Submitted 7/20; Revised 8/21; Published 9/21

Bayesian Distance Clustering

Leo L Duan LI.DUAN@UFL.EDU
Department of Statistics
University of Florida
Gainesville, FL 32611, USA

David B Dunson DUNSON@DUKE.EDU
Department of Statistical Science
Duke University
Durham, NC 27708, USA

Editor: Ruslan Salakhutdinov

Abstract

Model-based clustering is widely used in a variety of application areas. However, fundamental concerns remain about robustness. In particular, results can be sensitive to the choice of kernel representing the within-cluster data density. Leveraging on properties of pairwise differences between data points, we propose a class of Bayesian distance clustering methods, which rely on modeling the likelihood of the pairwise distances in place of the original data. Although some information in the data is discarded, we gain substantial robustness to modeling assumptions. The proposed approach represents an appealing middle ground between distance- and model-based clustering, drawing advantages from each of these canonical approaches. We illustrate dramatic gains in the ability to infer clusters that are not well represented by the usual choices of kernel. A simulation study is included to assess performance relative to competitors, and we apply the approach to clustering of brain genome expression data.

Keywords: Distance-based clustering, Mixture model, Model-based clustering, Model misspecification, Pairwise distance matrix, Partial likelihood

1. Introduction

Clustering is a primary focus of many statistical analyses, providing a valuable tool for exploratory data analysis and simplification of complex data. In the literature, there are two primary approaches – distance- and model-based clustering. Let $y_i \in \mathcal{Y}$, for $i = 1, \ldots, n$, denote the data and let $d(y, y')$ denote a distance between data points y and y'. Then, distance-based clustering algorithms are typically applied to the $n \times n$ matrix of pairwise distances $D_{(n)\times(n)} = \{d_{i,j}\}$, with $d_{i,j} = d(y_i, y_j)$ for all i, j pairs and $(n) = \{1, \ldots, n\}$. For reviews, see Jain (2010); Xu and Tian (2015). In contrast, model-based clustering takes a likelihood-based approach in building a model for the original data $y_{(n)}$ that has the form:

$$y_i \overset{iid}{\sim} f, \quad f(y) = \sum_{h=1}^{k} \pi_h \mathcal{K}(y; \theta_h), \tag{1}$$

where $\pi = (\pi_1, \ldots, \pi_k)'$ is a vector of probability weights in a finite mixture model, h is a cluster index, and $\mathcal{K}(y; \theta_h)$ is the density of the data within cluster h. Typically, $\mathcal{K}(y; \theta)$ is a density in a parametric family, such as the Gaussian, with θ denoting the parameters. The finite mixture model (1) can be obtained by marginalizing out the cluster index $c_i \in \{1, \ldots, k\}$ in the following model:

$$y_i \sim \mathcal{K}(\theta_{c_i}), \quad \text{pr}(c_i = h) = \pi_h. \tag{2}$$

Using this data-augmented form, one can obtain maximum likelihood estimates of the model parameters π and $\theta = \{\theta_h\}$ via an expectation-maximization algorithm (Fraley and Raftery, 2002). Alternatively, Bayesian methods are widely used to include prior information and characterize uncertainty in the parameters. For reviews, see Bouveyron and Brunet-Saumard (2014) and McNicholas (2016).

Distance-based algorithms tend to have the advantage of being relatively simple conceptually and computationally, while a key concern is the lack of characterization of uncertainty in clustering estimates and associated inferences. While model-based methods can address these concerns by exploiting a likelihood-based framework, a key disadvantage is a large sensitivity to the choice of kernel $\mathcal{K}(\cdot;\theta)$. Often, kernels are chosen for simplicity and computational convenience, and they place rigid assumptions on the shape of the clusters, which are not justified by the applied setting being considered.

We are not the first to recognize this problem, and there is literature attempting to address issues with kernel robustness in model-based clustering. One direction is to choose a flexible class of kernels, which can characterize a wide variety of densities. For example, one can replace the Gaussian kernel with one that accommodates asymmetry, skewness and/or heavier tails (Karlis and Santourian (2009); Juárez and Steel (2010); O'Hagan et al. (2016); Gallaugher and McNicholas (2018); among others). A related direction is to nonparametrically estimate the kernels specific to each cluster, while placing minimal constraints for identifiability, such as unimodality and sufficiently light tails. This direction is related to the mode-based clustering algorithms of Li et al. (2007); see also Rodríguez and Walker (2014) for a Bayesian approach using unimodal kernels. Unfortunately, as discussed by Hennig et al. (2015), a kernel that is too flexible leads to ambiguity in defining a cluster and identifiability issues: for example, one cluster can be the union of several clusters that are close. Practically, such flexible kernels demand a large number of parameters, leading to daunting computation cost.

A promising new strategy is to replace the likelihood with a robust alternative. Coretto and Hennig (2016) propose a pseudo-likelihood based approach for robust multivariate clustering, which captures outliers with an extra improper uniform component. Miller and Dunson (2018) propose a coarsened Bayes approach for robustifying Bayesian inference and apply it to clustering problems. Instead of assuming that the observed data are exactly generated from (1) in defining a Bayesian approach, they condition on the event that the empirical probability mass function of the observed data is within some small neighborhood of that for the assumed model. Both of these methods aim to allow small deviations from a simple kernel. It is difficult to extend these approaches to data with high complexity, such as clustering multiple time series, images, etc.

We propose a new approach based on a Bayesian model for the pairwise distances, avoiding a complete specification of the likelihood function for the data $y_{(n)}$. There is a rich literature proposing Bayesian approaches that replace an exact likelihood function with some alternative. Chernozhukov and Hong (2003) consider a broad class of such quasi-posterior distributions. Jeffreys (1961) proposed a substitution likelihood for quantiles for use in Bayesian inference; also refer to Dunson and Taylor (2005). Hoff (2007) proposed a Bayesian approach to inference in copula models, which avoids specifying models for the marginal distributions via an extended rank likelihood.

Johnson (2005) proposed Bayesian tests based on modeling frequentist test statistics instead of the data directly. These are just some of many examples.

Our proposed Bayesian distance clustering approach gains some of the advantages of model-based clustering, such as uncertainty quantification and flexibility, while significantly simplifying the model specification task. There is a connection between our approach and nonnegative matrix factorization (NMF) methods (Kuang et al., 2012; Zhao et al., 2015; Kuang et al., 2015). Certain NMF algorithms can be viewed as fast approximations to our likelihood-based approach. Our major contributions are: (i) establishing a novel link between model- and distance-based frameworks, (ii) introducing a principled choice for assigning kernels for distances (equivalent to the affinity/similarity score in NMFs), and (iii) providing a way to calibrate the parameters within the proposed probabilistic framework.

2. Partial likelihood for distances

2.1 Motivation for partial likelihood

Suppose that data $y_{(n)}$ are generated from model (1) or equivalently (2). We focus on the case in which $y_i = (y_{i,1}, \dots, y_{i,p})' \in \mathcal{Y} \subset \mathbb{R}^p$. The conditional likelihood of the data $y_{(n)}$ given clustering indices $c_{(n)}$ can be expressed as

$$L(y_{(n)}; c_{(n)}) = \prod_{h=1}^{k} \prod_{i:c_i=h} \mathcal{K}_h(y_i) = \prod_{h=1}^{k} L_h(y^{[h]}), \tag{3}$$

where we let $\mathcal{K}_h(y)$ denote the density of data within cluster h, and $y^{[h]} = \{y_i : c_i = h\} = \{y_i^{[h]}, i = 1, \dots, n_h\}$ is the data in cluster h. Since the information of $c_{(n)}$ is stored by the index with $[h]$, we will omit $c_{(n)}$ in the notation when $[h]$ appears. Referring to $y_1^{[h]}$ as the *seed* for cluster h, we can express the likelihood $L_h(y^{[h]})$ using the change-of-variables $(y_1^{[h]}, y_2^{[h]} \dots, y_{n_h}^{[h]}) \to (y_1^{[h]}, \tilde{d}_{2,1}^{[h]}, \dots, \tilde{d}_{n_h,1}^{[h]})$:

$$\begin{aligned} &\mathcal{K}_h(y_1^{[h]}) \prod_{i=2}^{n_h} G_h(\tilde{d}_{i,1}^{[h]} \mid y_1^{[h]}) \\ &= \mathcal{K}_h\big(y_1^{[h]} \mid \tilde{d}_{2,1}^{[h]}, \dots, \tilde{d}_{n_h,1}^{[h]}\big)\; G_h\big(\tilde{d}_{2,1}^{[h]}, \dots, \tilde{d}_{n_h,1}^{[h]}\big), \end{aligned} \tag{4}$$

where $\tilde{d}_{i,1}^{[h]} = y_i^{[h]} - y_1^{[h]}$ denotes the difference vector between $y_i^{[h]}$ and $y_1^{[h]}$ (with G the transformed kernel), and the second line is an equivalent factorization of the joint distribution, with $\mathcal{K}_h(y_1^{[h]} \mid .)$ the conditional density of $y_1^{[h]}$ given the differences. Expression (4) is a product of the densities of the seed and $(n_h - 1)$ differences. As the cluster size n_h increases, the relative contribution of the seed density $\mathcal{K}_h(y_1^{[h]} \mid .)$ will decrease and the likelihood becomes dominated by G_h. With this heuristic justification, we discard the $\mathcal{K}_h(y_1^{[h]} \mid .)$ term by treating the value of $y_1^{[h]}$ as random and integrating out $\mathcal{K}_h(y_1^{[h]} \mid .)$.

We now use a toy example to illustrate how to *derive* the function G_h from a known model-based likelihood (later we will show how to *specify* G_h directly, when the model-based likelihood is not known). Consider the case of $y_i \in \mathbb{R}$ from a Gaussian mixture, starting from the likelihood of those y_i's associated with $c_i = h$:

$$
\begin{aligned}
L_h(y^{[h]}) &= (2\pi\sigma_h^2)^{-n_h/2} \exp\left[-\frac{\sum_{i=1}^{n_h} (y_i^{[h]} - \mu_h)^2}{2\sigma_h^2} \right] \\
&= (2\pi\sigma_h^2)^{-n_h/2} \exp\left\{ -\frac{\sum_{i=1}^{n_h} \left[(y_i^{[h]} - y_1^{[h]})^2 + (y_1^{[h]} - \mu_h)^2 + 2(y_i^{[h]} - y_1^{[h]})(y_1^{[h]} - \mu_h) \right]}{2\sigma_h^2} \right\},
\end{aligned}
$$

To obtain G_h, based on the formula $f(d,y) = f(y \mid d) \int f(d,y)\mathrm{d}y$, we use the change-of-variable $\tilde{d}_{i,1}^{[h]} = y_i^{[h]} - y_1^{[h]}$, and integrate out $y_1^{[h]}$ [as the information of $y_1^{[h]}$ is now in $\mathcal{K}_h\big(y_1^{[h]} \mid \tilde{d}_{2,1}^{[h]}, \ldots, \tilde{d}_{n_h,1}^{[h]}\big)$]:

$$
\begin{aligned}
& G_h\big(\tilde{d}_{2,1}^{[h]}, \ldots, \tilde{d}_{n_h,1}^{[h]}\big) = \\
& \int (2\pi\sigma_h^2)^{-n_h/2} \exp\left[-\frac{\sum_{i=1}^{n_h} \tilde{d}_{i,1}^{[h]2} + n_h(y_1^{[h]} - \mu_h)^2 + 2(y_1^{[h]} - \mu_h)\sum_{i=1}^{n_h} \tilde{d}_{i,1}^{[h]}}{2\sigma_h^2} \right] \mathrm{d}y_1^{[h]} \\
& = (2\pi\sigma_h^2)^{-(n_h-1)/2} \frac{1}{\sqrt{n_h}} \exp\left[-\frac{\sum_{i=1}^{n_h} \tilde{d}_{i,1}^{[h]2} - (\sum d_{i,1}^{[h]})^2/n_h}{2\sigma_h^2} \right] \\
& \overset{(a)}{=} (2\pi\sigma_h^2)^{-(n_h-1)/2} \frac{1}{\sqrt{n_h}} \prod_{i,j} \exp\left(-\frac{\tilde{d}_{i,j}^{[h]2}}{4 n_h \sigma_h^2} \right),
\end{aligned}
$$

where (a) is due to $2\sum_{i<j} \tilde{d}_{ij}^{[h]2} = \sum_i \sum_j (\tilde{d}_{i,1}^{[h]} - \tilde{d}_{j,1}^{[h]})^2 = n_h \sum_i \tilde{d}_{i,1}^{[h]2} + n_h \sum_j \tilde{d}_{j,1}^{[h]2} - \sum_{j \neq i} 2\tilde{d}_{i,1}^{[h]}\tilde{d}_{j,1}^{[h]} - \sum_{i=j} 2\tilde{d}_{i,1}^{[h]}\tilde{d}_{j,1}^{[h]} = 2n_h[\sum \tilde{d}_{i,1}^{[h]2} - (\sum \tilde{d}_{i,1}^{[h]})^2/n_h]$.

In the above example, note that the right hand side appears to be a product density of *all* the pairwise differences $\tilde{D}^{[h]} = \{\tilde{d}_{i,j}^{[h]}\}_{(i,j)}$, besides those formed with the seed. This is due to the linear equality that $\tilde{d}_{i,j}^{[h]} = \tilde{d}_{i,1}^{[h]} - \tilde{d}_{j,1}^{[h]}$ — therefore, there are effectively only $(n_h - 1)$ free random variables; once they are given, the rest are completely determined.

Generalizing from this form, we now specify G as

$$
G_h\big(\tilde{d}_{2,1}^{[h]}, \ldots, \tilde{d}_{n_h,1}^{[h]}\big) = \prod_{i=1}^{n_h} \prod_{j>i} g_h^{1/n_h}(\tilde{d}_{i,j}^{[h]}), \tag{5}
$$

where $g_h : \mathbb{R}^p \to \mathbb{R}_+$ and each $\tilde{d}_{i,j}^{[h]}$ is assigned a marginal density. The power $1/n_h$ is a calibration parameter that adjusts the order discrepancy between the numbers of $(n_h - 1)n_h/2$ marginal densities and $(n_h - 1)$ effective random variables. We will formally justify this calibration in the theory section.

Remark 1 *To clarify, despite the simple product form,* (5) *should not be treated as the independent densities of* $n_h(n_h - 1)/2$ *differences. This is because these differences contain effectively* $(n_h - 1)$ *random variables* $\tilde{d}_{2,1}^{[h]}, \ldots, \tilde{d}_{n_h,1}^{[h]}$*, and* $(n_h - 1)(n_h - 2)/2$ *interaction terms* $\tilde{d}_{i,j}^{[h]} = \tilde{d}_{i,1}^{[h]} - \tilde{d}_{j,1}^{[h]}$*; these* *interaction terms induce dependence between $\tilde{d}_{i,1}^{[h]}$ and $\tilde{d}_{j,1}^{[h]}$:*

$$G_h\big(\tilde{d}_{2,1}^{[h]}, \ldots, \tilde{d}_{n_h,1}^{[h]}\big) = \bigg[\prod_{i=2}^{n_h} g_h^{1/n_h}(\tilde{d}_{i,1}^{[h]})\bigg]\bigg[\prod_{i=1}^{n_h-1}\prod_{i<j} g_h^{1/n_h}(\tilde{d}_{i,1}^{[h]} - \tilde{d}_{j,1}^{[h]})\bigg].$$

For example, for $\tilde{d}_{2,1}^{[h]}$ and $\tilde{d}_{3,1}^{[h]}$, the related terms in (5) *are:*

$$g^{1/n_h}(\tilde{d}_{2,1}^{[h]})g^{1/n_h}(\tilde{d}_{3,1}^{[h]})g^{1/n_h}(\tilde{d}_{2,1}^{[h]} - \tilde{d}_{3,1}^{[h]}),$$

which is a non-separable function of $\tilde{d}_{2,1}^{[h]}$ and $\tilde{d}_{3,1}^{[h]}$, and hence not independent.

We now state the assumptions that we use for clustering.

Assumption 1 *For those data within a cluster, $y_i^{[h]}$ and $y_j^{[h]}$ are independent and identically distributed.*

Focusing on marginally specifying each g_h, we can immediately obtain two key properties of $\tilde{d}_{i,j}^{[h]} = y_i^{[h]} - y_j^{[h]}$: (1) Expectation zero, and (2) Marginal symmetry with skewness zero. Hence, the distribution of the differences is substantially simpler than the original data distribution $\mathcal{K}_h$. This suggests using G_h for clustering will substantially reduce the model complexity and improve robustness.

We connect the density of the differences to a likelihood of 'distances' — here used as a loose notion including metrics, semi-metrics and divergences. Consider $d_{i,j} \in [0, \infty)$ as a transform of $\tilde{d}_{i,j}$, such as some norm $d_{i,j} = \|\tilde{d}_{i,j}\|$ (e.g. Euclidean or 1-norm); hence, a likelihood for $d_{i,j}$ is implicitly associated to a pushforward measure from the one on $\tilde{d}_{i,j}$ (assuming a measurable transform). For example, an exponential density on $d_{i,j} = \|\tilde{d}_{i,j}\|_1$ can be taken as the result of assigning a multivariate Laplace on $\tilde{d}_{i,j}$. We can further generalize the notion of difference from subtraction to other types, such as ratio, cross-entropy, or an application-driven specification (Izakian et al., 2015).

To summarize, this motivates the practice of first calculating a matrix of pairwise distances, and then assigning a partial likelihood for clustering. For generality, we slightly abuse notation and replace the difference array $\tilde{D}$ with the distance matrix D in (5), and denote the density by $G_h(D^{[h]})$. We will refer to (5) as the *distance likelihood* from now on. Conditional on the clustering labels,

$$L[D; c_{(n)}] = \prod_{h=1}^{k} G_h(D^{[h]}), \tag{6}$$

with $c_i \sim \sum_{h=1}^{k} \pi_h \delta_h$ independently, as is (2).

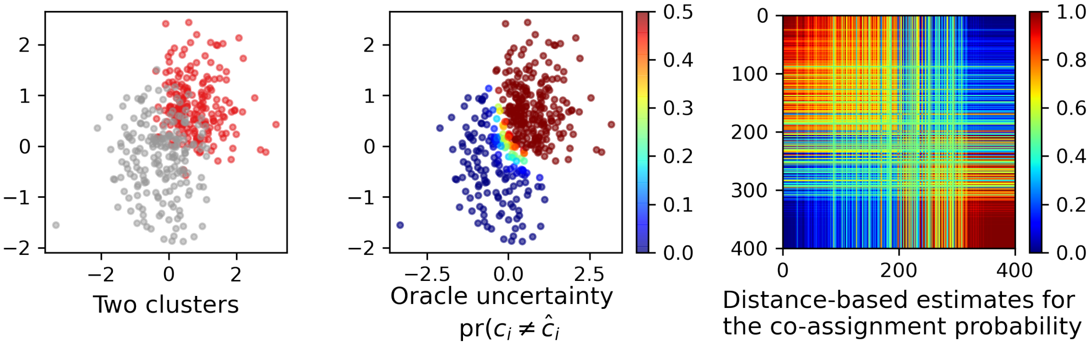


Figure 1: Illustration of the clustering uncertainty and its estimation using the distance-based clustering method. Left panel: two overlapping clusters (red and grey) generated from two skew Gaussian distributions with $n = 400$. Center panel: the oracle uncertainty $\text{pr}(c_i \neq \hat{c}_i \mid y_i)$ calculated based on the generative distribution. Right panel: the matrix of the co-assignment probabilities $\text{pr}(c_i = c_j \mid D)$ estimated using the distance likelihood of D.

To provide some intuition about the clustering uncertainty, we simulate two clusters using the bivariate skewed Gaussian distribution: in each dimension, the first cluster has a skewness of 4 (red in Figure 1, left panel), location of 0 and scale of 1, and the second has a skewness of -2, location of 0.5 and scale of 1 (grey in Figure 1, left panel); the two sub-coordinates are generated independently. Via the skew Gaussian density $\mathcal{K}_h(y_i)$ used to generate the data, we can compute the oracle assignment probability $\text{pr}(c_i = h \mid y_i)$ for $h = 1$ and 2, and the most likely cluster assignment $\hat{c}_i$ for each data point.

Clearly, due to the overlapping of the two clusters, there is a significant amount of uncertainty for those near the location $(0, 0)$, as the $\text{pr}(c_i \neq \hat{c}_i)$ remains away from 0 (Figure 1, center panel) — importantly, such uncertainty does not vanish even as $n \to \infty$, as these points will remain nearly equidistant to the two cluster centers. Using the distance likelihood on D, we can obtain an easy quantification of the uncertainty, by sampling c_i from the posterior distribution and calculating the co-assignment probabilities $\text{pr}(c_i = c_j \mid D)$; as shown in the right panel, the off-diagonal block shows that a significant portion of data that can be co-assigned to either the first or the second cluster with a non-trivial probability.

2.2 Choosing a distance density for clustering

To implement our Bayesian distance clustering approach, we need a definition of clusters, guiding us to choose a parametric form for $g_h(.)$ in (5). For conciseness, we will focus on the norm-based distances from now on. A popular intuition for a cluster is a group of data points, such that most of the distances among them are relatively small. That is, the probability of finding large distances within a cluster should be low. We now state the assumption.

Assumption 2 *With $\sigma_h > 0$, a scale parameter and ϵ_h a function that rapidly declines towards 0 as t increases.*

$$\text{pr}(d_{i,j}^{[h]} \geq t\sigma_h) \leq \epsilon_h(t) \quad \textit{for sufficiently large } t > 0. \tag{7}$$

For such a decline, it is common to consider the sub-exponential rate (Wainwright, 2019), $\epsilon_h(t) = \mathcal{O}\{\exp(-t/b)\}$ with some constant $b > 0$. Ideally, we want σ_h to be small, so that most of the distances within a cluster are small.

It is tempting to consider using a simple exponential density Exp(σ_h) for modeling $d^{[h]}_{i,j}$, however, we make an important observation here: the exponential distribution has a deterministic relationship between the mean σ_h and the variance σ^2_h — this means any slightly large $\mathbb{E}d^{[h]}_{i,j}$ (such as when the distribution of $d^{[h]}_{i,j}$ does not follow a exponential decay near zero) will inflate the estimate of σ_h, making it difficult to use small distances for clustering.

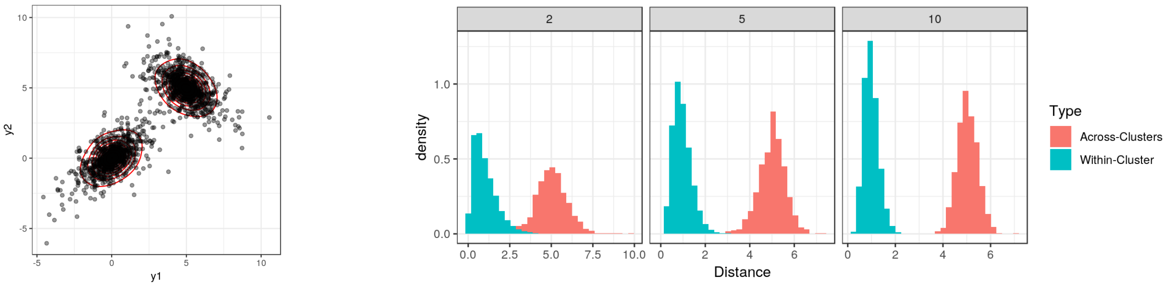


Figure 2: Histograms of Euclidean distances scaled by $1/\sigma_h$ (with $\sigma_h \approx \sqrt{p}$). Left is the first two dimensions, and the right show that the distances formed within a cluster (cyan) tend to be much smaller than the ones across clusters (red). Each cluster's data are generated from a multivariate Laplace distribution $y_i \sim$ Lap(μ_h, Σ_h) with $h = 1, 2$.

This motivates us to use a two-parameter distribution instead — in this article, we use Gamma (α_h, σ_h) for g_h in (5), whose variance $\alpha_h\sigma^2_h$ is no longer completely determined by the mean $\alpha_h\sigma_h$.

$$g_h(d^{[h]}_{i,j}) = \frac{1}{\Gamma(\alpha_h)\sigma_h^{\alpha_h}} x^{\alpha_h - 1} \exp\left(-d^{[h]}_{i,j}/\sigma_h\right). \tag{8}$$

We defer the prior choice for α_h and σ_h to a later section. For now, we verify that the Gamma distribution does have a sub-exponential tail that satisfies Assumption 1.

Lemma 2 *(Bound on the right tail) If d has the density* (8)*, for any $\alpha_h \geq 1$ and $t > 0$,*

$$pr(d \geq t\sigma_h) \leq Mt^{\alpha_h} \exp(-t),$$

where $M = (\alpha_h)^{-\alpha_h} \exp(\alpha_h)$.

Remark 3 *The polynomial term t^{α_h} allows deviation from the exponential distribution at small t; its effect vanishes as t increases.*

The assumption on the distances are connected to some implicit assumptions on the data distribution $\mathcal{K}(y_i)$. As such a link varies with the specific form of the distance, we again focus on the vector norm of subtraction $d^{[h]}_{i,j} = \|y^{[h]}_i - y^{[h]}_j\|_q$, with $\|x\|_q = (\sum_{j=1}^p x_j^q)^{1/q}$ and $q \geq 1$. We now show that a sub-exponential tail for the vector norm distance is a necessary result of assuming sub-exponential tails in $\mathcal{K}(y_i)$.

Lemma 4 *(Tail of vector norm distance) If there exist bound constants $m_1^{[h]}, m_2^{[h]} > 0$, such that for all $j = 1, \ldots, p$*

$$pr(|y_{i,j}^{[h]} - \mathbb{E}y_{i,j}^{[h]}| \geq t) \leq m_1^{[h]} \exp(-m_2^{[h]} t), \tag{9}$$

then, there exist another two constants $\nu_h, b_h > 0$, such that for any $q \geq 1$

$$pr(d_{ij}^{[h]} > t b_h p^{\eta}) \leq 2p \exp\{-t p^{(\eta - 1/q)}/2\} \quad \text{for } t > 2p^{1/q-\eta}\nu_h^2. \tag{10}$$

Remark 5 *The concentration property* (9) *is less restrictive than common assumptions on the kernel in a mixture model, such as Gaussianity, log-concavity or unimodality.*

3. Hyper-prior specification for α_h and σ_h

In Bayesian clustering, it is useful to choose the prior parameters in a reasonable range (Malsiner-Walli et al., 2017). Recall in our gamma density, α_h determines the mean for $d_{i,j}^{[h]}$ at $\alpha_h\sigma_h$. To favor small values for the mode while accommodating a moderate degree of uncertainty, we use a Gamma prior $\alpha_h \sim \text{Gamma}(1.5, 1.0)$.

To select a prior for σ_h, we associate it with a pre-specified maximum cluster number k. We can view k as a packing number — that is, how many balls (clusters) we can fit in a container of the data. To formalize, imagine a p-dimensional ellipsoid in $\mathbb{R}^p$ enclosing all the observed data. The smallest volume of such an ellipsoid is

$$vol(\text{Data}) = M \min_{\mu \in \mathbb{R}^p, Q \succ 0} (\det Q)^{-1/2}, \text{ s.t. } (y_i - \mu)^{\mathrm{T}} Q (y_i - \mu) \leq 1 \text{ for } i = 1, \ldots, n,$$

which can be obtained via a fast convex optimization algorithm (Sun and Freund, 2004), with $M = \tilde{\pi}^{p/2}/\Gamma(p/2+1)$ and $\tilde{\pi} \approx 3.14$.

If we view each cluster as a high-probability ball of points originating from a common distribution, then the diameter — the distance between the two points that are farthest apart — is $\sim 4\sigma_h$. This is calculated based on $\text{pr}(d \leq 4\sigma_h) \approx 0.95$ using the gamma density with shape 1.5 (the prior mean of α_h). We denote the ball by $\mathcal{B}_{2\sigma_h}$, with $vol(\mathcal{B}_{2\sigma_h}) = M(2\sigma_h)^p$.

Setting k to the packing number

$$k \simeq \frac{vol(\text{Data})}{vol(\mathcal{B}_{2\sigma_h})}$$

yields a sensible prior mean for σ_h. For conjugacy, we choose an inverse-gamma prior for σ_h with $\mathbb{E}(\sigma_h) = \beta_h$,

$$\sigma_h \sim \text{Inverse-Gamma}(2, \beta_\sigma), \qquad \beta_\sigma = \frac{1}{2}\left\{\frac{vol(\text{Data})}{kM}\right\}^{1/p}.$$

The above prior can be used as a default in broad applications, and does not require tuning to each new application.

4. Theory

We describe several interesting properties for the distance likelihood.

4.1 Calibration

Lemma 6 (Exchangeability) *When the product density* (5) *is used for all $G_h(D^{[h]})$, $h = 1, \ldots, k$, the distance likelihood* (6) *is invariant to permutations of the indices i:*

$$L\{y_{(n)}; c_{(n)}\} = L\{y_{(n^*)}; c_{(n^*)}\},$$

with $(n^*) = \{1_*, \ldots, n_*\}$ denoting a set of permuted indices.

We fill a missing gap between the model-based and distance likelihoods by considering an information-theoretic analysis of the two clustering approaches. This also leads to a principled choice of the power $1/n_h$ in (5).

To quantify the information in clustering, we first briefly review the concept of Bregman divergence (Bregman, 1967). Letting $\phi : \mathcal{S} \to \mathbb{R}$ be a strictly convex and differentiable function, with $\mathcal{S}$ the domain of ϕ, the Bregman divergence is defined as

$$B_\phi(x, y) = \phi(x) - \phi(y) - (x - y)^{\mathrm{T}} \nabla\phi(y),$$

where $\nabla\phi(y)$ denotes the gradient of ϕ at y. A large family of loss functions, such as squared norm and Kullback-Leibler divergence, are special cases of the Bregman divergence with suitable ϕ. For model-based clustering, when the regular exponential family ('regular' as the parameter space is a non-empty open set) is used for the component kernel $\mathcal{K}_h$, Banerjee et al. (2005) show that there always exists a re-parameterization of the kernel using Bregman divergence. Using our notation,

$$\mathcal{K}_h(y_i; \theta_h) = \exp\left\{T(y_i)'\theta_h - \psi(\theta_h)\right\} \kappa(y_i) \Leftrightarrow \exp\left[-B_\phi\left\{T(y_i), \mu_h\right\}\right] b_\phi\{T(y_i)\},$$

where $T(y_i)$ is a transformation of y_i, in the same form as the minimum sufficient statistic for θ_h (except this 'statistic' is based on only one data point y_i); μ_h is the expectation of $T(y_i)$ taken with respect to $\mathcal{K}_h(y; \theta_h)$; ψ, κ and b_ϕ are functions mapping to $(0, \infty)$.

With this re-parameterization, maximizing the model-based likelihood over $c_{(n)}$ becomes equivalent to minimizing the within-cluster Bregman divergence

$$H_y = \sum_{h=1}^{k} H_y^{[h]}, \quad H_y^{[h]} = \sum_{i=1}^{n_h} B_\phi\left\{T(y_i^{[h]}), \mu_h\right\}.$$

We will refer to H_y as the model-based divergence.

For the distance likelihood, considering those distances that can be viewed or re-parameterized as a pairwise Bregman divergence, we assume each $g(d_{i,j}^{[h]})$ in the distance likelihood (5) can be re-written with a calibrating power $\beta_h > 0$ as

$$g^{\beta_h}(d_{i,j}^{[h]}) = z^{\beta_h} \exp\left[-\beta_h B_\phi\left\{T(y_i^{[h]}), T(y_j^{[h]})\right\}\right],$$

with $z > 0$ the normalizing constant. A distance-based divergence H_d can be computed as

$$H_d = \sum_{h=1}^{k} H_d^{[h]}, \quad H_d^{[h]} = \beta_h \sum_{i=1}^{n_h}\sum_{j=1}^{n_h} \frac{1}{2} B_\phi\left\{T(y_i^{[h]}), T(y_j^{[h]})\right\}. \tag{11}$$

We now compare these two divergences H_y and H_d at their expectations.

Lemma 7 *(Expected Bregman Divergence) The distance-based Bregman divergence* (11) *in cluster h has*

$$\mathbb{E}_{y^{[h]}} H_d^{[h]} = \beta_h \mathbb{E}_{y_i^{[h]}} \mathbb{E}_{y_j^{[h]}} \sum_{i=1}^{n_h} \sum_{j=1}^{n_h} \frac{1}{2} B_\phi\{T(y_i^{[h]}), T(y_j^{[h]})\}$$

$$= (n_h \beta_h)\, \mathbb{E}_{y^{[h]}} \left[\sum_{i=1}^{n_h} \frac{B_\phi\{T(y_i^{[h]}), \mu_h\} + B_\phi\{\mu_h, T(y_i^{[h]})\}}{2} \right],$$

where the expectation over $y^{[h]}$ is taken with respect to $\mathcal{K}_h$.

Remark 8 *The term inside the expectation on the right hand side is the symmetrized Bregman divergence between $T(y_i^{[h]})$ and μ_h (Banerjee et al., 2005). Therefore, $\mathbb{E}_{y^{[h]}} H_d^{[h]} = (n_h\beta_h)\mathbb{E}_{y^{[h]}} H_y^{[h]}$ when $B_\phi(\cdot,\cdot)$ is symmetric.*

There is an order difference of $\mathcal{O}(n_h)$ between distance-based and model-based divergences. Therefore, a sensible choice is simply setting $\beta_h = 1/n_h$. This power is related to the weights used in composite pairwise likelihood (Lindsay, 1988; Cox and Reid, 2004).

4.2 Relationship to Graph Cut

It is also interesting to consider the matrix form of the distance likelihood. We use C as an $n \times k$ binary matrix encoding the cluster assignment, with $C_{i,h} = 1$ if $c_i = h$, and all other $C_{i,h'} = 0$. Then it can be verified that $C^{\mathrm{T}}C = \mathrm{diag}(n_1, \ldots, n_k)$. Hence the distance likelihood, with the Gamma density, is

$$G(D; C) \propto \exp\left[\mathrm{tr}\left\{C^{\mathrm{T}}(\log D)C\Lambda(C^{\mathrm{T}}C)^{-1}\right\}\right] \exp\left[-\mathrm{tr}\left\{C^{\mathrm{T}}DC\left(\Sigma C^{\mathrm{T}}C\right)^{-1}\right\}\right], \tag{12}$$

where D is the $n \times n$ distance matrix, log is applied element-wise, $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_k)$, and $\Lambda = \mathrm{diag}(\alpha_1 - 1, \ldots, \alpha_h - 1)$. If C contains zero columns, the inverse is replaced by a generalized inverse.

One may notice some resemblance of (12) to the loss function in graph partitioning algorithms. Indeed, if we simplify the parameters to $\alpha_1 = \cdots = \alpha_k = \alpha_0$ and $\sigma_1 = \cdots = \sigma_k = \sigma_0$, then

$$G(D; C) \propto \exp\left[\mathrm{tr}\left\{C^{\mathrm{T}}AC\left(C^{\mathrm{T}}C\right)^{-1}\right\}\right], \tag{13}$$

where $A = \kappa \mathbf{1}_{n,n} - D/\sigma_0 + (\alpha_0 - 1)\log D$ can be considered as an adjacency matrix of a graph formed by a log-Gamma distance kernel, with $\mathbf{1}_{n,n}$ as an $n \times n$ matrix with all elements equal to 1; κ a constant so that each $A_{i,j} > 0$ (since κ enters the likelihood as a constant $\mathrm{tr}\{C^{\mathrm{T}}\kappa\mathbf{1}_{n,n}C(C^{\mathrm{T}}C)^{-1}\} = n\kappa$, it does not impact the likelihood of C). To compare, the popular normalized graph-cut loss (Bandeira et al., 2013) is

$$\text{NCut-Loss} = \sum_{h=1}^{k} \sum_{i:c_i=h} \sum_{j:c_j\neq h} \frac{A_{i,j}}{2n_h}, \tag{14}$$

which is the total edges deleted because of partitioning (weighted by n_h^{-1} to prevent trivial cuts). There is an interesting link between (13) and (14).

Lemma 9 *Considering a graph with weighted adjacency matrix A, the normalized graph-cut loss is related to the negative log-likelihood (omitting constant)* (13) *via*

$$2\text{NCut-Loss} = -tr\big\{C^{\mathrm{T}}AC\big(C^{\mathrm{T}}C\big)^{-1}\big\} + \sum_{i=1}^{n} \frac{\sum_{j=1}^{n} A_{i,j}}{n_{c_i}}.$$

Remark 10 *The difference on the right is often known as the degree-based regularization (with* $\sum_{j=1}^{n} A_{i,j}$ *the degree,* n_{c_i} *the size of the cluster that data* i *is assigned to). When the cluster sizes are relatively balanced, we can ignore its effect.*

Such a near-equivalence suggests that we can exploit popular graph clustering algorithms, such as spectral clustering, for good initiation of C as a warm-start of the Markov chain Monte Carlo algorithm.

5. Posterior computation

For the posterior computation, Gibbs sampler is easy to derive as it involves updating one element of the parameter at a time, from the full conditional distribution. However, this could lead to a heavy computational cost for our model. To understand this, consider the update step for each c_i, which involves a draw from the categorical distribution:

$$\text{pr}(c_i = h \mid .) = \frac{\pi_h G(D; \tilde{C}_{i,h})}{\sum_{h'=1}^{k} \pi_{h'} G(D; \tilde{C}_{i,h'})},$$

where $\tilde{C}_h$ denotes a matrix equal to the current value of C, except replacing the ith row with $C_{i,h} = 1$ and $C_{i,j} = 0$ for other $j \neq h$. Since $G(D; C)$ involves a matrix inverse term $(C^{\mathrm{T}}C)^{-1}$, the above ratio cannot be simplified to reduce the computational burden. The evaluation cost for each $G(D; C)$ is dominated by the matrix multiplication steps within, hence having an overall cost of $\mathcal{O}(n^2k)$. Therefore, iterating over $h = 1, \ldots, k$ and $i = 1, \ldots, n$ will lead to a high cost in one sweep of update.

To solve this problem, we instead develop a more efficient algorithm based on the approximate Hamiltonian Monte Carlo (HMC) algorithm. We use a continuous relaxation of each row C_i (on a simplex vertex) into the interior of the simplex, and denote the relaxation by $W_i \in \Delta_{\backslash\partial}^{(k-1)}$. This is achieved via a tempered softmax re-parameterization (Maddison et al., 2017)

$$w_{i,h} = \frac{\exp(v_{i,h}/t)}{\sum_{h'=1}^{k} \exp(v_{i,h'}/t)}, \quad h = 1, \ldots, k.$$

At small $t > 0$ and close to 0, if one $v_{i,h}$ is slightly larger than the rest in $\{v_{i,1}, \ldots, v_{i,k}\}$, then $w_{i,h}$ will be close to 1, and all the other $w_{i,h'}$'s close to 0. In this article, we use $t = 0.1$ as a balance between the approximation accuracy and the numeric stability of the algorithm. In addition, we re-parameterize the other parameters using the softplus function $\sigma_h = \log[\exp(\tilde{\sigma}_h) + 1]$, $\alpha_h = \log[\exp(\tilde{\alpha}_h) + 1]$ for $h = 1, \ldots, k$, and and the softmax function $(\pi_1, \ldots, \pi_k) = \text{softmax}(\tilde{\pi}_1, \ldots, \tilde{\pi}_k)$ (as defined above except with $t = 1$), where $\tilde{\sigma}_h$, $\tilde{\alpha}_h$ and $\tilde{\pi}_h$ are all unconstrained parameters in $\mathbb{R}$ amenable to the off-the-shelf continuous HMC algorithm.

We denote the vectorized parameters by $\beta = (v_{1,1}, \ldots, v_{n,k}, \tilde{\sigma}_1, \ldots, \tilde{\sigma}_k, \tilde{\alpha}_1, \ldots, \tilde{\alpha}_k, \tilde{\pi}_1, \ldots, \tilde{\pi}_k)$. To sample from posterior distribution $\beta \sim \Pi_{\beta|D}(\cdot)$, the HMC uses an auxiliary momentum variable

v and samples from a joint distribution $\Pi(\beta, v) = \Pi(\beta \mid D)\Pi(v)$, where a common choice of $\Pi(v)$ is the density of $\mathrm{N}(0, M)$. Denote $U(\beta) = -\log \Pi(\beta \mid D)$ and $K(v) = -\log \pi(v) = v^T M^{-1} v/2$, which are commonly referred to as the potential energy and kinetic energy respectively. The total Hamiltonian energy function is $H(\beta, v) = U(\beta) + K(v)$.

At each state (β, v), a new proposal (β^*, v^*) is generated by simulating Hamiltonian dynamics satisfying the Hamilton's equations:

$$\frac{\partial \beta}{\partial t} = \frac{\partial H(\beta, v)}{\partial v} = M^{-1}v; \quad \frac{\partial v}{\partial t} = -\frac{\partial H(\beta, v)}{\partial \beta} = \frac{\partial \log \Pi(\beta \mid D)}{\partial \beta}.$$

Since the exact solution to the above is intractable, we can numerically approximate the temporal evolution using the leapfrog scheme, as described in the following pseudocode.

```
for Iteration =$1, 2, \ldots$ do
    Sample $v \sim \mathrm{N}(0, M)$, set $\beta^* \leftarrow \beta$ and $v^* \leftarrow v$;
    for $l = 1, \ldots, L$ do
        Update $v^* \leftarrow v^* + \frac{\epsilon}{2}\frac{\partial \log \Pi(\beta^* \mid D)}{\partial \beta^*}$;
        Update $\beta^* \leftarrow \beta^* + \epsilon M^{-1} v^*$;
        Update $v^* \leftarrow v^* + \frac{\epsilon}{2}\frac{\partial \log \Pi(\beta^* \mid D)}{\partial \beta^*}$;
        if $(\beta^* - \beta)^{\mathrm{T}} v^* < 0$ then
            Break;
    Sample $u \sim \mathrm{Uniform}(0, 1)$;
    if $u < \min\{1, \exp[-H(\beta^*, -v^*) + H(\beta, v)]\}$. then
        Set $\beta \leftarrow \beta^*$;
```

Algorithm 1: The pseudocode of the No-U-Turn Hamiltonian Monte Carlo sampler for the Bayesian distance clustering.

To accelerate the convergence of the Markov chain to stationarity, we first use the BFGS optimization algorithm (implemented in the PyTorch package) to first minimize $U(\beta)$ and obtain the posterior mode $\hat{\beta}$. We then initialize the Markov chain at $\beta = \hat{\beta}$.

A typical choice for the working parameter M^{-1} is to let it roughly scale with the covariance matrix of the posterior distribution (Neal, 2011). Using $\hat{\beta}$, we calculate the observed Fisher information at $\hat{\beta}$ [the Hessian matrix of $U(\beta)$ evaluated at $\hat{\beta}$, denoted by $\mathrm{Hess}_U(\hat{\beta})$], which gives an approximation to the inverse covariance of β. Although it is tempting to set $M^{-1} = [\mathrm{Hess}_U(\hat{\beta})]^{-1}$, the matrix inversion of the latter is often costly and ill-conditioned. To avoid this problem, we use a simpler and diagonal parameterization $M^{-1} = \mathrm{diag}(1/\mathrm{Hess}_U(\hat{\beta})_{i,i})$, which shows good empirical performances in all the examples within this article.

To run the HMC sampler, we use the No-U-Turn Sampler (NUTS-HMC) algorithm (Hoffman and Gelman, 2014) implemented in the 'hamiltorch' package (Cobb and Jalaian, 2020), which also automatically tunes the other two working parameters ϵ and L. After the automatic tuning, the algorithm reaches an acceptance rate close to 70% as commonly desired for good mixing of the Markov chains. To provide some running time, using a quad-core i7 CPU, at $n = 1000$, the HMC algorithm takes about 20 minutes for running $10,000$ iterations.

Remark 11 *On the computational cost, the most expensive step in the HMC algorithm is the calculation of the derivative of* $\log G(D; W)$ *with respect to the matrix* W*, which involves the following* *form:*

$$\frac{\partial tr[(X^{\mathrm{T}}AX)(X^{\mathrm{T}}BX)^{-1}]}{\partial X} = 2AX(X^{\mathrm{T}}BX)^{-1} - 2BX(X^{\mathrm{T}}BX)^{-1}(X^{\mathrm{T}}AX)(X^{\mathrm{T}}BX)^{-1}$$

where $X \in \mathbb{R}^{n\times k}$, symmetric $B \in \mathbb{R}^{n\times n}$ and symmetric $A \in \mathbb{R}^{n\times n}$. Since k is relatively small, the matrix inversion of the $k\times k$ matrix is not costly [$\mathcal{O}(k^3)$] and dominated by the matrix multiplication $\mathcal{O}(n^2k)$. Therefore, running over L leapfrog steps, the computational cost per iteration of HMC is $\mathcal{O}(Ln^2k)$.

Potentially, one could instead consider a Gibbs sampling algorithm, using a block-wise update of $C^{\mathrm{T}}(\log D)C$ and $C^{\mathrm{T}}DC$ (instead of a full evaluation of the matrix product) when sampling each row of C. Despite having a similar computing complexity, a strength of HMC is that we can take advantage of the highly parallelized matrix operation on C, which is often faster than the sequential looping over each row of C.

In comparison, the parametric/model-based clustering algorithm has a lower cost of $O(n)$, although this often comes with a risk of model misspecification for modern data. Therefore, choosing which class of methods involves a trade-off between computational speed versus model robustness.

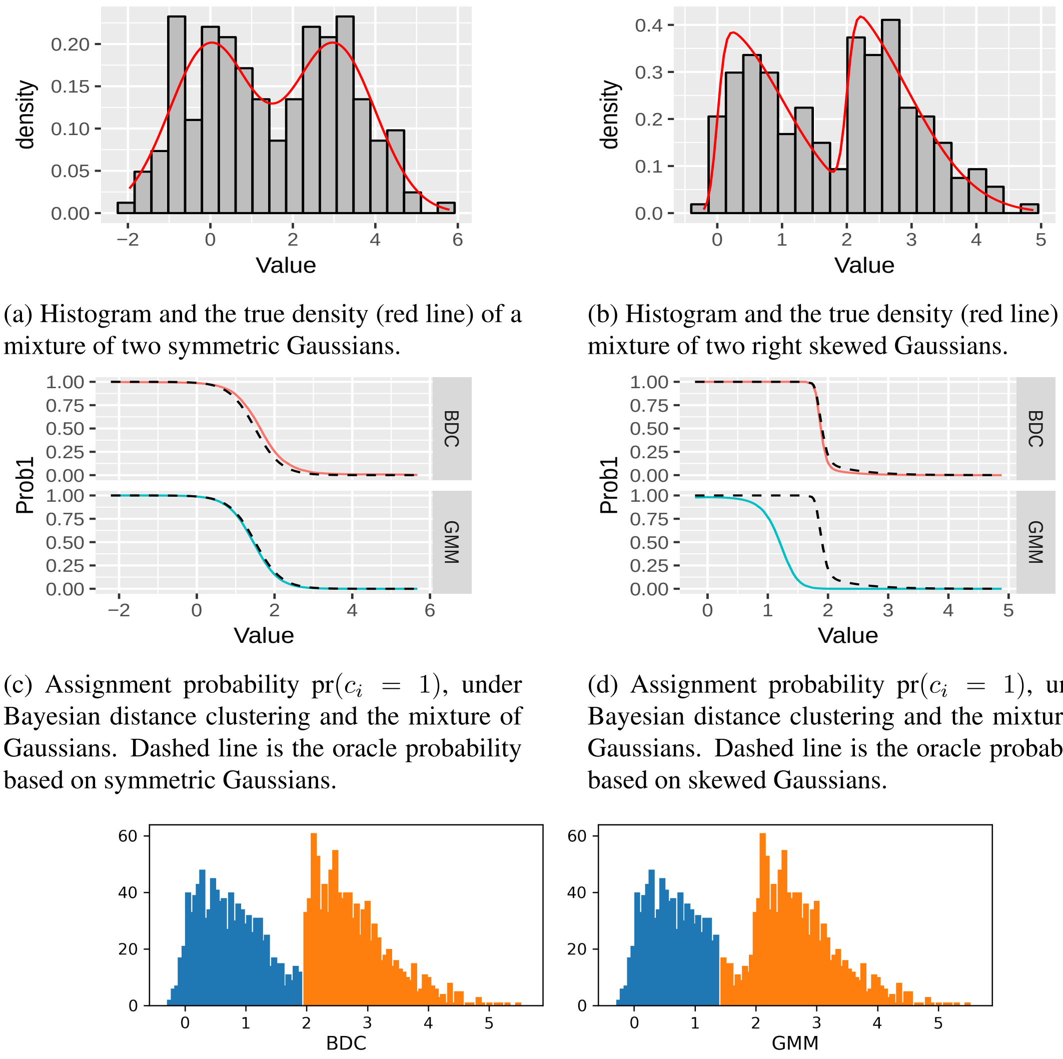
The posterior samples of CC^{T} give an estimate of the pairwise co-assignment probabilities $\mathrm{pr}(c_i = c_j) = \sum_{h=1}^{k} \mathrm{pr}(c_i = c_j = h)$. To obtain estimates for $\mathrm{pr}(c_i = h)$, we use symmetric simplex matrix factorization (Duan, 2020) on $\{\mathrm{pr}(c_i = c_j)\}_{i,j}$ to obtain an $n \times k$ matrix corresponding to $\{\mathrm{pr}(c_i = h)\}_{i,h}$. For the diagnostics on the convergence, we calculate the autocorrelation (ACF) and the effective sample size (ESS) for each parameter, and we provide some diagnostic plots in the appendix.

In this article, for the ease of visualization and interpretation, we use $\mathrm{pr}(c_i \neq \hat{c}_i \mid D)$ as a measure of the uncertainty on the point estimate $\hat{c}_i = \max_{h=1,\ldots,k} \mathrm{pr}(c_i = h \mid D)$. An alternative is to use the variation of information (Wade and Ghahramani, 2018) as a metric between the clusterings, leading to the discrete extension of the posterior mean and credible intervals. The readers can find the method and toolbox within the reference.

In addition, in the appendix, we show that the non-negative matrix factorization (NMF) algorithm, if using a calibrated similarity matrix as its input (such as using our distance likelihood), produces an almost indistinguishable result from the Bayesian distance clustering method. On the other hand, if the similarity is set less carefully (such as using the "default" choice in popular machine learning packages), we found a severe sensitivity that leads to over-/under-estimation of the uncertainty (as shown in Panel 4 of Figure 8). Therefore, for the sake of both conciseness and fairness, we choose to not present the NMF results without calibration in the numerical experiments.

6. Numerical experiments

6.1 Clustering with skewness-robust distance



(a) Histogram and the true density (red line) of a mixture of two symmetric Gaussians.

(b) Histogram and the true density (red line) of a mixture of two right skewed Gaussians.

(c) Assignment probability $\text{pr}(c_i = 1)$, under Bayesian distance clustering and the mixture of Gaussians. Dashed line is the oracle probability based on symmetric Gaussians.

(d) Assignment probability $\text{pr}(c_i = 1)$, under Bayesian distance clustering and the mixture of Gaussians. Dashed line is the oracle probability based on skewed Gaussians.

(e) The point estimates $\hat{c}_i$ (represented by two colors) using Bayesian distance clustering and the mixture of Gaussians.

Figure 3: Clustering data from a two-component mixture of skewed Gaussians in $\mathbb{R}$. Bayesian Distance clustering (BDC) gives posterior clustering probabilities close to the oracle probabilities regardless of whether the distribution is skewed or not (upper plots in panel c and d), while the mixture of Gaussians fails when the skewness is present (lower plot in panel d).

p	Bayes Dist. Clustering	Mix. of Gaussians	Mix. of Skewed Gaussians
1	0.80 (0.75, 0.84)	0.65 (0.55, 0.71)	0.81 (0.75, 0.85)
5	0.76 (0.71, 0.81)	0.55 (0.40, 0.61)	0.76 (0.72, 0.80)
10	0.72 (0.68, 0.76)	0.33(0.25, 0.46)	0.62 (0.53, 0.71)
30	0.71 (0.67, 0.76)	0.25 (0.20, 0.30)	0.43 (0.37, 0.50)

Table 1: Accuracy of clustering skewed Gaussians under different dimensions p. Adjusted Rand index (ARI) is computed for the point estimates using variation of information. The average and 95% confidence interval are shown.

As described in Section 2.1, the vector norm-based distance is automatically robust to skewness. To illustrate, we generate $n = 200$ data from a two-component mixture of skewed Gaussians:

$$\begin{aligned}&\text{pr}(c_i = 1) = \text{pr}(c_i = 2) = 0.5,\\ &y_{i,j} \mid c_i = h \sim \text{SN}(\mu_h, 1, \alpha_h) \text{ for } j = 1 \dots p,\end{aligned}$$

where $\text{SN}(\mu, \sigma, \alpha)$ has density $\pi(y \mid \mu, \sigma, \alpha) = 2f\{(y-\mu)/\sigma\}F\{\alpha(y-\mu)/\sigma\}$ with f and F the density and cumulative distribution functions for the standard Gaussian distribution.

We start with $p = 1$ and assess the performance of the Bayesian distance clustering model under both non-skewed ($\alpha_1 = \alpha_2 = 0, \mu_1 = 0, \mu_2 = 3$) and skewed distributions ($\alpha_1 = 8, \alpha_2 = 10, \mu_1 = 0, \mu_2 = 2$). The results are compared against the mixture of Gaussians as implemented in the *Mclust* package. Figure 3(a,c) show that for non-skewed Gaussians, the proposed approach produces clustering probabilities close to their oracle probabilities, obtained using knowledge of the true kernels that generated the data. When the true kernels are skewed Gaussians, Figure 3(b,d) shows that the mixture of Gaussians gives inaccurate estimates of the clustering probability, whereas Bayesian distance clustering remains similar to the oracle.

To evaluate the accuracy of the point estimate $\hat{c}_i$, we compute the adjusted Rand index (Rand, 1971) with respect to the true labels. We test under different $p \in \{1, 5, 10, 30\}$, and repeat each experiment 30 times. The results are compared against model-based clustering using symmetric and skewed Gaussians kernels, using independent variance structure. As shown in Table 1, the misspecified symmetric model deteriorates quickly as p increases. In contrast, Bayesian distance clustering maintains high clustering accuracy.

6.2 Clustering high dimensional data with subspace distance

For high-dimensional clustering, it is often useful to impose the additional assumption that each cluster lives near a different low-dimensional manifold. Clustering data based on these manifolds is known as subspace clustering. We exploit the sparse subspace embedding algorithm proposed by Vidal (2011) to learn pairwise subspace distances. Briefly speaking, since the data in the same cluster are alike, each data point can be approximated as a linear combination of several other data points in the same subspace; hence a sparse locally linear embedding can be used to estimate an $n \times n$ coefficient matrix $\hat{W}$ through

$$\hat{W} = \arg \min_{W: w_{i,i}=0, \sum_j w_{i,j}=1} \sum_{i=1}^{n} \|y_i - Wy_i\|_2^2 + \|W\|_1,$$

Bayes Dist. Clustering	Spectral Clustering	HDClassif
0.57 (0.54, 0.60)	0.50 (0.48, 0.52)	0.35 (0.31, 0.43)

Table 2: Accuracy of clustering MNIST hand-written digit data. Adjusted Rand index (ARI) is computed for the point estimates using variation of information. The average ARI and 95% confidence intervals are shown.

where the sparsity of $\hat{W}$ ensures only the data in the same linear subspace can have non-zero embedding coefficients. Afterward, we can define a subspace distance matrix as

$$d_{i,j} = 2 - \left(\frac{|\hat{w}_{i,j}|}{\max_{j'} |\hat{w}_{i,j'}|} + \frac{|\hat{w}_{j,i}|}{\max_{j} |\hat{w}_{j,j}|} \right),$$

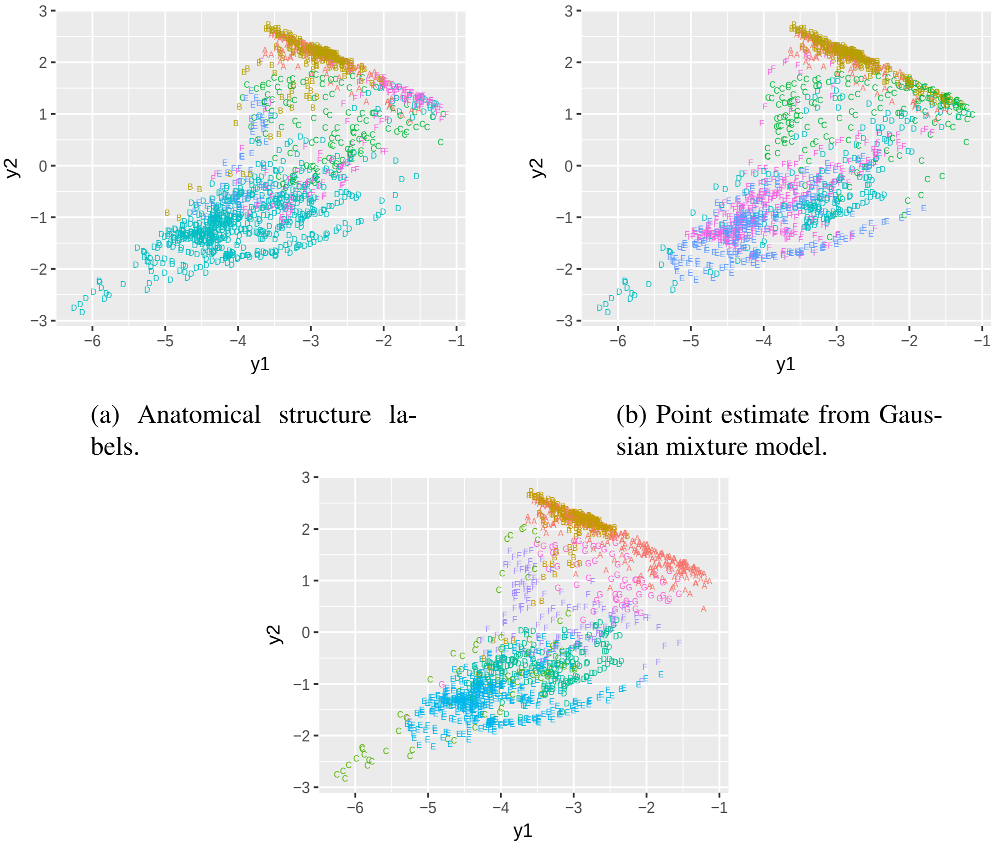
where we follow Vidal (2011) to normalize each row by its absolute maximum. We then use this distance matrix in our Bayesian distance clustering method.

To assess the performance, we use the MNIST data of hand-written digits of $0-9$, with each image having $p = 28 \times 28$ pixels. In each experiment, we take $n = 10,000$ random samples to fit the clustering models, among which each digit has approximately 1000 samples, and we repeat experiments 10 times. For comparison, we also run the near low-rank mixture model in *HDclassif* package (Bergé et al., 2012) and spectral clustering based on the p-dimensional vector norm. Our method using subspace distances shows clearly higher accuracy, as shown in Table 2.

7. Clustering brain regions

We carry out a data application to segment the mouse brain according to the gene expression obtained from the Allen Mouse Brain Atlas dataset (Lein et al., 2007). Specifically, the data are *in situ* hybridization gene expression, represented by expression volume over spatial voxels. Each voxel is a $(200\mu m)^3$ cube. We take the mid-coronal section of 41×58 voxels. Excluding the empty ones outside the brain, they have a sample size of $n = 1781$. For each voxel, there are records of expression volume over 3241 different genes. To avoid the curse of dimensionality for distances, we extract the first $p = 30$ principal components and use them as the source data.

Since gene expression is closely related to the functionality of the brain, we will use the clusters to represent the functional partitioning, and compare them in an unsupervised manner with known anatomical regions. The voxels belong to 12 macroscopic anatomical regions (Table 4).



(a) Anatomical structure labels.

(b) Point estimate from Gaussian mixture model.

(c) Point estimate from Bayesian Distance Clustering.

Figure 4: Clustering mouse brain using gene expression: visualizing the clustering result on the first two principal components.

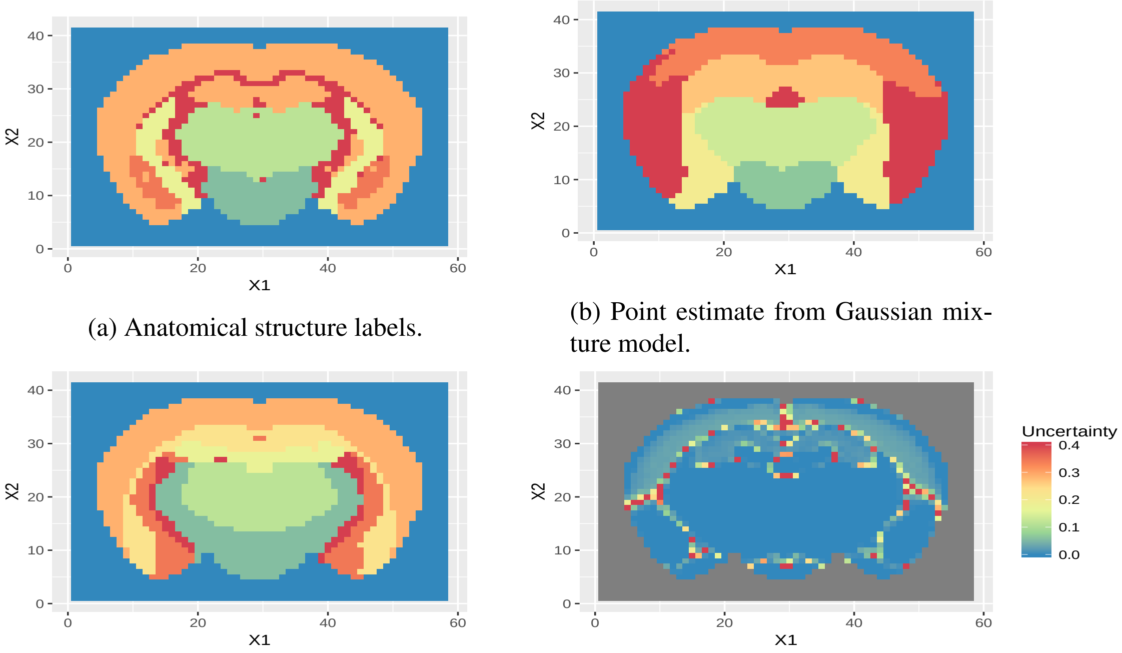
For clustering, we use an over-fitted mixture with $k = 20$ and small Dirichlet concentration parameter $\alpha = 1/20$. As shown by Rousseau and Mengersen (2011), asymptotically, small $\alpha < 1$ leads to the automatic emptying of small clusters; we observe such behavior here in this large sample. In the Markov chain, most iterations have 7 major clusters. Table 5 lists the voxel counts at $\hat{c}_{(n)}$.

Comparing the two tables, although we do not expect a perfect match between the structural and functional partitions, we do see a correlation in group sizes based on the top few groups. Indeed, visualized on the spatial grid (Figure 5), the point estimates from Bayesian distance clustering have very high resemblance to the anatomical structure. Comparatively, the clustering result from the Gaussian mixture model is completely different.

To benchmark against other distance clustering approaches, we compute various similarity scores and list the results in Table 3. Competing methods include spectral clustering (Ng et al., 2002), DBSCAN (Ester et al., 1996) and HDClassif (Bergé et al., 2012); the first two are applied on the same dimension-reduced data as used by Bayesian distance clustering, while the last one

	BDC	GMM	Spectral Clustering	DBSCAN	HDClassif
Adjusted Rand Index	0.49	0.31	0.45	0.43	0.43
Normalized Mutual Information	0.51	0.42	0.46	0.44	0.47
Adjusted Mutual Information	0.51	0.42	0.47	0.45	0.47

Table 3: Comparison of label point estimates using Bayesian distance clustering (BDC), Gaussian mixture model (GMM), spectral clustering, DBSCAN and HDClassif. The similarity measure is computed with respect to the anatomical structure labels.



(a) Anatomical structure labels.

(b) Point estimate from Gaussian mixture model.

(c) Point estimate from Bayesian Distance Clustering.

(d) Uncertainty based on Bayesian Distance Clustering: $\mathrm{pr}(c_i \neq \hat{c}_i)$

Figure 5: Clustering mouse brain using gene expression: visualizing the clustering result on the spatial grid of brain voxels. Comparing with the anatomical structure (panel a), Bayesian Distance Clustering (panel c) has a higher similarity than the Gaussian mixture model (panel b). Most of the uncertainty (panel d) resides in the inner layers of the cortical plate (upper parts of the brain).

is applied directly on the high dimensional data. Among all the methods, the point estimates of Bayesian Distance Clustering have the highest similarity to the anatomical structure.

Figure 5(d) shows the uncertainty about the point clustering estimates, in terms of the probability $\mathrm{pr}(c_i \neq \hat{c}_i)$. Besides the area connecting neighboring regions, most of the uncertainty resides in the inner layers of the cortical plate (upper parts of the brain). As a result, the inner cortical plate can be either clustered with the outer layer or with the inner striatum region. Therefore, from a practical perspective, when segmenting the brain according to the functionality, it is more appropriate to treat the inner layers as a separate region.

8. Discussion

The use of a distance likelihood reduces the sensitivity to the choice of a mixture kernel, giving the ability to exploit distances for characterizing complex and structured data. While we avoid specifying the kernel, one potential weakness is that there can be sensitivity to the choice of the distance metrics. For example, the Euclidean distance tends to produce a more spherical cluster, compared to the weighted Euclidean distance (see appendix). However, our results suggest that this sensitivity is often less than that of the assumed kernel. In many settings, there is a rich literature considering how to carefully choose the distance metric to reflect structure in the data (Pandit and Gupta, 2011). In such cases, the sensitivity of clustering results to the distance can be viewed as a positive. Clustering method necessarily relies on some notion of distances between data points.

Another issue is that we give up the ability to characterize the distribution of the original data. An interesting solution is to consider a modular modeling strategy that connects the distance clustering to a post-clustering inference model while restricting the propagation of cluster information in one direction only. Related modular approaches have been shown to be much more robust than a single overarching full model (Jacob et al., 2017).

Our concentration characterization of the within-cluster distance based on the vector norm holds for any arbitrary p. On the other hand, high-dimensional clustering is a subtle topic with challenging issues: (i) not all the coordinates in $\mathbb{R}^p$ contain discriminative information that is favorable for one particular partition; hence some alternative distances (Vidal, 2011), feature selection (Witten and Tibshirani, 2010), or multi-view clustering (Duan, 2020) may be necessary; (ii) the selection of number of clusters k becomes difficult, and it was recently discovered (Chandra et al., 2020) that the model-based framework could lead to nonsensical results of converging to either one cluster or n clusters even under a correctly specified model, as $p \to \infty$.

One interesting extension of this work is to combine with the nearest neighbor algorithm — we can choose to ignore the large distances and instead focus on modeling the K smallest ones for each data point — this could also significantly reduce the $\mathcal{O}(n^2)$ computing and storage cost, via the sparse matrix computation. One possible model is to replace our Gamma distance density with a two-component mixture: one Gamma component concentrating near zero for modeling small distances, and one Uniform$(0, \max_{i,j} d_{i,j})$ for handling large distances. Since the uniform density is a constant that does not depend on the specific value of the distance (as long it is in the support of the uniform), it effectively eliminates the influence of large distances in clustering.

Acknowledgement

This work was partially supported by grants R01-ES027498 and R01-MH118927 of the United States National Institutes of Health. The authors thank Amy Herring for helpful comments.

References

Afonso S Bandeira, Amit Singer, and Daniel A Spielman. A Cheeger Inequality for the Graph Connection Laplacian. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1611–1630, 2013.

Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. Clustering with Bregman Divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

Laurent Bergé, Charles Bouveyron, and Stéphane Girard. HDclassif: An R package for Model-based Clustering and Discriminant Analysis of High-dimensional Data. *Journal of Statistical Software*, 46(6):1–29, 2012.

Charles Bouveyron and Camille Brunet-Saumard. Model-based Clustering of High-dimensional Data: A Review. *Computational Statistics & Data Analysis*, 71:52–78, 2014.

Lev M Bregman. The Relaxation Method of Finding the Common Point of Convex Sets and Its Application To the Solution of Problems in Convex Programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

Noirrit Kiran Chandra, Antonio Canale, and David B Dunson. Bayesian Clustering of High-Dimensional Data. *arXiv preprint arXiv:2006.02700*, 2020.

Victor Chernozhukov and Han Hong. An MCMC Approach to Classical Estimation. *Journal of Econometrics*, 115(2):293–346, 2003.

Adam D Cobb and Brian Jalaian. Scaling Hamiltonian Monte Carlo Inference for Bayesian Neural Networks with Symmetric Splitting. *arXiv preprint arXiv:2010.06772*, 2020.

Pietro Coretto and Christian Hennig. Robust Improper Maximum Likelihood: Tuning, Computation, and a Comparison with Other Methods for Robust Gaussian Clustering. *Journal of the American Statistical Association*, 111(516):1648–1659, 2016.

David R Cox and Nancy Reid. A Note on Pseudolikelihood Constructed from Marginal Densities. *Biometrika*, 91(3):729–737, 2004.

Leo L Duan. Latent Simplex Position Model. *Journal of Machine Learning Research*, 21, 2020.

David B Dunson and Jack A Taylor. Approximate Bayesian Inference for Quantiles. *Journal of Nonparametric Statistics*, 17(3):385–400, 2005.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, volume 96, pages 226–231, 1996.

Chris Fraley and Adrian E Raftery. Model-based Clustering, Discriminant Analysis, and Density Estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

Michael PB Gallaugher and Paul D McNicholas. Finite Mixtures of Skewed Matrix Variate Distributions. *Pattern Recognition*, 80:83–93, 2018.

Christian Hennig, Marina Meila, Fionn Murtagh, and Roberto Rocci. *Handbook of Cluster Analysis*. CRC Press, 2015.

Peter D Hoff. Extending the Rank Likelihood for Semiparametric Copula Estimation. *The Annals of Applied Statistics*, 1(1):265–283, 2007.

Matthew D Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

Hesam Izakian, Witold Pedrycz, and Iqbal Jamal. Fuzzy Clustering of Time Series Data Using Dynamic Time Warping Distance. *Engineering Applications of Artificial Intelligence*, 39:235–244, 2015.

Pierre E Jacob, Lawrence M Murray, Chris C Holmes, and Christian P Robert. Better Together? Statistical Learning in Models Made of Modules. *arXiv preprint arXiv:1708.08719*, 2017.

Anil K Jain. Data Clustering: 50 Years Beyond K-means. *Pattern Recognition Letters*, 31(8): 651–666, 2010.

Harold Jeffreys. *The Theory of Probability*. OUP Oxford, 1961.

Valen E Johnson. Bayes Factors Based on Test Statistics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(5):689–701, 2005.

Miguel A Juárez and Mark FJ Steel. Model-based Clustering of Non-gaussian Panel Data Based on Skew-t Distributions. *Journal of Business and Economic Statistics*, 28(1):52–66, 2010.

Dimitris Karlis and Anais Santourian. Model-based Clustering with Non-elliptically Contoured Distributions. *Statistics and Computing*, 19(1):73–83, 2009.

Da Kuang, Chris Ding, and Haesun Park. Symmetric Nonnegative Matrix Factorization for Graph Clustering. In *Proceedings of the 2012 SIAM international conference on data mining*, pages 106–117. SIAM, 2012.

Da Kuang, Sangwoon Yun, and Haesun Park. SymNMF: Nonnegative Low-Rank Approximation of a Similarity Matrix for Graph Clustering. *Journal of Global Optimization*, 62(3):545–574, 2015.

Ed S Lein, Michael J Hawrylycz, Nancy Ao, Mikael Ayres, Amy Bensinger, Amy Bernard, Andrew F Boe, Mark S Boguski, Kevin S Brockway, and Emi J Byrnes. Genome-wide Atlas of Gene Expression in the Adult Mouse Brain. *Nature*, 445(7124):168, 2007.

Jia Li, Surajit Ray, and Bruce G Lindsay. A Nonparametric Statistical Approach to Clustering via Mode Identification. *Journal of Machine Learning Research*, 8:1687–1723, 2007.

Bruce G Lindsay. Composite Likelihood Methods. *Contemporary Mathematics*, 80(1):221–239, 1988.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. *International Conference on Learning Representations*, 2017.

Gertraud Malsiner-Walli, Sylvia Frühwirth-Schnatter, and Bettina Grün. Identifying Mixtures of Mixtures Using Bayesian Estimation. *Journal of Computational and Graphical Statistics*, 26(2): 285–295, 2017.

Paul D McNicholas. Model-based Clustering. *Journal of Classification*, 33(3):331–373, 2016.

Jeffrey W Miller and David B Dunson. Robust Bayesian Inference via Coarsening. *Journal of the American Statistical Association*, pages 1–13, 2018.

Radford M Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2:113–162, 2011.

Andrew Y Ng, Michael I Jordan, and Yair Weiss. On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, 2002.

Adrian O'Hagan, Thomas Brendan Murphy, Isobel Claire Gormley, Paul D McNicholas, and Dimitris Karlis. Clustering With the Multivariate Normal Inverse Gaussian Distribution. *Computational Statistics & Data Analysis*, 93:18–30, 2016.

Shraddha Pandit and Suchita Gupta. A Comparative Study on Distance Measuring Approaches for Clustering. *International Journal of Research in Computer Science*, 2(1):29–31, 2011.

William M Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

Carlos E Rodríguez and Stephen G Walker. Univariate Bayesian Nonparametric Mixture Modeling with Unimodal Kernels. *Statistics and Computing*, 24(1):35–49, 2014.

Judith Rousseau and Kerrie Mengersen. Asymptotic Behaviour of the Posterior Distribution in Overfitted Mixture Models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(5):689–710, 2011.

Peng Sun and Robert M Freund. Computation of Minimum-volume Covering Ellipsoids. *Operations Research*, 52(5):690–706, 2004.

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

Sara Wade and Zoubin Ghahramani. Bayesian Cluster Analysis: Point Estimation and Credible Balls. *Bayesian Analysis*, 13(2):559–626, 2018.

Martin J Wainwright. *High-dimensional Statistics: A Non-asymptotic Viewpoint*, volume 48. Cambridge University Press, 2019.

Daniela M Witten and Robert Tibshirani. A Framework for Feature Selection in Clustering. *Journal of the American Statistical Association*, 105(490):713–726, 2010.

Dongkuan Xu and Yingjie Tian. A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, 2(2):165–193, 2015.

Han Zhao, Pascal Poupart, Yongfeng Zhang, and Martin Lysy. Sof: Soft-Cluster Matrix Factorization for Probabilistic Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Appendix

Proof of Lemma 2

Proof We first focus on $x \sim \text{Gamma}(\alpha, 1)$, by the Markov's inequality

$$\text{pr}(x \geq t) \leq \frac{\mathbb{E}\exp(sX)}{\exp(st)} = (1-s)^{-\alpha} e^{-ts},$$

where $s < 1$. Minimizing the right hand side over s yields $s^* = 1 - \alpha/t$, and

$$\text{pr}(x \geq t) \leq (\frac{t}{\alpha})^{\alpha} e^{-t+\alpha} = \alpha^{-\alpha} e^{\alpha} t^{\alpha} e^{-t}.$$

Scaling x by σ_h and adjusting the constant yield the results.

■

Proof of Lemma 4

Proof Equivalently, the sub-exponential tail can be characterized by the bound on its moment generating function

$$\mathbb{E}\exp\{t(y^{[h]}_{i,k} - \mu^{[h]}_{k})\} \leq \exp(\nu_h^2 t^2/2) \quad \forall |t| \leq 1/b_h,$$

for $k = 1, \ldots, p$. It immediately follows that the pairwise difference $\tilde{d}^{[h]}_{i,j} = y^{[h]}_i - y^{[h]}_j$ between two iid random variables must be sub-exponential as well, with

$$\mathbb{E}\exp(t\tilde{d}^{[h]}_{i,j,k}) \leq \exp(\nu_h^2 t^2) \quad \forall |t| \leq 1/b_h.$$

That is, sub-exponential–$(\sqrt{2}\nu_h, b_h)$. Then the vector norm

$$\begin{aligned}
\text{pr}(d^{[h]}_{i,j} > p^{\eta} t) &= \text{pr}(\sum_{k=1}^{p} |\tilde{d}^{[h]}_{i,j,k}|^q > p^{\eta q} t^q) \\
&\leq p\ \text{pr}(|\tilde{d}^{[h]}_{i,j,k}|^q > p^{\eta q - 1} t^q) \\
&= p\ \text{pr}(|\tilde{d}^{[h]}_{i,j,k}| > p^{\eta - 1/q} t) \\
&\leq 2p \exp\{-t p^{(\eta - 1/q)}/(2b_h)\} \quad \text{ for } t > p^{1/q - \eta} 2\nu_h^2/b_h.
\end{aligned}$$

where the first inequality is due to $\text{pr}(\sum_{i=1}^{p} a_i > b) \leq \text{pr}(\text{There is at least one } i\text{: } a_i > b/p) \leq \sum_{i=1}^{p} \text{pr}(a_i > b/p)$ and second inequality uses $\mathbb{E}d^{[h]}_{i,j,k} = 0$ and the property of sub-exponential tail (Wainwright, 2019). ■

Proof of Lemma 7

Proof For a clear exposition, we omit the sub/super-script h for now and use $x_i = T(y_i)$

$$
\begin{aligned}
\mathbb{E}_{y_i}\mathbb{E}_{y_j}\sum_{i=1}^{n}\sum_{j=1}^{n} B_\phi(x_i,x_j) =& \mathbb{E}_{y_i}\mathbb{E}_{y_j}\sum_{i=1}^{n}\sum_{j=1}^{n}\{\phi(x_i)-\phi(x_j)-\langle\, x_i-x_j,\nabla\phi(x_j)\rangle\}\\
=&\mathbb{E}_{y_j}\sum_{j=1}^{n}\sum_{i=1}^{n}\{\mathbb{E}_{y_i}\phi(x_i)-\phi(\mu)-\langle\,\mathbb{E}_{y_i}x_i-\mu,\nabla\phi(\mu)\rangle\\
&+\phi(\mu)-\phi(x_j)-\langle\,\mathbb{E}_{y_i}x_i-x_j,\nabla\phi(x_j)\rangle\\
=&n\sum_{i=1}^{n}\mathbb{E}_{y_i}\{\phi(x_i)-\phi(\mu)-\langle\, x_i-\mu,\nabla\phi(\mu)\rangle\}\\
&+n\sum_{j=1}^{n}\mathbb{E}_{y_j}\{\phi(\mu)-\phi(x_j)-\langle\,\mu-x_j,\nabla\phi(x_j)\rangle\}\\
=&n\sum_{i=1}^{n}\mathbb{E}_{y}\{B_\phi(x_i,\mu)+B_\phi(\mu,x_i)\},
\end{aligned}
$$

where $\langle .,.\rangle$ denotes dot product, the second equality is due to Fubini theorem and $\mathbb{E}_{y_i}x_i-\mu=0$. ■

Proof of Lemma 9

Proof Using $\mathbf{1}_{n,m}$ $n\times m$ matrix with all elements equal 1. Since $C^{\mathrm{T}}C=\text{diag}(n_1,\ldots,n_k)$, the 2 times of normalized graph cut loss can be written as

$$
\begin{aligned}
&\text{tr}\big[A(\mathbf{1}_{n,k}-C)\big(C^{\mathrm{T}}C\big)^{-1}C^{\mathrm{T}}\big]\\
&=-\text{tr}\big\{AC\big(C^{\mathrm{T}}C\big)^{-1}C^{\mathrm{T}}\big\}+\text{tr}\big[A\mathbf{1}_{n,k}\big(C^{\mathrm{T}}C\big)^{-1}C^{\mathrm{T}}\big].
\end{aligned}
$$

For the second term

$$
\begin{aligned}
&\text{tr}\big[A\mathbf{1}_{n,k}\big(C^{\mathrm{T}}C\big)^{-1}C^{\mathrm{T}}\big]\\
=&\text{tr}\big[C^{\mathrm{T}}A\mathbf{1}_{n,k}\big(C^{\mathrm{T}}C\big)^{-1}\big]\\
=&\sum_{i=1}^{n}\frac{\sum_{j=1}^{n}A_{i,j}}{n_{c_i}}.
\end{aligned}
$$

■

Anatomical Structure Name	Voxel Count
Cortical plate	718
Striatum	332
Thalamus	295
Midbrain	229
Basic cell groups and regions	96
Pons	56
Vermal regions	22
Pallidum	14
Cortical subplate	6
Hemispheric regions	6
Cerebellum	5
Cerebral cortex	2

Table 4: Names and voxel counts in 12 macroscopic anatomical structures in the coronal section of the mouse brain. They represent the *structural* partitioning of the brain.

Index	Voxel Count
1	626
2	373
3	176
4	113
5	79
6	39
7	12

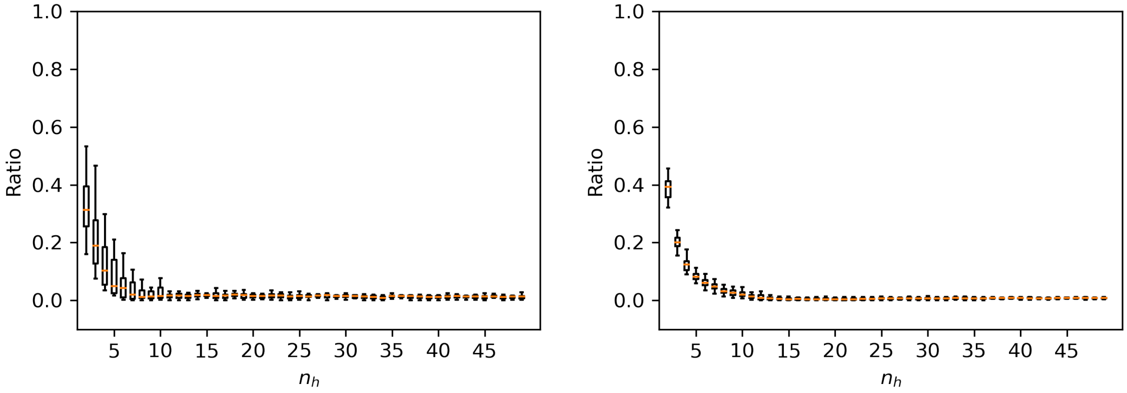
Table 5: Group indices and voxel counts in 7 clusters found by Bayesian Distance Clustering, using the gene expression volume over the coronal section of the mouse brain. They represent the *functional* partitioning of the brain.

Numerical experiments showing the effect of discarding the seed

We show numerically that as n_h increases, the seed conditional density $\mathcal{K}_h\big(y_1^{[h]} \mid \tilde{d}_{2,1}^{[h]}, \ldots, \tilde{d}_{n_h,1}^{[h]}\big)$ becomes very small in magnitude compared to the distance likelihood $G_h\big(\tilde{d}_{2,1}^{[h]}, \ldots, \tilde{d}_{n_h,1}^{[h]}\big)$. To compute those two terms, we use the Gaussian example presented in Section 2.1, simulated with $\sigma_h^2 = 1$, $\mu_h = 0$. Since the values of those densities are very close to zero, we use the log-scale and compute the ratio,

$$\left| \frac{\log \mathcal{K}_h\big(y_1^{[h]} \mid \tilde{d}_{2,1}^{[h]}, \ldots, \tilde{d}_{n_h,1}^{[h]}\big)}{\log \mathcal{K}_h\big(y_1^{[h]} \mid \tilde{d}_{2,1}^{[h]}, \ldots, \tilde{d}_{n_h,1}^{[h]}\big) G_h\big(\tilde{d}_{2,1}^{[h]}, \ldots, \tilde{d}_{n_h,1}^{[h]}\big)} \right|$$

as a measure of how small $\mathcal{K}_h(y_1^{[h]} \mid .)$ compared to G_h. We repeat each experiment for 30 times and create the box plots. As can be seen from Figure 6 (left panel), the ratio quickly drops to near zero after $n_h \geq 10$. In addition, we repeat the same experiment except using a multivariate Gaussian $\mathrm{N}(0, I_p)$; and the findings are very similar (right panel).

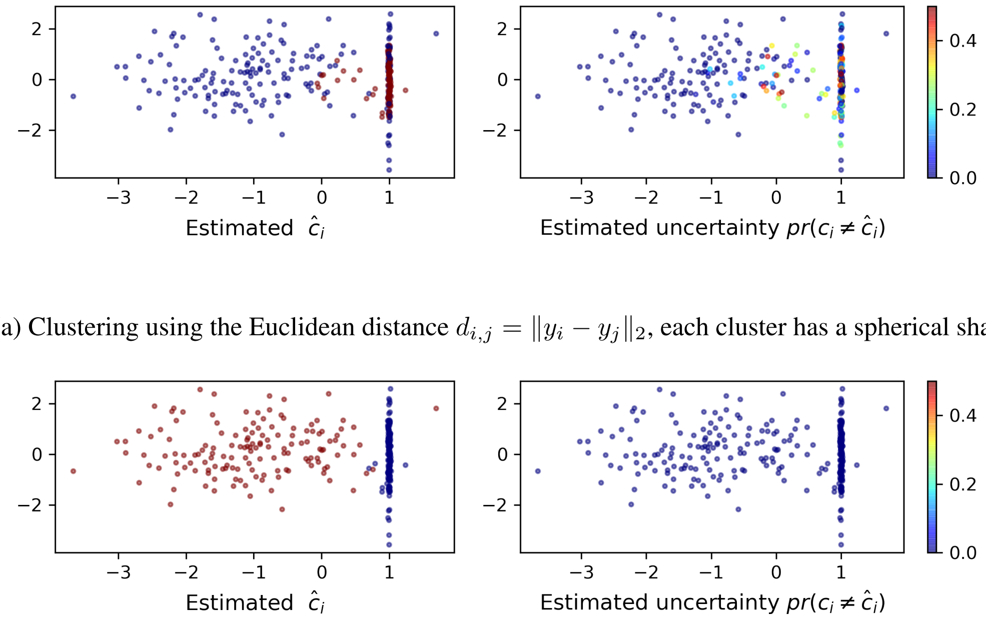


(a) $p = 1$.

(b) $p = 30$.

Figure 6: Numerical experiments show that the seed conditional density $\mathcal{K}_h\big(y_1^{[h]} \mid \tilde{d}_{2,1}^{[h]}, \ldots, \tilde{d}_{n_h,1}^{[h]}\big)$ becomes much smaller than the distance likelihood as n_h increases.

The effect of distances on the shapes of clusters



(a) Clustering using the Euclidean distance $d_{i,j} = \|y_i - y_j\|_2$, each cluster has a spherical shape.

(b) Clustering using the weighted Euclidean distance $d_{i,j} = \sqrt{(y_i - y_j)^{\mathrm{T}} S (y_i - y_j)}$, with $S = \mathrm{diag}(s_1, s_2)$, each induced cluster has an elliptical shape.

Figure 7: Experiments show that the choice of distances may impact the shapes of the clusters and the uncertainty. Although, if one wants to separate clusters based on different 'shapes', a model on the cluster-specific covariances could be more useful than one on the pairwise distances.

Connection with the nonnegative matrix factorization methods

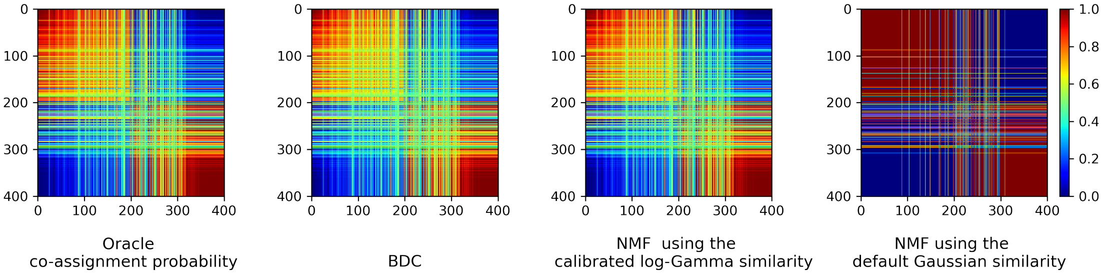


Figure 8: Comparing the performance of the Bayesian distance clustering (BDC) and the nonnegative matrix factorization (NMF) using the symmetric simplex matrix factorization (Zhao et al., 2015; Duan, 2020). We apply the algorithms on the two clusters of skew Gaussian data as generated in Section 2.1, and plot the estimated co-assignment probability matrix for each method. As can be seen from Panels 2 and 3, when NMF is well calibrated in its similarity function (as we defined in (13), with parameters set to the posterior mean estimate from BDC), it produces a result almost indistinguishable from BDC — both are very close to the oracle co-assignment probability (Panel 1). On the other hand, NMF using an uncalibrated similarity $\exp(-d_{i,j}^2)$ produces a result (Panel 4) very different from the oracle.