_{i,j}|E_{i,j}| \leq \epsilon$.

Combining the above,

$$\begin{aligned}\alpha_{i,j} &= D_i^{1/2}D_j^{1/2}\vec{b}_{i,j}^{\mathrm{T}}\left\{D^{-1} + D^{-1/2}ED^{-1/2}\right\}\vec{b}_{i,j} \\ &= D_i^{1/2}D_j^{1/2}\left\{(D_i^{-1} + D_j^{-1}) + (D_i^{-1}E_{i,i} + D_j^{-1}E_{j,j} - 2D_i^{-1/2}D_j^{-1/2}E_{i,j})\right\}.\end{aligned}$$

Now we can bound the difference between M and $x(I - N)$ minimized over x:

$$\begin{aligned}&\min_x \max_{i,j}|\{(\alpha - xJ)\circ(I - N)\}_{i,j}| \\ &= \min_x \max_{i,j}|(\alpha_{i,j} - x)(D_i^{-1/2}D_j^{-1/2}A_{i,j})| \\ &= \min_x \max_{i,j}\left|\left\{(D_i^{-1} + D_j^{-1}) - xD_i^{-1/2}D_j^{-1/2} + (D_i^{-1}E_{i,i} + D_j^{-1}E_{j,j} - 2D_i^{-1/2}D_j^{-1/2}E_{i,j})\right\}A_{i,j}\right| \\ &\leq \min_x \max_{i,j}\left|\left\{(D_i^{-1} + D_j^{-1}) - xD_i^{-1/2}D_j^{-1/2} + (D_i^{-1}\epsilon + D_j^{-1}\epsilon + 2D_i^{-1/2}D_j^{-1/2}\epsilon)\right\}A_{i,j}\right| \\ &\overset{(a)}{\leq} \max_{i,j}\left\{(1+\epsilon)(D_i^{-1/2} - D_j^{-1/2})^2A_{i,j}\right\},\end{aligned}$$

where (a) chooses $x = 2(1 + \epsilon) + 2\epsilon$.

Using Theorem 2 from Yu et al. (2015), there exists a orthonormal matrix $R \in \mathbb{R}^{K\times K}$, such that,

$$\|\Psi_{1:K} - \phi_{1:K}R\|_F \leq \frac{2^{3/2}\min\{\sqrt{K}\|(\alpha - xJ)\circ(I-N)\|_{op},\ \|(\alpha - xJ)\circ(I-N)\|_F\}}{\xi_K - \xi_{K+1}} \quad (1)$$

for any $x > 0$.

Since $|\{(\alpha - xJ)\circ(I-N)\}_{i,j}|$ is upper-bounded hence is sub-Gaussian with bound parameter $\sigma_e = \max_{i,j}\{(1+\epsilon)(D_i^{-1/2} - D_j^{-1/2})^2 A_{i,j}\}$.

Using Theorem 1 of Duan, Michailidis and Ding 2020 (arXiv preprint:1910.02471), with probability $1 - \delta_t$

$$\|\Psi_{1:K} - \phi_{1:K}R\|_F \leq \frac{2^{3/2}(\sqrt{Kp}\sigma_e)}{\xi_K - \xi_{K+1}}t.$$

where $\delta_t = \exp[-(t^2/64 - \log(5\sqrt{2}))p]$. Taking $t = 14$, we have $(t^2/64 - \log(5\sqrt{2})) > 1$. Therefore, we have with probability at least $1-\exp(-p)$ {which is greater than $1-\exp(-n)$}.

$$\|\Psi_{1:K} - \phi_{1:K}R\|_F \leq \frac{40\sqrt{Kp}\sigma_e}{\xi_K - \xi_{K+1}}.$$

□

S 2.2 Proof of Theorem 2 and 3

Let the conditional probability associated with Gaussian leaf density f be $\Pr\{B(y_1, M_n/2) \mid y_1\} = m_n$, where $B(y_1, M_n/2)$ stands for an open ball of radius $M_n/2$ around y_1. If the true number of clusters is $K = K_0$, then m_n^{n-K-1} is the probability that the distances $\{d^0_{\ell,n}\}_l$ in the minimum spanning tree under null are all below M. Specifically, let $E_n = \{d^0_{\ell,n} \leq M_n/2 : 1 \leq \ell \leq n-K-1\} =$. Then $\Pr(E_n) = m_n^{n-K-1}$.

With $x_i \sim \mathrm{N}(0, \sigma^{0,n})$, we have $m_n = \Pr(\frac{\sum_{i=1}^p x_i^2}{\sigma^{0,n}} < \frac{M_n^2}{2^2\sigma^{0,n}}) = 1 - \frac{\Gamma\left(p/2, \frac{M^2}{2^3\sigma^{0,n}}\right)}{\Gamma(p/2)}$, where $\Gamma(\cdot,\cdot)$ stand for the upper incomplete gamma function using cumulative distribution function of χ^2 distribution. Since p belongs to the set of natural numbers, we have $\Gamma(p/2, x) < C_1 x^{p/2}e^{-x}$ except for $p = 1.5$ and any $x > 0$ with some constant C_1 which depends on p (Pinelis,

2020). However, for large x, we have $\Gamma(p/2, x) < C_1 x^{p/2} e^{-x}$ even for $p = 1.5$. We then have $m_n > 1 - \frac{C_2}{\Gamma(p/2)} (\log n)^{p/2-1} e^{-\tilde{m}_0 \log n} = 1 - \frac{C_2}{\Gamma(p/2)} \frac{(\log n)^{p/2-1}}{n^{\tilde{m}_0}}$ where $C_2 = C_1 (\tilde{m}_0)^{p/2-1}$. Since $\tilde{m}_0 > (p/2 + 2)$, we have $m_n^{n-K-1} > 1 - (n - K - 1) \frac{C_2}{\Gamma(p/2)} \frac{(\log n)^{p/2-1}}{n^{\tilde{m}_0}} \to 1$ as $n \to \infty$ as $\frac{\log n}{n}$ goes to 0. Hence $\text{pr}(E_n) \to 1$.

We further have, $\sum_{n \geq K} [1 - \{1 - \frac{C_2}{\Gamma(p/2)} \frac{(\log n)^{p/2-1}}{n^{\tilde{m}_0}}\}^{n-K-1}] < \sum_n n \frac{C_2}{\Gamma(p/2)} \frac{(\log n)^{p/2-1}}{n^{\tilde{m}_0}}$. Thus for $\tilde{m}_0 > 2 + p/2$, we have $\sum_n [1 - \{1 - \frac{C_2}{\Gamma(p/2)} \frac{(\log n)^{p/2-1}}{n^{\tilde{m}_0}}\}^{n-K-1}] < \infty$. Then, by the Borel-Cantelli Lemma, we also have almost sure convergence of this event.

We now show for $y \in E_n$, the ratio of the maximum posterior probability assigned to a non-true clustering arrangement to the posterior probability assigned to the true clustering arrangement converges to zero.

Proof of Theorem 2

Note that for any σ and a given R, the posterior $\Pi(\mathcal{T}_{\text{MST},R}, \sigma \mid y)$ is maximized at the $\mathcal{T}$, which is a combination of minimum spanning trees constructed within the regions R_k's. Thus,

We have $\frac{\Pi(R_0|y)}{\Pi(\mathcal{T}_{\text{MST},R^0}, \sigma^{0,n}|y)} > 1$ as $\Pi(R_0 \mid y) = \sum_{\mathcal{T}} \Pi(\mathcal{T}, R^0, \sigma^{0,n} \mid y)$.

$$\frac{\Pi(\mathcal{T}_{\text{MST},R}, \sigma^{0,n} \mid y)}{\Pi(\mathcal{T}_{\text{MST},R^0}, \sigma^{0,n} \mid y)} \leq \left(\frac{\epsilon_2}{\epsilon_1}\right)^K \exp\left(- \sum_{\ell=1}^{n-K_0-1} d_{\ell,n}^2/(2\sigma^{0,n}) + \sum_{\ell=1}^{n-K-1} (d_{\ell,n}^0)^2/(2\sigma^{0,n}) \right),$$

where $\sum_{\ell=1}^{n-K-1} d_{\ell,n}^2$ is the total squared norm distance on the minimum spanning tree under the partition regions R excluding the edges with the root node and $\sum_{\ell=1}^{n-K_0-1} (d_{\ell,n}^0)^2$ is the same under $\mathcal{T}_{\text{MST},R^0}$. The above is because based on the Prim's algorithm (Prim, 1957), the minimum spanning tree is equal to the result of sequential growing a tree starting from one node, each time by adding an edge (along with a node) with the shortest distance between one node in the existing tree and one of the remaining nodes not yet in the tree. Clearly,

at each step, the edge choice is unaffected when changing distance from d to d^2; therefore, the minimum spanning trees based on the sum of $d_{l,n}^2$ and the sum of $d_{l,n}$ are the same.

Since, $\inf_{x\in R_i^0, y\in R_j^0} \|x-y\|_2 > M_n$, for all $i \neq j$, for at least one ℓ, we must have $d_{\ell,n} > M$. Due to the above result, with probability at least m_n^n, we have $\sum_{\ell=1}^{n-K-1}(d_{\ell,n})^2 > \sum_{\ell=1}^{n-K-1}(d_{\ell,n}^0)^2 + M_n^2/4$, which implies $\frac{\Pi(\mathcal{T}_{\text{MST},R},\sigma^{0,n}|y)}{\Pi(\mathcal{T}_{\text{MST},R^0},\sigma^{0,n}|y)} < n^{-\tilde{m}_0}$. And we further have that $m_n^n \to 1$ as $n \to 1$.

Proof of Theorem 3

First, we consider that the alternative partitioning has a lower number of clusters than the null. Let that be K, which is less than K_0. Then we have

$$\frac{\Pi(\mathcal{T}_{\text{MST},R}, \sigma^{0,n} \mid y)}{\Pi(\mathcal{T}_{\text{MST},R^0}, \sigma^{0,n} \mid y)}$$
$$\leq \lambda^{K-K_0}\frac{K_0!}{K!}\left(\frac{\epsilon_2}{\epsilon_1}\right)^K \frac{(\sigma^{0,n})^{(K_0-K)/2}}{\epsilon_1^{K_0-K}} \exp\left(-\sum_{\ell=1}^{n-K-1} d_{\ell,n}^2/(2\sigma^{0,n}) + \sum_{\ell=1}^{n-K_0-1}(d_{\ell,n}^0)^2/(2\sigma^{0,n})\right),$$

We again must have $\sum_{\ell=1}^{n-K-1}(d_{\ell,n})^2 > \sum_{\ell=1}^{n-K_0-1}(d_{\ell,n}^0)^2 + M_n^2/4$ with probability $m_n^n \to 1$ as the alternative partitioning will have edges with length greater than M_n.

Next, we show the above when the alternative partitioning has a larger number of clusters than the null. Specifically, for $K > K_0$, we replace A3 and vary the conditions on $r(y)$ with n.

Then we have

$$\frac{\Pi(\mathcal{T}_{\text{MST},R}, \sigma^{0,n} \mid y)}{\Pi(\mathcal{T}_{\text{MST},R^0}, \sigma^{0,n} \mid y)}$$
$$\leq \lambda^{K-K_0}\frac{K_0!}{K_2!}\left(\frac{c_2}{c_1}\right)^K c_2^{K-K_0}(\sigma^{0,n})^{(K_0-K)/2}$$
$$\times \exp\left(-(K-K_0)M_n^2/(2\sigma^{0,n}) - \sum_{\ell=1}^{n-K-1} d_{\ell,n}^2/(2\sigma^{0,n}) + \sum_{\ell=1}^{n-K_0-1}(d_{\ell,n}^0)^2/(2\sigma^{0,n})\right),$$

Again, for any $K > K_0$, the above ratio goes to zero as $n \to \infty$ we have $1/(\sigma^{0,n}n) \to 0$ with probability at least $m_n^n \to 1$.

S 3 Posterior consistency of the clustering

Here, we study the clustering consistency of our Bayesian methods when the number of clusters is known.

Theorem S 1 *Under some assumptions outlined below, we have* $\Pi(R \neq R_0|y) \to 0$ *almost surely, unless* $R_i^0 \subseteq R_{\xi(i)}$ *for some permutation map* $\xi(\cdot)$ *when number of clusters in known.*

The total number of possible clusters with n data points and K clusters is $\binom{n-1}{K-1}$, which is of order n^K. To show clustering consistency, we require the following assumption,

- (S1, Diminishing scale and minimum separation) We let $\sigma^{0,n} = C'(1/n\log^{1+\iota} n$ for some $\iota > 0$ and $C' > 0$ and $\inf_{x\in R_k^0, y\in R_{k'}^0} \|x-y\|_2 > M_n$, for all $k \neq k'$ with some positive constant $M_n > 0$ such that $M_n^2/\sigma^{0,n} = 8\tilde{m}_0 n\log(n)$ for all (i,j) and is known for some constant $\tilde{m}_0 > p/2+2$.

In the above assumption, the main requirement is $M_n^2/\sigma^{0,n} = 8\tilde{m}_0 n\log(n)$ which is achieved by allowing the scale to decay faster than Assumption A1. Alternatively, one may increase M_n instead of reducing $\sigma^{0,n}$. However, from a practical point of view, one would expect M_n to be a non-increasing function of n.

$$\frac{\Pi(R \mid y)}{\Pi(R_0 \mid y)} = \frac{\frac{\Pi(R|y)}{\Pi(\mathcal{T}_{\mathrm{MST},R},\sigma^{0,n}|y)}}{\frac{\Pi(R_0|y)}{\Pi(\mathcal{T}_{\mathrm{MST},R^0},\sigma^{0,n}|y)}} \frac{\Pi(\mathcal{T}_{\mathrm{MST},R},\sigma^{0,n} \mid y)}{\Pi(\mathcal{T}_{\mathrm{MST},R^0},\sigma^{0,n} \mid y)},$$

We have $\frac{\Pi(R_0|y)}{\Pi(\mathcal{T}_{\mathrm{MST},R^0},\sigma^{0,n}|y)} > 1$ and $\frac{\Pi(R|y)}{\Pi(\mathcal{T}_{\mathrm{MST},R},\sigma^{0,n}|y)} < n^{n-2}$ (the total number of possible spanning trees with n points) hence $\frac{\Pi(R|y)}{\Pi(R_0|y)} \lesssim n^{n-2}\frac{\Pi(\mathcal{T}_{\mathrm{MST},R},\sigma^{0,n}|y)}{\Pi(\mathcal{T}_{\mathrm{MST},R^0},\sigma^{0,n}|y)}$

When the number of clusters is known,

$$\frac{1-\Pi(R^0 \mid y)}{\Pi(R^0 \mid y)} = \sum_{R\neq R^0} \frac{\Pi(R \mid y)}{\Pi(R^0 \mid y)} \lesssim \exp((n+K-2)\log n)\frac{\Pi(\mathcal{T}_{\mathrm{MST},R},\sigma^{0,n} \mid y)}{\Pi(\mathcal{T}_{\mathrm{MST},R^0},\sigma^{0,n} \mid y)}$$

And applying the steps from our previous section, we have $\frac{1-\Pi(R^0|y)}{\Pi(R^0|y)} < \exp(-n\log n)$, goes to zero and thus completes the proof.

S 4 Additional Numerical Experiments

S 4.1 Uncertainty Quantification on Clustering Data from a Mixture Model

We now present some uncertainty quantification results, for clustering data that are from a mixture model. We experiment with $n = 400$ data points in $\mathbb{R}^2$ generated from a two-component mixture distribution:

$$y_i \sim 0.5\mathcal{K}(\cdot \mid \mu_1) + 0.5\mathcal{K}(\cdot \mid \mu_2),$$

for $i = 1, \ldots, n$, with $\mu_1 = (0, 0)$ and $\mu_2 = (b, b)$ two location parameters. We experiment with two settings, with $\mathcal{K}$ as (i) independent bivariate Gaussian distribution $\mathrm{N}(\mu_k, I_2)$, (ii) independent bivariate t distribution with 5 degrees of freedom $t_5(\mu_k)$.

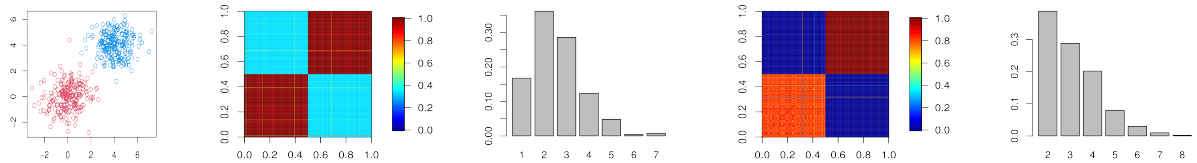
When fitting models, we consider the unknown K scenario, and use the distribution $\Pi(\mathcal{T}) \propto \lambda^K$ for the Bayesian forest model, with $\lambda = 0.5$. For comparison, we use the Dirichlet process Gaussian mixture model (DP-GMM) with a Gamma$(2, 20)$ hyper-prior on the concentration parameter (with prior mean 0.1). We use the "dirichletprocess" package in R (Ross and Markwick, 2018) for estimating the posterior distribution from DPMM. Notice that our two choices of $\mathcal{K}$ above correspond to fitting a Dirichlet process mixture with correctly specified components and one with misspecified components, respectively.

To estimate the posterior, for each model, we ran the MCMC algorithm for 1,000 iterations and discarded the first 500 iterations. We calculated the posterior co-assignment

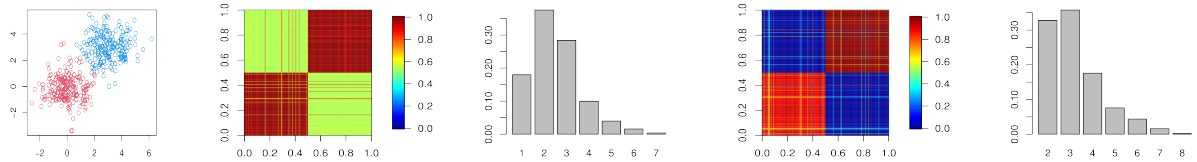
probability matrix $\Pr(c_i = c_j \mid y)$, and the posterior number of clusters $\Pr(K \mid y)$.

When the data are from the Gaussian mixture (Figure S.5), both the DP-GMM and the forest model lead to satisfactory performances, with the mode of $\Pr(K \mid y)$ equal/close to the ground truth at $K = 2$. It is interesting to note that there is a proportion of the posterior sample from the forest model corresponds to $K = 1$. This is likely due to less parametric assumption imposed on the shape of the clusters, compared to the DP-GMM. Nevertheless, the posterior mode of the forest model correctly falls on $K = 2$.

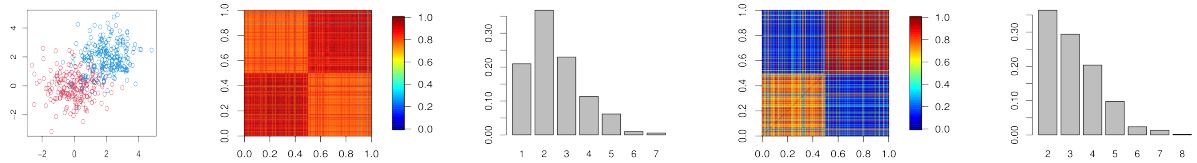
On the other hand, when the data are from the t_5 mixture (Figure S.6), we find the DP-GMM always show an over-estimation problem. Such issues are due to the misspecification in the component distribution, and Cai et al. (2021) have shown that switching to a finite Gaussian mixture with a prior on K does not solve the problem. In comparison, the clustering of the forest model shows much less sensitivity to model specification.



(a) Data from two-component GMM ($b = 4$). | (b) $\Pr(c_i = c_j \mid y)$ from the forest model. | (c) $\Pr(K \mid y)$ from the forest model. | (d) $\Pr(c_i = c_j \mid y)$ from DP-GMM. | (e) $\Pr(K \mid y)$ from DP-GMM.

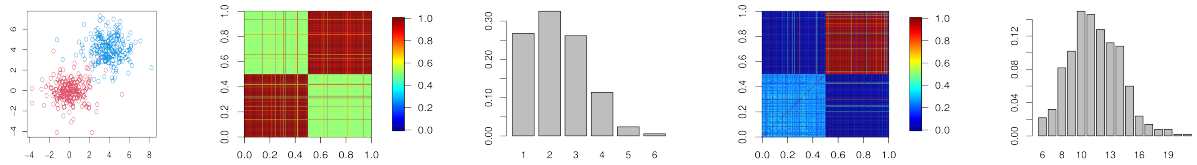


(f) Data from two-component GMM ($b = 3$). | (g) $\Pr(c_i = c_j \mid y)$ from the forest model. | (h) $\Pr(K \mid y)$ from the forest model. | (i) $\Pr(c_i = c_j \mid y)$ from DP-GMM. | (j) $\Pr(K \mid y)$ from DP-GMM.

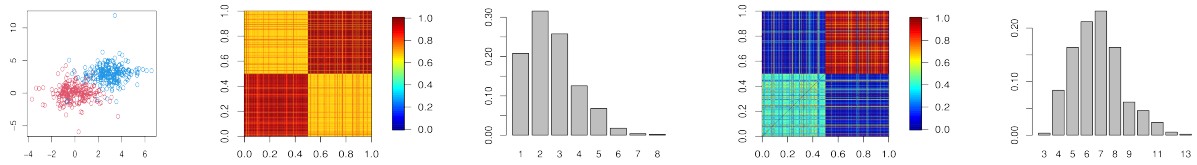


(k) Data from two-component GMM ($b = 2$). | (l) $\Pr(c_i = c_j \mid y)$ from the forest model. | (m) $\Pr(K \mid y)$ from the forest model. | (n) $\Pr(c_i = c_j \mid y)$ from DP-GMM. | (o) $\Pr(K \mid y)$ from DP-GMM.

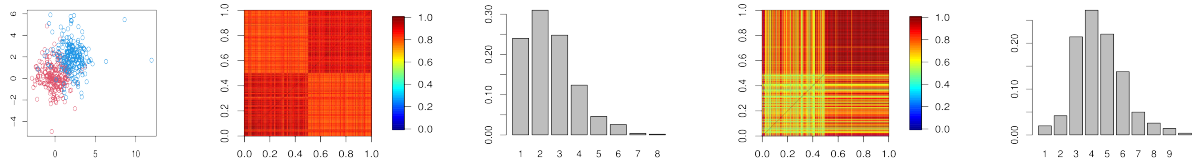
Figure S.4: Uncertainty quantification in clustering data generated from a two-component Gaussian mixture model.



(a) Data from two-component t_5-MM ($b = 4$). (b) $\Pr(c_i = c_j \mid y)$ from the forest model. (c) $\Pr(K \mid y)$ from the forest model. (d) $\Pr(c_i = c_j \mid y)$ from DP-GMM. (e) $\Pr(K \mid y)$ from DP-GMM.



(f) Data from two-component t_5-MM ($b = 3$). (g) $\Pr(c_i = c_j \mid y)$ from the forest model. (h) $\Pr(K \mid y)$ from the forest model. (i) $\Pr(c_i = c_j \mid y)$ from DP-GMM. (j) $\Pr(K \mid y)$ from DP-GMM.

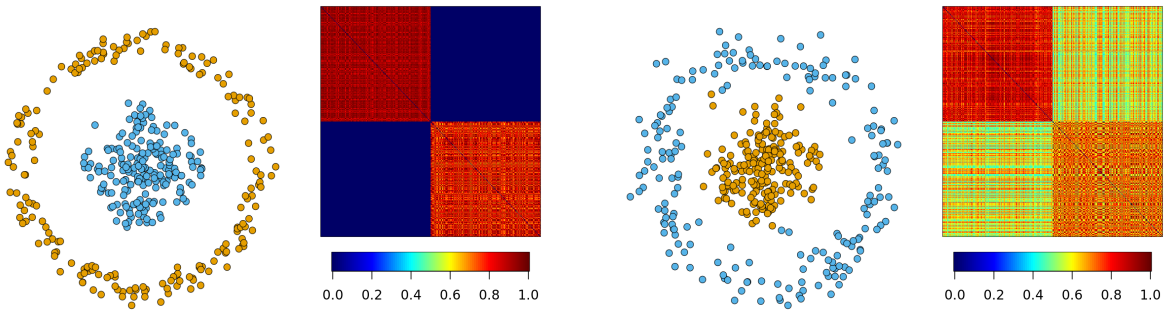


(k) Data from two-component t_5-MM ($b = 2$). (l) $\Pr(c_i = c_j \mid y)$ from the forest model. (m) $\Pr(K \mid y)$ from the forest model. (n) $\Pr(c_i = c_j \mid y)$ from DP-GMM. (o) $\Pr(K \mid y)$ from DP-GMM.

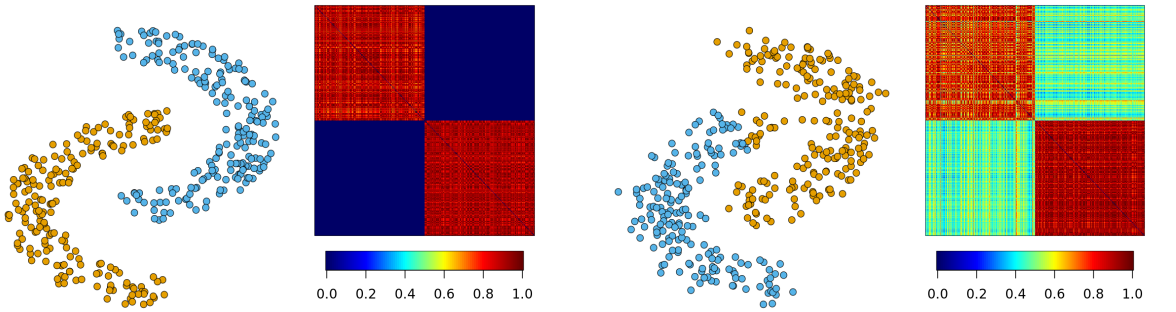
Figure S.5: Uncertainty quantification in clustering data generated from a two-component t_5 mixture model.

S 4.2 Additional Experiments on Clustering Near-Manifold Data

We conduct additional simulations on clustering near-manifold data. The results are shown in Figure S.7.



(a) Posterior point estimate. (b) $\Pr(c_i = c_j \mid y)$. (c) Posterior point estimate. (d) $\Pr(c_i = c_j \mid y)$.



(e) Posterior point estimate. (f) $\Pr(c_i = c_j \mid y)$. (g) Posterior point estimate. (h) $\Pr(c_i = c_j \mid y)$.

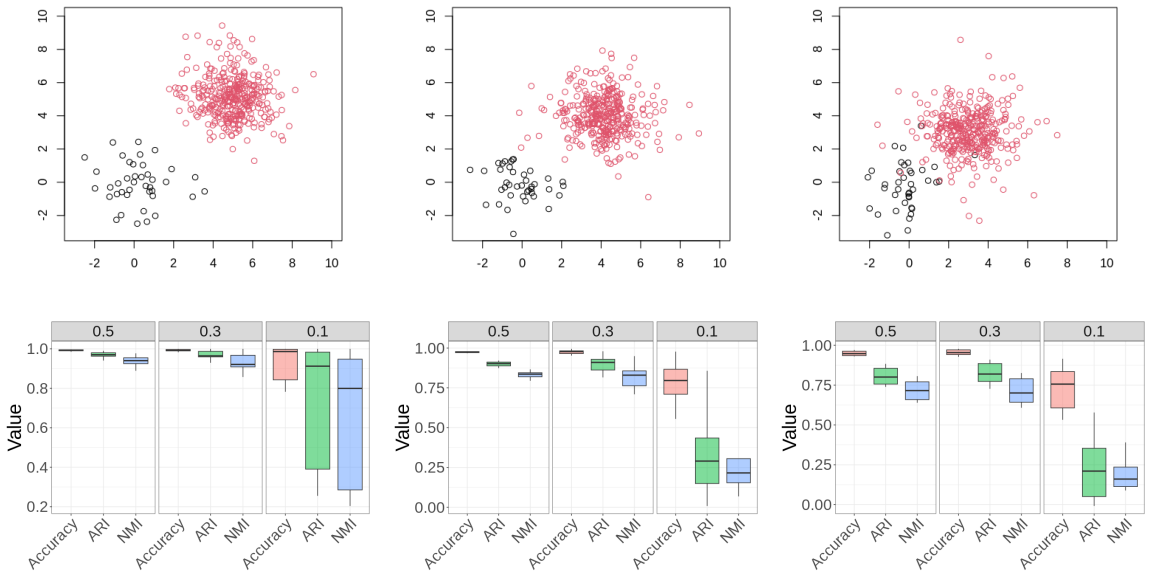
Figure S.6: Clustering data generated near manifolds.

S 4.3 Additional Simulations on Uncertainty and Clustering Accuracy

We now compare the uncertainty and clustering accuracy. We consider three possible scenarios as different sources of uncertainties: increasingly imbalanced cluster sizes, an increasing number of clusters, and an increasing number of noisy points between clusters. In addition, we gradually reduce the separation between clusters, so that the uncertainty can increase as well.

When measuring the clustering accuracy of the point estimate, we calculate the adjusted Rand index (ARI), normalized mutual information (NMI), as well as the clustering accuracy rate (the match rate between the point estimate of $\hat{c}_i$ and each ground-truth label, minimized over all possible label switchings in $\hat{c}_i$). We run 10 times of experiments under each combination of values, and show the boxplots.

For the first scenario, we generate $n = 400$ data points from a two-component independent bivariate t distribution with 5 degrees of freedom, $y_i \sim \tilde{w}_1 t_5(\cdot \mid [0, 0]) + (1 - \tilde{w}_1) t_5(\cdot \mid [\tilde{b}, \tilde{b}])$. We experiment with different values of $\tilde{w}_1 \in \{0.5, 0.3, 0.1\}$ to have different degrees of cluster size imbalance, as well as different values of $\tilde{b} \in \{5, 4, 3\}$ to have different degrees of separation between cluster centers. The results are shown in Figure 7.



(a) Two clusters of imbalanced sizes with two means separated by vector $[5, 5]$ (above). The experiments are repeated with the proportion of Cluster 1 size taken from $\{0.5, 0.3, 0.1\}$, and the clustering accuracy measures are shown below.

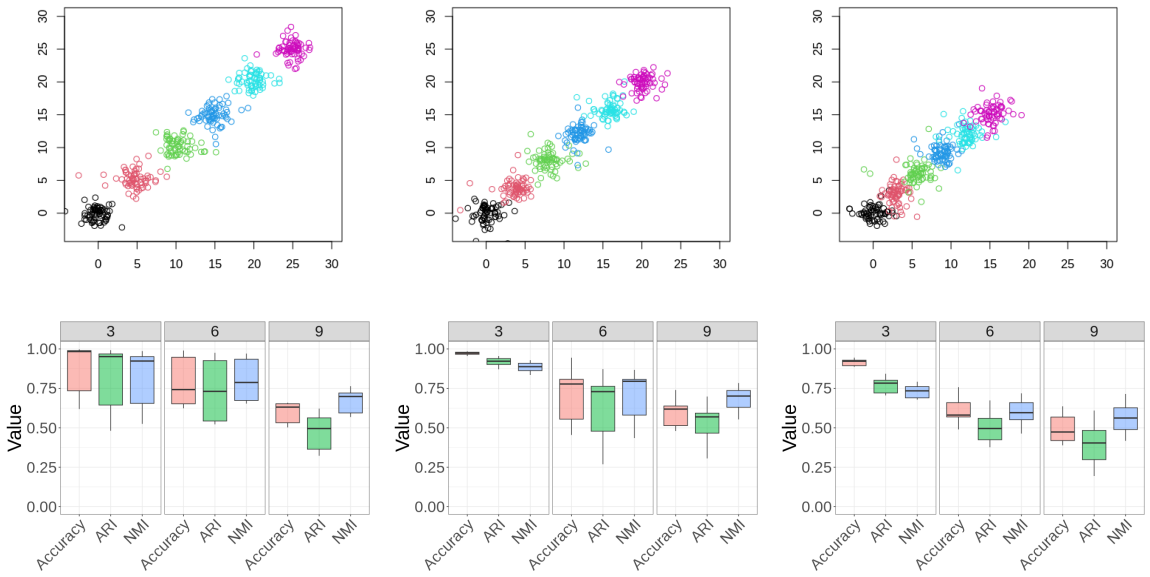
(b) Two clusters of imbalanced sizes with two means separated by vector $[4, 4]$ (above). The experiments are repeated with the proportion of Cluster 1 size taken from $\{0.5, 0.3, 0.1\}$, and the clustering accuracy measures are shown below.

(c) Two clusters of imbalanced sizes with two means separated by vector $[3, 3]$ (above). The experiments are repeated with the proportion of Cluster 1 size taken from $\{0.5, 0.3, 0.1\}$, and the clustering accuracy measures are shown below.

Figure 7: Clustering accuracy decreases as the cluster sizes become more imbalanced. The adjusted Rand index (ARI), normalized mutual information (NMI), and the clustering accuracy rate (Accuracy, the match rate between $\hat{c}_i$ and the ground truth, minimized over all possible label switchings in $\hat{c}_i$) are shown.

For the second scenario, we generate $n = 400$ data points from a $\tilde{K}$-component bivariate t distribution with 5 degrees of freedom, $y_i \sim \sum_{k=1}^{\tilde{K}} (1/\tilde{K}) t_5(\cdot \mid [b_k, b_k])$. We experiment

with different values of $\tilde{K} \in \{3, 6, 9\}$ to have different numbers of clusters, as well as different values of $b_k = 3(k-1)$, $4(k-1)$ or $5(k-1)$ to have different degrees of separation between cluster centers. The results are shown in Figure 8.



(a) Increasing number of clusters.

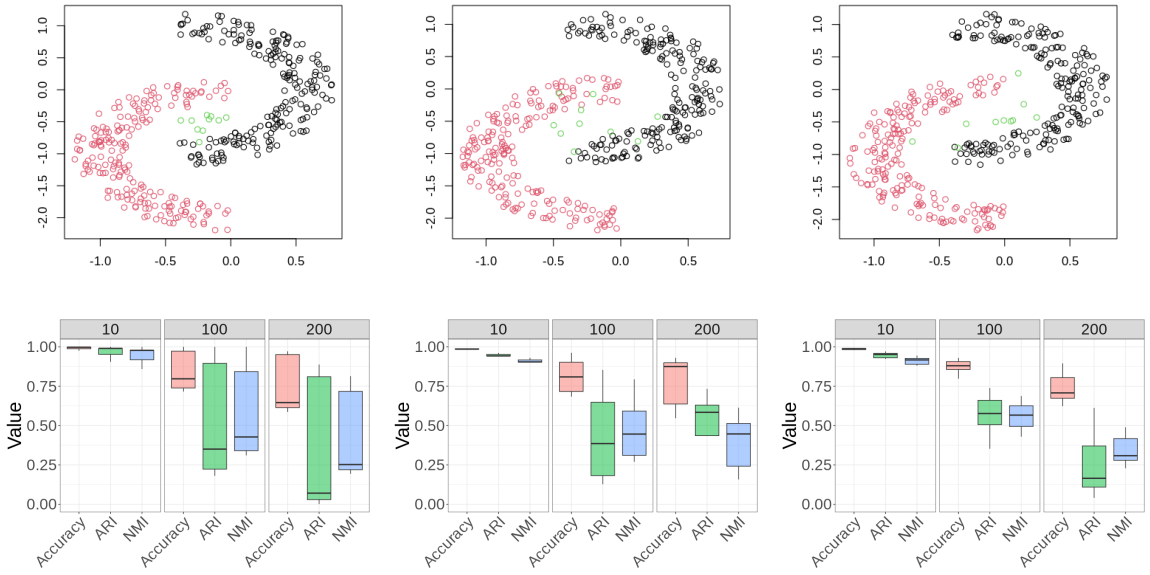
(b) Increasing number of clusters.

(c) Increasing number of clusters.

Figure 8: Clustering accuracy decreases as the number of clusters increases. The adjusted Rand index (ARI), normalized mutual information (NMI) and the clustering accuracy rate (Accuracy, the match rate between $\hat{c}_i$ and the ground-truth, minimized over all possible label switchings in $\hat{c}_i$) are shown.

For the third scenario, we first generate $n = 400$ data points near the two moon manifolds that are well separated from one another, then we add m number of points generated from Gaussian distribution with variance $\tilde{\gamma}^2$, and its center placed between the two manifolds. We experiment with different values of $m \in \{10, 100, 200\}$, so that the clusters would appear somewhat connected to each other as m increases; we also vary $\tilde{\gamma}^2 \in \{0.1^2, 0.2^2, 0.3^2\}$

to have different levels of noise. The results are shown in Figure 9.



(a) Noisy points (green, with variance 0.1^2) between clusters (above). The clustering accuracy measures are collected (below) with the number of noisy points taken from $\{10, 100, 200\}$.

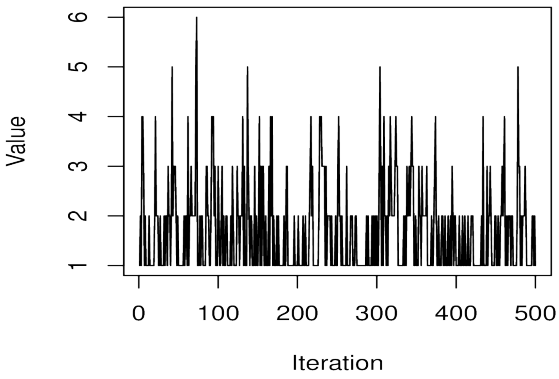
(b) Noisy points (green, with variance 0.2^2) between clusters (above). The clustering accuracy measures are collected (below) with the number of noisy points taken from $\{10, 100, 200\}$.

(c) Noisy points (green, with variance 0.3^2) between clusters (above). The clustering accuracy measures are collected (below) with the number of noisy points taken from $\{10, 100, 200\}$.

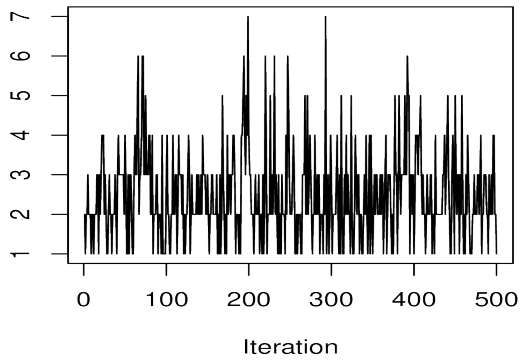
Figure 9: Clustering accuracy decreases as the number of noisy points between clusters increases. The adjusted Rand index (ARI), normalized mutual information (NMI), and the clustering accuracy rate (Accuracy, the match rate between $\hat{c}_i$ and the ground truth, minimized over all possible label switchings in $\hat{c}_i$) are shown.

S 4.4 Diagnostic plots for Markov chain Monte Carlo

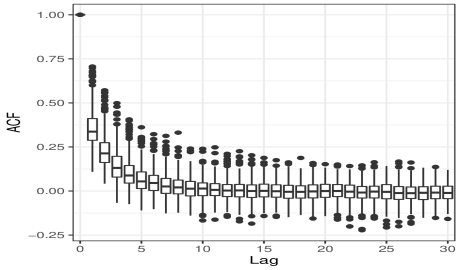
The MCMC algorithm that we describe in the main text shows a fast mixing of Markov chains. To illustrate this, we use the Markov chain collected from the experiment related to Figure S.6(k), and calculate the autocorrelations in (i) the degrees for each node in the forest $D_{i,i}$'s, and (ii) the number of clusters K. We plot the results in Figure S.8.



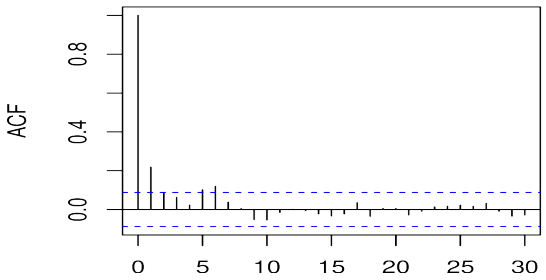
(a) Traceplot of one degree in the forest $D_{1,1}$.



(b) Traceplot of the number of clusters K.



(c) Boxplot of the autocorrelations for $D_{i,i}$'s.

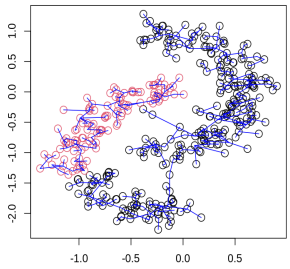


(d) Autocorrelation for the number of clusters K.

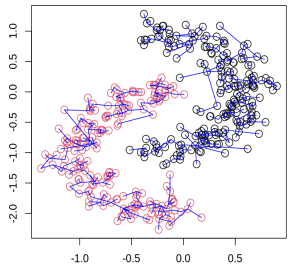
Figure S.10: Traceplots and autocorrelation plots show fast mixing of the MCMC algorithm.

To demonstrate the high efficiency of updating $\mathcal{T}$ in a block via the random-walk covering algorithm (Broder, 1989; Aldous, 1990; Mosbah and Saheb, 1999), we plot the sampled $\mathcal{T}$ at three contiguous iterations (after burn-ins) in Figure S.9. The forest shows a rapid

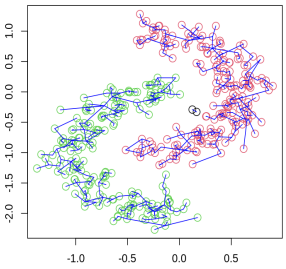
change from iteration to iteration — indeed, the proportion of edge changes (the number of edges $\{(i,j) : (i,j) \in \mathcal{T}_{[t]}, (i,j) \notin \mathcal{T}_{[t+1]}\}$ divided by the total number of edges) is around 50% at each iteration.



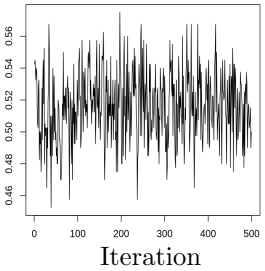
(a) Sampled $\mathcal{T}$ at iteration 1 after burn-ins.



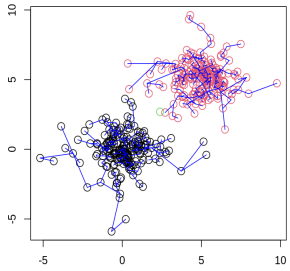
(b) Sampled $\mathcal{T}$ at iteration 2 after burn-ins.



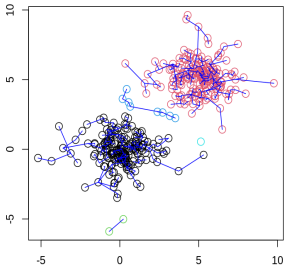
(c) Sampled $\mathcal{T}$ at iteration 3 after burn-ins.



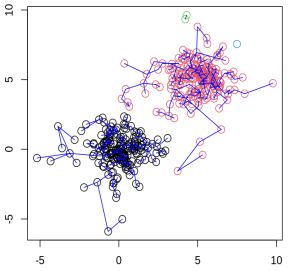
(d) Proportion of edge changed from $\mathcal{T}_{[t]}$ to $\mathcal{T}_{[t+1]}$.



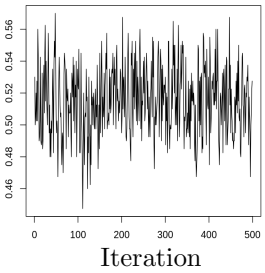
(e) Sampled $\mathcal{T}$ at iteration 1 after burn-ins.



(f) Sampled $\mathcal{T}$ at iteration 2 after burn-ins.



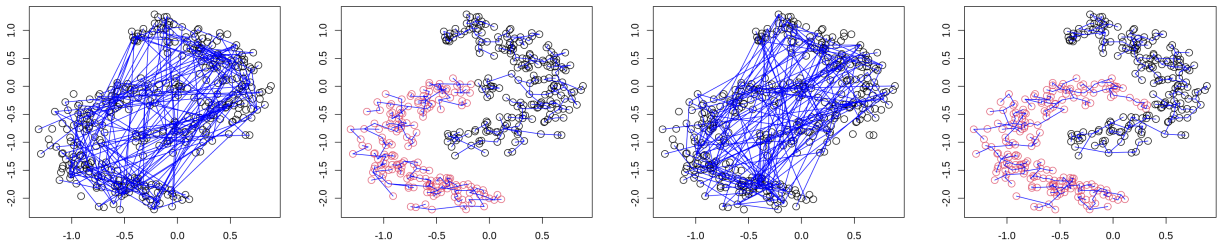
(g) Sampled $\mathcal{T}$ at iteration 3 after burn-ins.



(h) Proportion of edge changed from $\mathcal{T}_{[t]}$ to $\mathcal{T}_{[t+1]}$.

Figure S.11: The forest $\mathcal{T}$ changes rapidly from one iteration to another: Panels a-c plot the forests (blue) at three contiguous iterations, and Panel d shows the proportion of edge changes in each iteration, as measured by the number of edges $\{(i,j) : (i,j) \in \mathcal{T}_{[t]}, (i,j) \notin \mathcal{T}_{[t+1]}\}$ divided by the total number of edges. Panels e-h show the results from another experiment. In both cases, the forest $\mathcal{T}$ has about 50% of edges changed from one iteration to the next.

Further, we assess the convergence by randomly initializing $(\mathcal{T}_{[0]}, \theta_{[0]})$ at 5 different start points, and run 5 separate Markov chains. Specifically, for the elements $\tilde{\sigma}_i$ and γ in $\theta_{[0]}$, we initialize them randomly from Inverse-Gamma$(0.5, 0.5)$, then we draw $\mathcal{T}_{[0]} \sim \Pi(\mathcal{T} \mid \theta_{[0]}, y)$. Figure S.10 shows two randomly initialized $\mathcal{T}$'s. The traceplots of the parameters show the convergence of 5 Markov chains, and we calculate the Gelman–Rubin statistics (potential scale reduction factor, Gelman and Rubin (1992)) and find all of them smaller than 1.1, which indicates convergence.

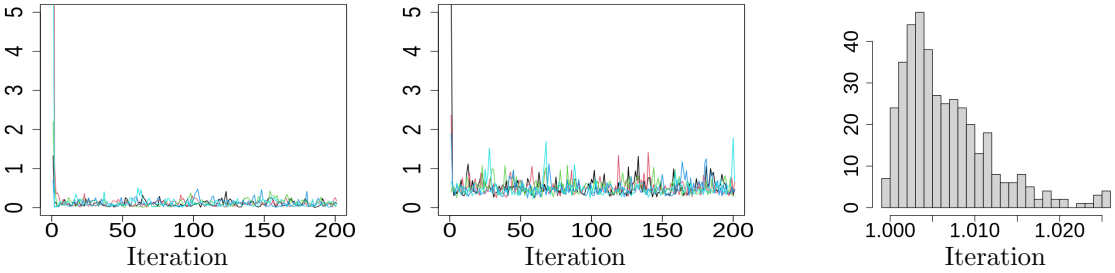


(a) Randomly initialized $\mathcal{T}$ in Chain 1.

(b) One sampled $\mathcal{T}$ in Chain 1 after burn-ins.

(c) Randomly initialized $\mathcal{T}$ in Chain 2.

(d) One sampled $\mathcal{T}$ in Chain 2 after burn-ins.



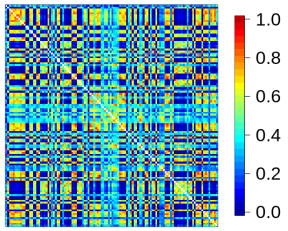
(e) Traceplot for the parameter $\tilde{\sigma}_1$ from 5 chains.

(f) Traceplot for the parameter γ from 5 chains.

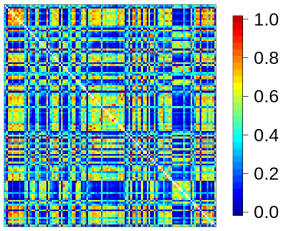
(g) Histogram of the Gelman–Rubin statistics for all $\tilde{\sigma}_i$'s and γ.

Figure S.12: The convergence of five randomly initialized Markov chains.

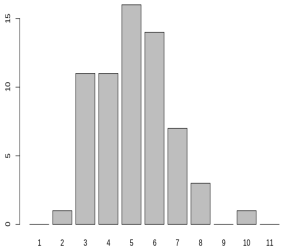
S 4.5 Additional Details on the Multi-view Clustering in the Alzheimer's Disease Study



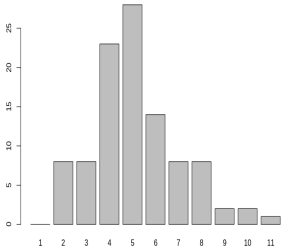
(a) The average $\Pr(c_i = c_j \mid y)$ for the ROIs in the healthy group.



(b) The average $\Pr(c_i = c_j \mid y)$ for the ROIs in the diseased group.



(c) Frequency plot of the number of clusters of ROIs for each subject in the healthy group.



(d) Frequency plot of the number of clusters for each subject in the diseased group.

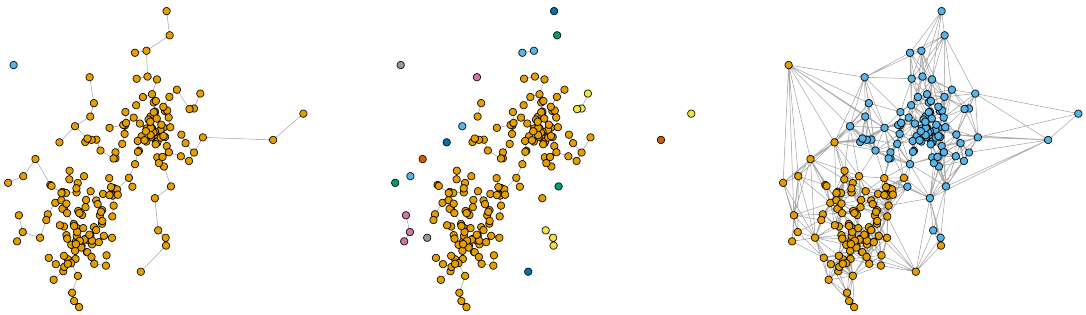
Figure S.13: Clustering estimates for the healthy and diseased group.

S 4.6 Comparison with Minimum Spanning Tree-based Cut

Since our Bayesian forest model uses spanning trees, it is natural to compare with the clustering algorithm based on cutting the minimum spanning tree (MST). To formalize, the minimum spanning tree-based cut (MST-Cut) algorithm first finds the MST: $\hat{T} =$

$\arg\min_{T\in\mathbb{T}_n} \sum_{(i,j)\in T} \|y_i - y_j\|$, where $\mathbb{T}_n$ denotes all spanning trees that connect n nodes, with $\|y_i - y_j\|$ as some distance between the two points. Then, one removes the longest $(K-1)$ edges (with length defined as $\|y_i - y_j\|$) to create K clusters. This algorithm is shown to be equivalent to the single-linkage clustering algorithm (Hartigan, 1981).

As we could imagine, such MST-Cut algorithms work well when clusters are well separated. In that case, those clusters will more likely be connected by the longest few edges. However, such algorithms will suffer sensitivity issues, when any one or more of the following happens: 1) when clusters are close to each other; 2) when a few isolated points are lying between two clusters; 3) when one or more clusters are from a heavy-tailed distribution, with a few points away from the bulk of the cluster. As a result, the longest edges in the MST may not be ideal for partitioning data.



(a) Partitioning the data by cutting the longest edge in the minimum spanning tree.

(b) Partitioning the data by cutting the top 10% longest edges in the minimum spanning tree.

(c) Partitioning the data using the Bayesian spanning forest model.

Figure 14: Comparing point estimates from the minimum spanning tree-based cut (MST-Cut) algorithms and the Bayesian spanning forest model.

To illustrate this problem, we use a simulation with data from a two-component t-distribution in $\mathbb{R}^2$ with 3 degrees of freedom. One component has the mean $(0, 0)$ and

the other has $(4, 4)$, and both have the scale parameter equal to I_2. And we generate $n = 200$ data points. As shown in Figure 14(a), due to the heavy tail and closeness of the two clusters, cutting the longest edge in the MST (using Euclidean distances) yields a trivial and sub-optimal partition. Further, cutting the top 10% longest edges still does not produce the desired result (Panel b).

Fundamentally, the reason is that relying on the minimum spanning tree (that is, one tree) leads to an underestimation of the graph uncertainty. Different from the MST-Cut algorithms, the Bayesian forest model effectively uses the marginal distribution [(3) in the main text] incorporating the multiplicity of those likely trees (with edges shown in Panel c). As the result, it leads to better performance than the MST-Cut algorithm.

S 4.7 Empirical Evidence for the Fast Convergence of Eigenvectors

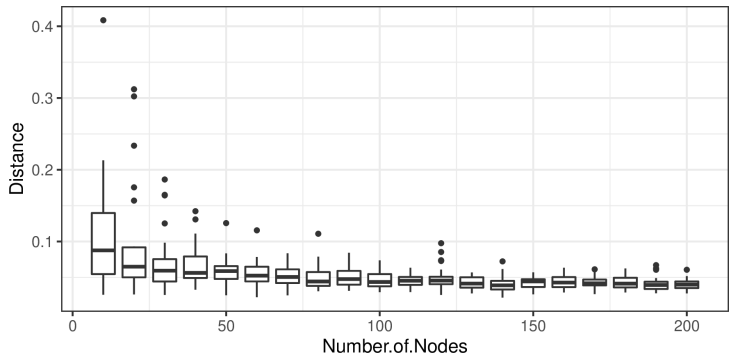


Figure S.15: The difference between eigenvectors converges to zero rapidly as n increases.

We now use simulations to illustrate the closeness between the K leading eigenvectors of the marginal connecting probability matrix M and the ones of the normalized Laplacian N. It is important to clarify that such closeness does not depend on how the data are generated.

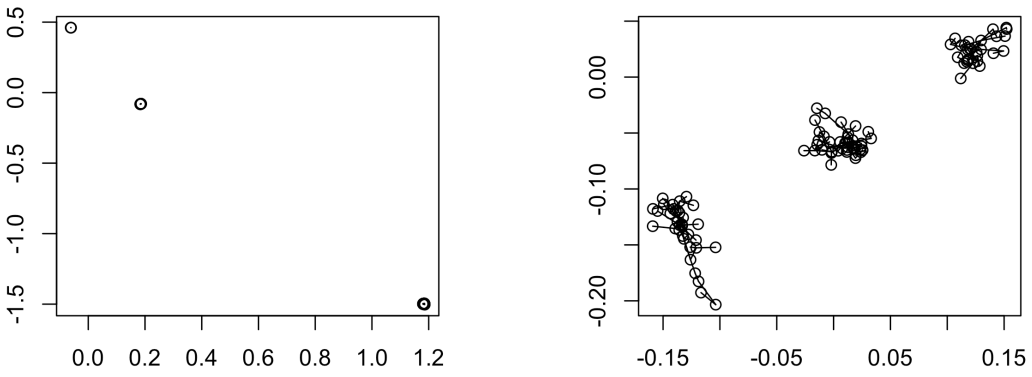
Therefore, for simplicity, we generate y_i from a simple three-component Gaussian mixture in $\mathbb{R}^2$ with means in $(0,0),(2,2),(4,4)$ and all variances equal to I_2, then we fit our forest model, and estimate $\sigma_{i,j}$'s using posterior mean. Based on the posterior mean of $\sigma_{i,j}$, we compute M and N, and then compute distances between their leading eigenvectors $\min_{R:RR'=I_K} \|\Psi_{1:K} - \Phi_{1:K}R\|$. We conduct such experiments under different sample sizes n ranging from 10 to 200; for each n, we repeat experiments for 30 times. As our theory requires a spectral gap $\xi_K - \xi_{K+1}$ not too close to zero, we choose to compare the top $K = 5$ eigenvectors. As shown in the boxplot of Figure S.4, the distance between two sets of eigenvectors quickly drops to near zero, for $n \geq 50$.

S 4.8 Illustration on Forest Process Realizations

In Figure S.16, we plot two realizations of forest process in $\mathbb{R}^2$, based on isotropic Gaussian for f and Cauchy for r, and a ground-up construction of $\mathcal{T}$ with

$$\pi_j^{[i]} = \frac{1}{i-1+\alpha} \text{for } j = 1,\ldots,(i-1), \qquad \pi_i^{[i]} = \frac{\alpha}{i-1+\alpha},$$

at $\alpha = 0.5$ and for $i = 1,\ldots,100$. In the first realization, we draw the scale parameters γ and $\sigma_{i,j}$ from the shrinkage priors used in Section 2.4 of the main text. Since $\sigma_{i,j}$ is very close to zero, the generated data appear close to three point masses. To illustrate a forest structure, in the second realization, we use fixed $\gamma = 0.1^2$ and $\sigma_{i,j} = 0.01^2$, and we can see three clusters, where each is connected by a tree.



(a) Data simulated from forest process using $\sigma_{i,j}$ and γ from the shrinkage priors in Section 2.4.

(b) Data simulated from forest process using $\sigma_{i,j} = 0.01^2$ and $\gamma = 0.1^2$.

Figure S.16: Illustration of two forest process realizations.

S 5 Alternative Model for the Scale Parameters

As an alternative to specifying a prior on the scale parameter $\tilde{\sigma}_i$ in the leaf density, the heuristic of setting $\tilde{\sigma}_i$ to a low order statistic of $\{\|y_i - y_j\|_2\}_{j=1}^n$ is shown to enjoy a good empirical performance in spectral clustering (Zelnik-Manor and Perona, 2005). In this section, we extend this heuristic to a formal model-based solution.

To start, we first relate the small distances to the $\tilde{k}$-nearest neighbor density estimator. Loftsgaarden and Quesenberry (1965) show that for $y_1, \cdots, y_n$ iid from a distribution with probability density f, with a growing $\tilde{k} \to \infty$, $\tilde{k}/n \to 0$ as n increase, if f is continuous at y_i, the $\tilde{k}$-nearest neighbor density estimator $f_n(y_i) = \tilde{k}/\left[nV_{\tilde{k}}(y_i)\right]$ is consistent for estimating $f(y_i)$, where $V_{\tilde{k}}(y_i)$ is the volume of the ball centered at y_i and with radius equal to the

distance to the $\tilde{k}$-th nearest neighbor, denoted by $d_i^{(\tilde{k})}$ from now on.

Although we no longer consider data as iid, $d_i^{(\tilde{k})}$ is still informative on how dense the data points are near y_i. Therefore, to bring information from $d_i^{(\tilde{k})}$ into the spanning forest model-based clustering, we consider a generative model that simultaneously depends on a spanning forest (with K component trees) and a $\tilde{k}$-nearest neighbor graph $\tilde{G}_{nn}$. We can use a likelihood

$$\mathcal{L}(y; \tilde{G}_{nn}, \mathcal{T}, \theta) \propto \Big\{ \prod_{i=1}^{n} \frac{(\tilde{\sigma}_i)^{\alpha_\sigma}}{\Gamma(\alpha_\sigma)} [\frac{d_i^{(\tilde{k})}}{\sqrt{p}}]^{-\alpha_\sigma - 1} \exp\Big[- \frac{\tilde{\sigma}_i}{d_i^{(\tilde{k})}/\sqrt{p}} \Big] \Big\}$$
$$\cdot \prod_{k=1}^{K} \Big\{ r(y_{k^*}; \theta) \prod_{(i,j) \in T_k} (2\pi\sigma_{i,j})^{-p/2} \exp\left(- \frac{\|y_i - y_j\|_2^2}{2\tilde{\sigma}_i \tilde{\sigma}_j} \right) \Big\}.$$

And one can verify that each term on the right-hand side is integrable in y_i, even if $d_i^{(\tilde{k})} = \|y_i - y_j\|_2$ happened (that is, when $(i,j) \in T_k$ and j happened to be the $\tilde{k}$-nearest neighbor of i); therefore, the right-hand side forms a proper likelihood. We choose the inverse-gamma distribution for each $d^{(\tilde{k})}/\sqrt{p}$, as it leads to an equivalent Gamma$(\alpha_\sigma + 1, d_i^{(\tilde{k})}/\sqrt{p})$ distribution for $\tilde{\sigma}_i$ that produces a shrinkage effect on $\tilde{\sigma}_i$ (Brown and Griffin, 2010), and it enjoys closed-form Gibbs sampling update via the generalized inverse gaussian distribution. We test the above model using $\tilde{k} = \lceil n^{1/10} \rceil$, and $\alpha_\sigma = 1$ on all the examples presented in the article, and the results are quite similar to the ones shown in the main text.

S 6 Code

We provide the implementation in the R source code. The code is available on `https://github.com/royarkaprava/Bayesian_forest_clustering`

References

Aldous, D. J. (1990). The Random Walk Construction of Uniform Spanning Trees and Uniform Labelled Trees. *SIAM Journal on Discrete Mathematics 3*(4), 450–465.

Bergé, L., C. Bouveyron, and S. Girard (2012). HDclassif: An R Package for Model-Based Clustering and Discriminant Analysis of High-Dimensional Data. *Journal of Statistical Software 46*(6), 1–29.

Broder, A. Z. (1989). Generating Random Spanning Trees. In *Annual Symposium on Foundations of Computer Science*, Volume 89, pp. 442–447.

Brown, P. J. and J. E. Griffin (2010). Inference With Normal-Gamma Prior Distributions in Regression Problems. *Bayesian Analysis 5*(1), 171 – 188.

Cai, D., T. Campbell, and T. Broderick (2021). Finite Mixture Models Do Not Reliably Learn the Number of Components. In *International Conference on Machine Learning*, pp. 1158–1169. PMLR.

Gelman, A. and D. B. Rubin (1992). Inference From Iterative Simulation Using Multiple Sequences. *Statistical Science*, 457–472.

Georghiades, A. S., P. N. Belhumeur, and D. J. Kriegman (2001). From Few to Many: Illumination Cone Models for Face Recognition Under Variable Lighting and Pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*(6), 643–660.

Hartigan, J. A. (1981). Consistency of Single Linkage for High-Density Clusters. *Journal of the American Statistical Association 76*, 388–394.

Horst, A. M., A. P. Hill, and K. B. Gorman (2020). palmerpenguins: Palmer Archipelago (Antarctica) Penguin Data. R package version 0.1.0.

Karatzoglou, A., A. Smola, K. Hornik, and A. Zeileis (2004). Kernlab—An S4 Package for Kernel Methods in R. *Journal of Statistical Software 11*, 1–20.

Kuhn, H. W. (1955). The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly 2*(1-2), 83–97.

Loftsgaarden, D. O. and C. P. Quesenberry (1965). A Nonparametric Estimate of a Multivariate Density Function. *The Annals of Mathematical Statistics 36*(3), 1049–1051.

Mosbah, M. and N. Saheb (1999). Non-Uniform Random Spanning Trees on Weighted Graphs. *Theoretical Computer Science 218*(2), 263–271.

Müller, P., F. Quintana, and G. L. Rosner (2011). A Product Partition Model With Regression on Covariates. *Journal of Computational and Graphical Statistics 20*(1), 260–278.

Ord, K. (1975). Estimation Methods for Models of Spatial Interaction. *Journal of the American Statistical Association 70*(349), 120–126.

Pinelis, I. (2020). Exact Lower and Upper Bounds on the Incomplete Gamma Function. *arXiv preprint arXiv:2005.06384*.

Prim, R. C. (1957). Shortest Connection Networks and Some Generalizations. *The Bell System Technical Journal 36*(6), 1389–1401.

Ross, G. J. and D. Markwick (2018). dirichletprocess: An R Package for Fitting Complex Bayesian Nonparametric Models.

Scrucca, L., M. Fop, T. B. Murphy, and A. E. Raftery (2016). MCLUST 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models. *The R Journal 8*(1), 289.

Vidal, R. (2011). Subspace Clustering. *IEEE Signal Processing Magazine 28*(2), 52–68.

Wu, S., X. Feng, and W. Zhou (2014). Spectral Clustering of High-Dimensional Data Exploiting Sparse Representation Vectors. *Neurocomputing 135*, 229–239.

Yu, Y., T. Wang, and R. J. Samworth (2015). A Useful Variant of the Davis–Kahan Theorem for Statisticians. *Biometrika 102*(2), 315–323.

Zelnik-Manor, L. and P. Perona (2005). Self-Tuning Spectral Clustering. In *Advances in Neural Information Processing Systems*, Volume 17.