

Bayesian Clustering of Shapes of Curves

Zhengwu Zhang^{a,*}, Debdeep Pati^a, Anuj Srivastava^a

^a*Department of Statistics, Florida State University, FL, 32306*

Abstract

Unsupervised clustering of curves according to their shapes is an important problem with broad scientific applications. The existing model-based clustering techniques either rely on simple probability models (e.g., Gaussian) that are not generally valid for shape analysis or assume the number of clusters. We develop an efficient Bayesian method to cluster curve data using an elastic shape metric that is based on joint registration and comparison of shapes of curves. The elastic-inner product matrix obtained from the data is modeled using a Wishart with parameters that follow certain prior models. This allows for automatic inference on the number of clusters. Posterior is sampled through an efficient Markov chain Monte Carlo procedure based on the Chinese restaurant process to infer (1) the posterior distribution on the number of clusters, and (2) clustering configuration of shapes. This method is demonstrated on a variety of synthetic data and real data examples on protein structure analysis, cell shape analysis in microscopy images, and clustering of shaped from MPEG7 database.

Keywords: clustering; shapes of curves; Chinese restaurant process; Wishart distribution.

1. Introduction

The automated clustering of objects is an important area of research in unsupervised classification of large object databases. The general goal here is to choose groups (clusters) of objects so as to maximize homogeneity within clusters and minimize homogeneity across clusters. The clustering problem has been addressed by researchers in many disciplines. A few well-known methods are metric based e.g. K-means (MacQueen et al., 1967), hierarchical clustering (Ward, 1963), clustering based on principal components, spectral clustering (Ng et al., 2002) and so on (Jain and Dubes, 1988; Ozawa, 1985). Traditional clustering methods are complemented by methods based on a probability model where one assumes a data generating distribution (e.g., Gaussian) and infers clustering configurations that maximize certain objective function (Banfield and Raftery, 1993; Fraley and Raftery, 1998, 2002, 2006; MacCullagh and Yang, 2008). A model-based clustering can be useful in addressing challenges posed by traditional clustering methods. This is because a probability model allows the number of clusters to be treated as a parameter in the model, and can be embedded in a Bayesian framework providing quantification of uncertainty in the number of clusters and clustering configurations.

A popular probability model is obtained by considering that the population of interest consists of K different sub-populations and the density of the observation y from the k^{th} sub-population is f_k . Given observations y_1, \dots, y_n , we introduce indicator random variables (c_1, \dots, c_n) such that $c_i = k$ if y_i comes from

*Corresponding author. Tel. : +1 850-644-7412

Email addresses: zhengwu@stat.fsu.edu (Zhengwu Zhang), debdeep@stat.fsu.edu (Debdeep Pati), anuj@stat.fsu.edu (Anuj Srivastava)

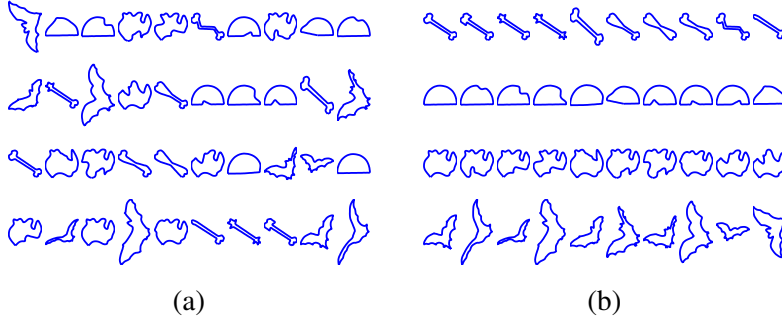


Figure 1: Clustering shapes of curves. (a) 40 shapes in arbitrary order. (b) Clustering result of the 40 shapes. Each row is a cluster.

the k th sub-population. The maximum likelihood inference is based on finding the value of (c, f_1, \dots, f_k) that maximizes the likelihood $\prod_{i=1}^n f_{c_i}(y_i)$. Typically K is assumed to be known or a suitable upper bound is assumed for convenience. When $y_i \in \mathbb{R}^p$, f_k is commonly parametrized by a multivariate Gaussian density with mean vector μ_k and covariance matrix Σ_k . An alternative is to use a nonparametric Bayesian approach which has an appealing advantage of allowing K to be unknown and inferring it from the data. A further bonus of a Bayesian approach is the easy availability of an entire distribution of the inferred number of clusters.

The vast majority of the literature on model-based clustering is almost exclusively focused on Euclidean data. This is primarily due to the easy availability of parametric distributions on the Euclidean space as well as computational tractability of estimating the cluster centers. For clustering functional data, e.g. shapes of curves, one encounters several challenges. Unlike Euclidean data, where the notions of cluster centers and cluster variance are standard, these quantities and the resulting quantification of homogeneity within clusters is not obvious for shape spaces. Moreover, it is important to use representations and metrics for clustering objects that are invariant to shape-preserving transformations (rigid motions, scaling, and re-parametrization). For example, Kurtek et al. (2012) takes a model-based approach for clustering of curves using an elastic metric that has proper invariances. However, under the chosen representations and metrics, even simple summary statistics of the observed data are difficult to compute. Other existing shapes clustering methods (Belongie et al., 2002; Liu et al., 2012) either extract finite-dimensional features to represent the shapes or project the high-dimensional shape space to a low-dimensional space (Yankov and Keogh, 2006; Auder and Fischer, 2012), and then apply clustering methods for Euclidean data; these are not generally valid in all applications. Also, several methods (Srivastava et al., 2005; Gaffney and Smyth, 2005) have been proposed to cluster non-Euclidean data based on a distance-based notion of dispersion, thus, avoiding the computation of shape means (e.g. Karcher means), but they all assume a given number of clusters.

In this paper we develop a model-based clustering method for curve data that does not require the knowledge of cluster number K apriori. This approach is based on modeling a summary statistic that encodes the clustering information, namely the pairwise inner-product matrix. The salient points of this approach are: (1) The comparison of curves is based on pairwise inner-product matrix under elastic shape analysis, so that the analysis is invariant to all desired shape-preserving transformations. (2) The inner-product matrix is modeled using a mixture of Wishart distributions induced by the Chinese restaurant process (Vogt et al., 2010). A model directly on the inner-product matrix has an appealing advantage of reducing computational cost substantially by avoiding computation of the Karcher means. (3) We formulate and sample from a pos-

terior on the number of clusters, and use the mode of this distribution for final clustering. We illustrate our ideas through several synthetic and real data examples. The results show that our model on the inner-product matrix leads to a more accurate estimate of the number of clusters as well as the clustering configurations compared to a Bayesian nonparametric model directly on the data, even in the Euclidean case.

This paper is organized as follows. We start by introducing two case studies in Section 2. The mathematical details of the metric used for computing the inner-product and the model specifications are presented in Section 3. In Section 4, we illustrate our methodology on several synthetic data examples and the case studies on clustering cell shapes and protein structures. Section 5 closes the paper with some conclusions.

2. Case studies

We propose to undertake two specific case studies involving clustering of curve data.

2.1. Clustering of protein sequences

Protein structure analysis is an outstanding scientific problem in structural biology. A large number of new proteins are regularly discovered and scientists are interested in learning about their functions in larger biological systems. Since protein functions are closely related to their folding patterns and structures in native states, the task of structural analysis of proteins becomes important. In terms of evolutionary origins, proteins with similar structures are considered to have common evolutionary origin. The Structural Classification of Proteins (SCOP) database (Murzin et al., 1995) provides a manual classification of protein structural domains based on similarities of their structures and amino acid sequences. Refer to Fig. 2 for a snapshot of the proteins in \mathbb{R}^3 and the 3-coordinates of the protein sequences. Clustering protein sequences is extremely important to trace the evolutionary relationship between proteins and for detecting conserved structural motifs. In this article, we focus on an automated clustering of protein sequences based on their global structures.

2.2. Clustering of cell shapes

The problem of studying shapes of cellular structures using microscopic image data is very important medical diagnosis (Rohde et al., 2008) and genetic engineering (Thomas et al., 2002). This research involves extracting cell contours from images using segmentation techniques (Hagwood et al., 2012) and then studying shapes of these extracted contours for medical diagnosis. We will focus on the problem of clustering of cells according to their shapes; these clusters can be further used for statistical modeling and hypothesis testing although these steps are not pursued in the current paper. The specific database used here was obtained by segmenting the 2D microscopy images, as described in Hagwood et al. (2012, 2013). Fig. 3 (b) shows some examples of the cell contours used in this paper. In this article, we consider two types of cell shapes: DLEX-p46 cell shapes and NIH-3T3 cell shapes. Visually, DLEX-P46 cells are round, denoting normal cell shapes whereas, NIH-3T3 cells have an elongated, spindly appearance, denoting progression of some pathological conditions.

3. Methodology

Previous methods of clustering non-Euclidean objects can be broadly categorized into two parts: (1) clustering based on representation of the data in an infinite-dimensional quotient space under a chosen Riemannian metric and (2) clustering based on suitable summary statistic of the data e.g. distance matrices. Any representation in the infinite-dimensional space involves the calculation of the mean and the covariance

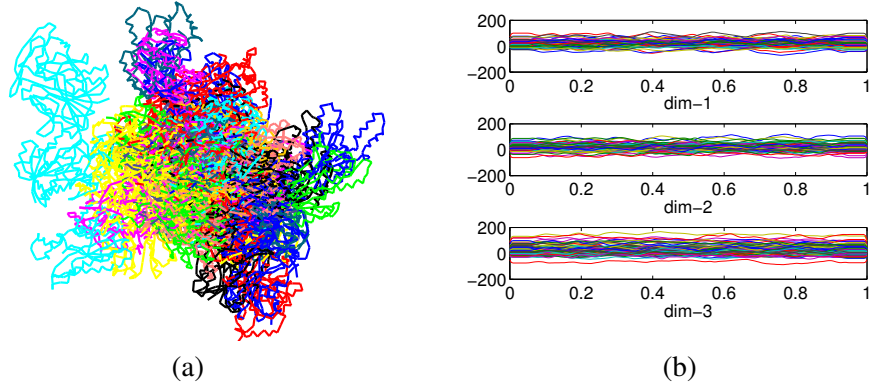


Figure 2: Protein sequences. (a) raw protein structure data in \mathbb{R}^3 . (b) 3-dimensional components of the protein sequences, where the x -axis indicates the length of each sequence.

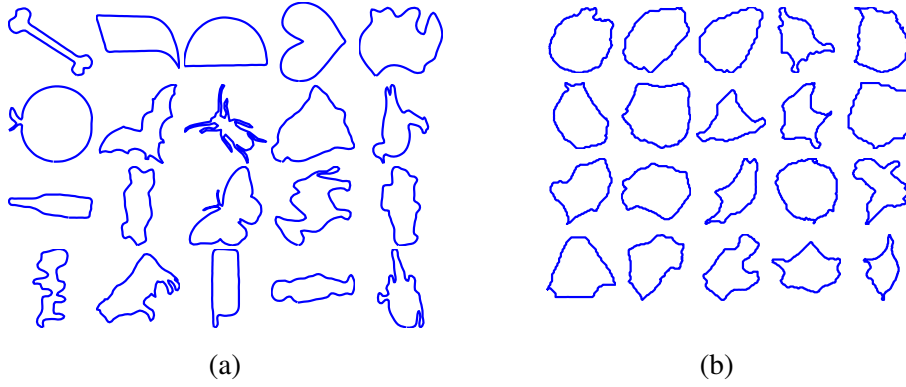


Figure 3: Example shapes. (a) example shapes in the MPEG-7 dataset; (b) example cell shapes.

matrix (Kurtek et al., 2012; Tucker et al., 2013) which is computationally expensive. To avoid calculating the mean and the covariance matrix, Srivastava et al. (2005) developed a method for clustering functional data based on pairwise distance matrix and resorted to stochastic simulated annealing for fast implementation. Although the method is quite efficient, one requires the knowledge of the number of clusters apriori.

In this paper, we develop a mixture of Wisharts model for the pairwise inner-product matrices to cluster shapes of curves. Since we model a summary statistic of the data as in (Adametz and Roth, 2011; Vogt et al., 2010) instead of the infinite dimensional data points, our method is computationally efficient. However, unlike (Adametz and Roth, 2011; Vogt et al., 2010) which consider a standard \mathbb{L}^2 metric to calculate the distance matrices, the inner product matrix is calculated using a specific representation of curves called *square-root velocity function* (SRVF) (Srivastava et al., 2011). This along with some registration techniques make the inner-product invariant to the shape preserving transformations, thus eliminating the drawback of (Adametz and Roth, 2011; Vogt et al., 2010). Moreover, a Bayesian nonparametric approach allows us to do automatic inference on the number of clusters.

Below, we describe the mathematical framework for computing the inner-product matrix.

3.1. Inner product matrix using elastic shape analysis

We adapt the elastic shape analysis introduced in Srivastava et al. (2011) to calculate the inner-product matrix in the *square-root velocity function* (SRVF) space for the non-Euclidean functional data. Let $\beta : D \rightarrow \mathbb{R}^p$ be a parameterized curve in \mathbb{R}^p with domain D . We restrict our attention to those β which are absolutely continuous on D . Usually $D = [0, 1]$ for open curves and $D = \mathbb{S}^1$ for closed curves. Define $\mathcal{F} = \{\beta : D \rightarrow \mathbb{R}^p : \beta \text{ is absolutely continuous on } D\}$ and a continuous mapping: $Q : \mathbb{R}^p \rightarrow \mathbb{R}^p$ as

$$Q(x) \equiv \begin{cases} x/\sqrt{|x|} & \text{if } |x| \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Here, $|\cdot|$ is the Euclidean 2-norm in \mathbb{R}^p . For the purpose of studying the shape of a curve β , we will represent it as: $q : D \rightarrow \mathbb{R}^p$, where $q(t) \equiv Q(\dot{\beta}(t))$. The function $q : D \rightarrow \mathbb{R}^p$ is called *square-root velocity function* (SRVF). It can be shown that for any $\beta \in \mathcal{F}$, the resulting SRVF is square integrable. Hence, we will define $\mathbb{L}^2(D, \mathbb{R}^p)$ to be the set of all SRVFs. For every $q \in \mathbb{L}^2(D, \mathbb{R}^p)$ there exists a curve β (unique up to a constant, or a translation) such that the given q is the SRVF of that β .

There are several motivations for using SRVF for functional data analysis. First, an elastic metric becomes the standard \mathbb{L}^2 metric under the SRVF representation (Srivastava et al., 2011). This elastic metric is invariant to the re-parameterization of curves and provides nice physical interpretations. Although the original elastic metric has a complicated expression, the SRVF transforms it into the \mathbb{L}^2 metric, thus providing a substantial simplification in terms of computing the metric.

By representing a parameterized curve β by its SRVF q , we have taken care of the translation variability, but the scaling, rotation and the re-parameterization variabilities still remain. In some applications like clustering of protein sequences, it is not advisable to remove the scaling variabilities as the length can be a predictor of its biological functions. On the contrary, in applications like clustering images with the camera placed at variable distances, it is necessary to remove the scales by rescaling all curves to be of unit length, i.e., $\int_D |\dot{\beta}(t)| dt = \int_D |q(t)|^2 dt = 1$. The set of all SRVFs representing unit-length curves is a unit hypersphere in the Hilbert manifold $\mathbb{L}^2(D, \mathbb{R}^p)$. We will use \mathcal{C}^o to denote this hypersphere, i.e., $\mathcal{C}^o = \{q \in \mathbb{L}^2(D, \mathbb{R}^p) | \int_D |q(t)|^2 dt = 1\}$. A rigid rotation in \mathbb{R}^p is represented as an element of $SO(p)$, the special orthogonal group of $p \times p$ matrices. The rotation action is defined to be $SO(p) \times \mathcal{C}^o \rightarrow \mathcal{C}^o$ as follows. If a curve is rotated by a rotation matrix $O \in SO(p)$, then its SRVF is also rotated by the same matrix, i.e. the SRVF of $O\beta(t)$ is $Oq(t)$, where q is the SRVF of β . A re-parameterization function is an element of

Γ , the set of all orientation-preserving diffeomorphisms of D . For any $\beta \in \mathcal{F}$ and $\gamma \in \Gamma$, the composition $\beta \circ \gamma$ denotes the re-parameterization of β by γ . The SRVF of $\beta \circ \gamma$ is given by: $\tilde{q}(t) = q(\gamma(t))\sqrt{\dot{\gamma}(t)}$. We will use (q, γ) to denote $q(\gamma(t))\sqrt{\dot{\gamma}(t)}$ in the following.

It is easy to show that the actions of $SO(p)$ and Γ on \mathcal{C}^o commute each other, thus we can form a joint action of the product group $SO(p) \times \Gamma$ on \mathcal{C}^o according to $((O, \gamma), q) = O(q \circ \gamma)\sqrt{\dot{\gamma}}$. The action of the product group $\Gamma \times SO(p)$ is by isometries under the chosen Riemannian metric. The orbit of an SRVF $q \in \mathcal{C}^o$ is the set of SRVFs associated with all the reparameterizations and rotations of a given curve and is given by: $[q] = \text{closure}\{(q, (O, \gamma)) | (O, \gamma) \in SO(p) \times \Gamma\}$. The specification of orbits is important because each orbit uniquely represents a shape and, therefore, analyzing the shapes is equivalent to the analysis of orbits. The set of all such orbits is denoted by \mathcal{S} and termed the *shape space*. \mathcal{S} is actually a quotient space given by $\mathcal{S} = \mathcal{C}^o / (SO(p) \times \Gamma)$. Now we can define an inner-product on the space \mathcal{S} which is invariant to translation, scaling, rotation and reparameterization of curves.

Definition 1. (*Inner-product on shape space of curves*). For given curves $\beta_1, \beta_2 \in \mathcal{F}$ and the corresponding SRVFs, q_1, q_2 , we define the inner-product, s_{β_1, β_2} or $\langle [q_1], [q_2] \rangle_{\mathbb{H}}$, to be:

$$s_{\beta_1, \beta_2} = \sup_{\gamma \in \Gamma, O \in SO(d)} \langle q_1, (q_2, (O, \gamma)) \rangle.$$

Note that this inner product is well-defined because the action on $SO(p) \times \Gamma$ is by isometries.

Optimization over $SO(p)$ and Γ : The maximization over $SO(p)$ and Γ can be performed iteratively as in (Srivastava et al., 2011). In our case, we use Dynamic Programming algorithm to solve for an optimal γ first. Then we fix γ , and search for the optimal rotation O in $SO(p)$ using a rotational Procrustes algorithm (Kurtsek et al., 2012).

3.2. Model for the inner product matrix

Let $S_{+n}(\mathbb{R})$ denote the set of all $n \times n$ symmetric non-negative definite matrices over \mathbb{R} . Depending on whether we rescale the curves to have unit length or not, we define two classes of inner product matrices: i) $U_{+n}(\mathbb{R}) = \{A \in S_{+n}(\mathbb{R}) : a_{ii} = 1, |a_{ij}| \leq 1, 1 \leq i \neq j \leq n\}$ and ii) $S_{+n}(\mathbb{R})$. In this article, we do not make a distinction between these two cases and specify our model for the larger subspace $S_{+n}(\mathbb{R})$ irrespective of whether we rescale the curves or not. As illustrated using the experimental results in Section 4, having a probability model on a slightly larger space does not pose any practical issues when we actually rescale the curves as long as the probability model puts sufficient mass on the set $U_{+n}(\mathbb{R})$. This is achieved by enforcing a special structure on our model (see (3)) so that the expectation of the probability density can be made to lie inside the space $U_{+n}(\mathbb{R})$ for appropriate choices of the hyperparameters. Hence standard concentration results imply that the model gives sufficient mass around elements of $U_{+n}(\mathbb{R})$.

For a scaled inner product matrix $S \in S_{+n}(\mathbb{R})$, let $S \sim W_n(\Sigma, d)$, the Wishart distribution with degrees of freedom d and parameter $\Sigma = \mathbb{E}(S)$ of rank n ($d > n$). To allow rank-deficient S and Σ , a *generalized Wishart distribution* with degrees of freedom d ($d < n$) can be defined as

$$p(S | \Sigma, d) \propto |S|^{(d-n-1)/2} |\Sigma^{-1}|^{d/2} \exp \left\{ -\frac{d}{2} \text{tr}(\Sigma^{-1}S) \right\}, \quad (1)$$

where $|\cdot|$ implies the product of non-zero eigenvalues and $\text{tr}(\cdot)$ is the sum of the diagonal elements. For an observation of S , the log-likelihood function is

$$l(\Sigma; S, d) \propto -\frac{d}{2} \log(|\Sigma|) - \frac{d}{2} \text{tr}(\Sigma^{-1}S) \quad (2)$$

for $\Sigma \in S_{+n}(\mathbb{R})$. One can easily identify this as an exponential family distribution with canonical parameter $W = \Sigma^{-1}$, and the deviance is minimized at $\Sigma = S$ (McCullagh, 2009). Therefore, Σ encodes the similarity between the observed shapes measured by the inner product S . For instance, Σ_{jk} encodes the similarity between y_i and y_j as measured by the inner product $\langle [q_i], [q_j] \rangle_{\mathbb{H}}$, where $[q_i]$ and $[q_j]$ are the SRVFs of y_i and y_j , respectively.

Clustering is equivalent to finding an optimal partition of the data. We use $P = \{P_1, P_2, \dots, P_K\} \in \mathcal{P}$ to denote a partition of set $\{1, 2, \dots, n\}$ into K classes, where \mathcal{P} denotes the set of all partitions of $\{1, 2, \dots, n\}$. A partition P can also be represented by *membership indicators* $\{c_i, i = 1, \dots, n\}$, where $c_i = j$ if $i \in P_j, j = 1, \dots, K$, or a *membership matrix* $B \in \mathbb{R}^{n \times n}$, defined as $B_{ij} = \begin{cases} 1 & \text{if } c_i = c_j \\ 0 & \text{otherwise} \end{cases}$. If we assume: (1) observed shapes $\{y_i \in \mathcal{F}, i = 1, \dots, n\}$ come from several sub-populations, and (2) observed shapes from the same population are placed next to each other; one would expect to observe a block pattern in the inner product matrix S because the observations from the same cluster will have similar inner product. To perform Bayesian inference on the parameters, we define the following prior distribution on Σ that enables clustering of the observations. Motivated by (MacCullagh and Yang, 2008; Adametz and Roth, 2011; Vogt et al., 2010), consider the following decomposition of Σ . Let

$$\Sigma = \alpha I + \beta B \quad (3)$$

where $\alpha, \beta \in \mathbb{R}$, I is the identity matrix and $B \in \mathbb{R}^{n \times n}$ is the membership matrix. Equation (3) decomposes the scalar matrix Σ into a sparse matrix αI and a low-rank matrix βB , where B encodes the clustering information. For convenience of introducing a conjugate prior for α (Vogt et al., 2010), we re-parameterize this model into $\Sigma = \alpha(I + \theta B)$, where $\theta = \beta/\alpha$. Intuitively, the parameter θ controls the strength of similarity between two observations measured by their inner product - a large θ indicates a strong association, and vice versa. Refer to Fig. 4 for an illustration of the inner-product matrix S , membership matrix B and the corresponding Σ matrix.

In a clustering based on the inner-product matrix S via Bayesian inference, the primary goal is to infer the posterior distribution $P(B|S, \alpha, \theta, d)$ after specifying appropriate prior distributions. Below, we discuss specification of prior distributions for those parameters.

3.3. Priors and hyperpriors

A popular method of inducing a prior distribution on the space of partitions \mathcal{P} is the Chinese restaurant process (CRP) (Pitman, 2006) induced by a Dirichlet process (Ferguson, 1973, 1974). Since a prior on $\{c_i, i = 1, \dots, n\}$ induces a prior on \mathcal{P} and, hence, on the space of membership matrices B , it is enough to specify a prior on $\{c_i, i = 1, \dots, n\}$. We assume

$$P(c_n = j \mid c_1, \dots, c_{n-1}) = \begin{cases} \frac{n_j}{n-1+\xi} & \text{if } c_n = j \text{ for some } 1 \leq j \leq K \\ \frac{\xi}{n-1+\xi} & \text{otherwise,} \end{cases} \quad (4)$$

where $n_j = \#\{i : 1 \leq i < n, c_i = j\}$ and $\xi > 0$ is the precision parameter of the CRP. ξ controls the prior probability of introducing new clusters. The expected cluster size under CRP is given by $\sum_{i=1}^n \frac{\xi}{\xi+i-1} \sim \xi \log(\frac{\xi+n}{\xi})$.

3.3.1. Hyperpriors

We need to choose hyperpriors for parameters associated with the prior distributions.

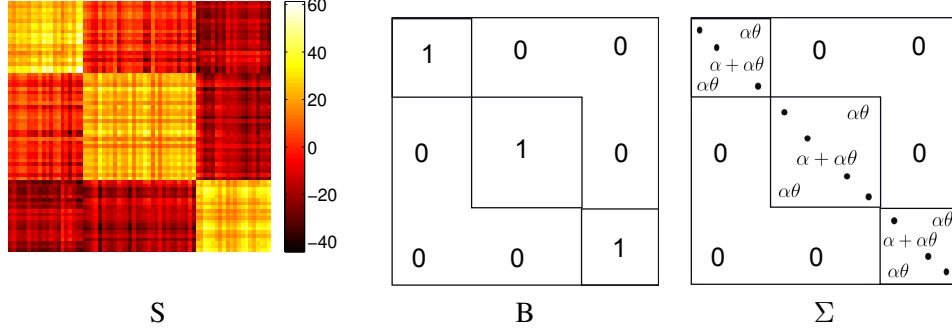


Figure 4: From the left to right: an inner product matrix S , a partition matrix B and a scale matrix Σ .

Priors on α and θ : α is assigned an inverse Gamma distribution, denoted $\alpha \sim \text{Inv-Gamma}(r, s)$ for constants $r, s > 0$. An inverse Gamma distribution for α allows us to marginalize out α in the posterior distribution, thus obviating the need to sample from its conditional posterior distribution in the Gibbs sampler (refer to Section 3.3.2). Recall that θ controls the strength of similarity within cluster. Thus a large θ will encourage tight clusters (elements in each cluster are very similar). We will explore the sensitivity of the final clustering to θ in Section 4. We assume a discrete uniform distribution for θ on the set $\{\theta_1, \dots, \theta_m\}$, with $P(\theta = \theta_i) = \frac{1}{m}, i = 1, \dots, m$.

Priors on ξ and d : Recall that the ξ controls the prior probability of introduction of new clusters in the CRP (4). We start with an initial guess of the number of clusters C_0 using standard algorithms for shape clustering (Yankov and Keogh, 2006; Auder and Fischer, 2012). In our experience, $C_0/\log n$ provides reasonable choice for ξ .

Also, recall that d is the degrees of freedom for the Wishart distribution. Since d represents the rank of the inner-product matrix S , it is natural to estimate d using the number of largest eigenvalues of S which explains 95% of the total variation. This forms an empirical Bayes estimate of d , denoted d_{EB} . Let the eigenvalues of S to be $\{\lambda_1, \dots, \lambda_m\}$, where $m \leq n$ and $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$. d_{EB} is taken to be the smallest integer such that $\frac{\sum_{j=1}^{d_{EB}} \lambda_j}{\sum_{i=1}^m \lambda_i} \geq 0.95$.

3.3.2. Posterior computation and final selection of clusters

Next, we develop a Gibbs sampling algorithm to sample from the posterior distribution of the unknown parameters. To that end, we propose the following simplifications to the likelihood. The trace and determinant that involve the α and θ in equation (1) can be computed analytically (Vogt et al., 2010; Adametz and Roth, 2011). Observe that

$$|\Sigma^{-1}| = \alpha^{-n} \prod_{j=1}^J (1 + \theta n_j)^{-1} \quad (5)$$

n_j is the number of elements in j^{th} cluster. Clearly, the j^{th} cluster corresponds to the j^{th} diagonal block in B ; refer to Fig. 4(a). Let $S_{jj}, j = 1, \dots, J$ be a *sub-square-matrix* in S corresponding to the j^{th} diagonal block in B , and $\bar{S}_{jj} = \mathbf{I}_j^t S \mathbf{I}_j$. Let $\mathbf{I}_j \in \mathbb{R}^{n \times 1}$ be such that the i^{th} element is $\mathbf{1}(c_i = j)$ for $i = 1, \dots, n$. Then

$$\text{tr}(\Sigma^{-1} S) = \sum_{j=1}^J \frac{1}{\alpha} \left\{ \text{tr}(S_{jj}) - \frac{\theta}{1 + n_j \theta} \bar{S}_{jj} \right\} = \frac{1}{\alpha} \left\{ \text{tr}(S) - \sum_{j=1}^J \frac{\theta}{1 + n_j \theta} \bar{S}_{jj} \right\}. \quad (6)$$

Table 1: Suggestions for specifying hyper-priors in our model.

Hyper-parameters	Description	Suggested values
θ	Parameter for Σ , $\Sigma = \alpha(I + \theta B)$	$\theta \sim \text{Uniform}(\theta_1, \dots, \theta_n)$ large θ - tight clusters, small θ - loose clusters
α	Parameter for Σ	$\alpha \sim \text{Inv-Gamma}(r, s)$, where r, s are constants
d	Degrees of freedom of Wishart	Estimated using the rank of S
ξ	Parameter for CRP	$\xi = K / \log(n)$, K is initial estimated # of clusters

Substituting (5) and (6) in (1) with (3), we obtain,

$$P(S | B, \alpha, \theta, d) \propto \alpha^{-nd/2} \prod_{j=1}^J (1 + \theta n_j)^{-d/2} \exp \left[-\frac{d}{2\alpha} \left\{ \text{tr}(S) - \sum_{j=1}^J \frac{\theta}{1 + n_j \theta} \bar{S}_{jj} \right\} \right]. \quad (7)$$

If $\alpha \sim \text{Inv-Gamma}(r_0 d/2, s_0 d/2)$ it is possible to integrate out α analytically in (7) as $P(S | B, \theta, d) = \int P(\alpha) P(S | B, \alpha, \theta, d) d\alpha$ yielding

$$P(S | B, \theta, d) \propto \prod_{j=1}^J (1 + \theta n_j)^{-d/2} \left[\frac{d}{2} \left\{ \text{tr}(S) - \sum_{j=1}^J \frac{\theta}{1 + n_j \theta} \bar{S}_{jj} + s_0 \right\} \right]^{-(n+r_0)d/2}. \quad (8)$$

Using the prior distributions for θ, B with the d_{EB} plugged in the likelihood (7), we get the posterior distribution of the membership matrix B :

$$P(B|S, \theta, d, \xi) \propto P(S|B, \theta, d_{EB}) P(B|\xi) P(\theta). \quad (9)$$

Fig. 5 shows the graphical model representation of our Bayesian model. Some suggestions of specifying hyper-priors are summarized in Table 1. We use Markov chain Monte Carlo (MCMC) algorithm to obtain posterior samples $B^{(1)}, \dots, B^{(M)}$ for a suitable large integer $M > 0$ using (9). The detailed algorithm is described in the following.

Algorithm 1. Posterior sampling using the MCMC

Given the prior parameters $d_{EB}, r, s, \xi, \theta$, and the inner-product matrix S from the n observations, and let $N_\theta = \text{length}(\theta)$, we want to sample $Iter$ number of posterior samples of membership matrices B :

1. Initialize the cluster number K (a large integer), the cluster indices $\{c_i, i = 1, \dots, n\}$ and obtain the initial membership matrix $B^{(0)}$;
2. **For** each sweep of the MCMC ($it = 0$ to $Iter$)
 - (a) **For** each $\theta_i, i = 1, \dots, N_\theta$, obtain posteriors $P(\theta_i | \cdot) \propto P(S | \theta_i, B^{(it)}, d_{EB}) p(\theta_i)$ using (8). Normalize $\{P(\theta_i | \cdot)\}$ and sample θ^{it} from the discrete distribution on the support points $\theta_i, i = 1, \dots, N_\theta$ with probabilities $\{P(\theta_1 | \cdot), \dots, P(\theta_{N_\theta} | \cdot)\}$. The complexity for this step is $O(N_\theta * k_{B^{(it)}})$, where $k_{B^{(it)}}$ is the number of clusters obtained from $B^{(it)}$.
 - (b) **For** each observation ($i = 1$ to n)
 - i. **For** each cluster ($j = 1$ to $k_{B^{(it)}} + 1$)

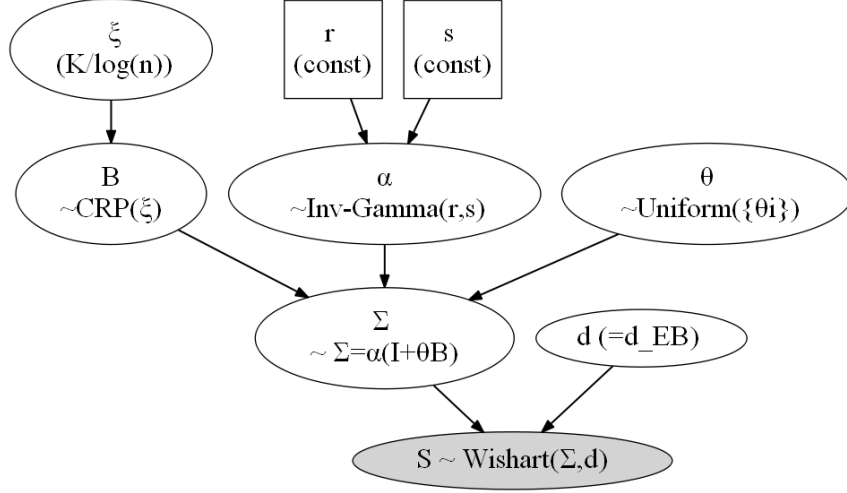


Figure 5: Graphical model representation of our Bayesian model. Squares indicate fix parameters and circles indicate random variables.

- A. Assign current observation (y_i) to the j -th cluster, update the membership matrix $B^{(it)}$ to B'_{it} , and calculate the posterior $\pi_j = P(B'_{it}|S, \theta^{it}, d_{EB}, \xi)$ using (9). The complexity for this step is $O(1)$ ¹.
- ii. Normalize $\{\pi_1, \dots, \pi_{k_{B^{(it)}}+1}\}$ and sample c_i from a discrete distribution on support points $\{1, 2, \dots, k_{B^{(it)}} + 1\}$ with probabilities $(\pi_1, \dots, \pi_{k_{B^{(it)}}+1})$. Update $B^{(it)}$. Complexity for this step is $O\{\log(k_{B^{(it)}})\}$ (Bringmann and Panagiotou, 2012).
- (c) After completing Step 2b, we obtain one MCMC sample of $B^{(it)}$.
3. Repeat step 2 so that we have Iter many samples. Discard the first few samples (burn-in), and relabel the remaining $B^{(it)}$ s as $B^{(1)}, \dots, B^{(M)}$.

From algorithm 1, the complexity of each sweep of the MCMC is $O\{N_\theta K + nK \log K\}$. Usually $N_\theta \leq n$, leading to an overall complexity of $O(nK \log K)$.

Once we obtain the posterior samples $\{B^{(i)}, i = 1, \dots, M\}$, our goal is to estimate the clustering configuration. However, the space of membership matrices B is very huge, and we would expect the posterior to explore only an insignificant fraction of the space based on a moderate values of M . Therefore, instead of using the mode of $\{B^{(i)}, i = 1, \dots, M\}$, we devise the following alternate strategy to estimate the clustering configuration more accurately. We treat the set of the membership matrices, denoted as \mathcal{F}_B , as a subset of symmetric $n \times n$ matrices with restrictions: (1) $B(i, j) = \{0, 1\}$ for all $i, j = 1, \dots, n$; (2) $B(i, \cdot) = B(j, \cdot)$ and $B(\cdot, i) = B(\cdot, j)$ if i^{th} observation and j^{th} observation are in the same cluster. The final matrix B^* is obtained by calculating the **extrinsic mean** of the posterior samples defined as follows.

Algorithm 2. Calculating extrinsic mean of membership matrices

Given the samples $B^{(1)}, \dots, B^{(M)}$, the extrinsic mean B^* is calculated as the followings:

¹Since only one observation changes the cluster index, one can explicitly calculate the difference between the old values of (5) and (6) and new values in $O(1)$ steps.

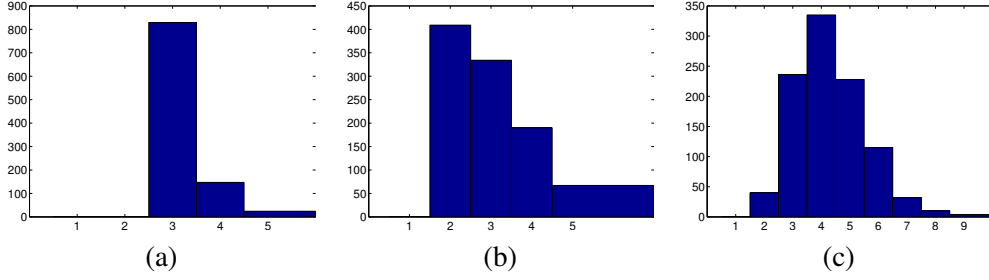


Figure 6: Posterior distribution of clustering number k .

1. Find the mode of the number of clusters k_0 based on the samples $B^{(1)}, \dots, B^{(M)}$.
2. Calculate the Euclidean mean and project it onto the membership matrix space:
 - (a) **Euclidean mean:** Let $\bar{B} = \frac{1}{M} \sum_{t=1}^M B^{(t)}$.
 - (b) **Projection:** Project the Euclidean mean onto the space of membership matrices by a thresholding operation $B^* = \text{threshold}(\bar{B}, t^*)$ where t^* is the largest threshold such that B^* has k_0 clusters. Setting $k = N$ and $\text{iter} = M$, the projection procedure involving the thresholding operation is described below:

While ($k \neq k_0$), **do**

 - i. Set $J_{\text{array}} = \{1, \dots, N\}$, $B^* = \text{zeros}(N, N)$. Also set $\text{iter} = \text{iter} - 1$, let $t^* = \text{iter}/M$.
 - ii. **For** j in J_{array} , **calculate**
 - A. $\mathbf{v} = \mathbf{1}(\bar{B}(j, \cdot) > t^*)$; record the index of elements in \mathbf{v} equal to 1, denoted as set C .
Let $J_{\text{array}} = J_{\text{array}} - C$, which means remove elements in C from J_{array} .
 - B. **For** i in set C , set $B^*(i, \cdot) = \mathbf{v}$ and $B^*(\cdot, i) = \mathbf{v}^t$.
 - iii. Set $k = \#B^*$, which is number of clusters in B^* .

Fig. 6 shows some generic illustrations of the posterior distribution on the number of clusters obtained from $B^{(1)}, \dots, B^{(M)}$. One may notice that the Euclidean mean $\bar{B} \notin \mathcal{F}_B$. Actually $\bar{B}(i, j)$ represents the posterior probability that the i^{th} and the j^{th} observations are clustered together. To project \bar{B} into \mathcal{F}_B , we find the largest value t^* to threshold \bar{B} , such that the thresholded \bar{B} , denoted as B^* , has k_0 clusters, and $B^* \in \mathcal{F}_B$. In other words, we assign two data points to the same cluster if the posterior probability of being clustered together is greater than or equal to t^* .

4. Experimental Results

In this section, we demonstrate the performance of our model (Wishart-CRP, denoted by W-CRP) both on synthetic data (in Section 4.1) and the case studies (in Section 4.2) introduced earlier in Section 2. For the Euclidean datasets, we generated 8000 samples from the posterior distribution and discarded a burn-in of 1000, whereas those numbers for the non-Euclidean data are 4000 and 1000, respectively. Convergence was monitored using trace plots of the deviance as well as several parameters. The high effective sample size of the main parameters of interest shows good mixing of the Markov chain. Also we get essentially identical posterior modes of with different starting points and moderate changes to hyperparameters.

4.1. Synthetic Examples

We consider several simulation settings for both Euclidean and non-Euclidean datasets.

4.1.1. Euclidean Data

Elicitation of hyperpriors:

In the Euclidean case, the data are generated from a Gaussian mixture model $p(x | \lambda) = \sum_{i=1}^{k_0} w_i g(x | \mu_i, \Sigma_i)$, $x \in \mathbb{R}^2$, with the mixing weights w_i , $i = 1, 2, 3$ and the component-specific Gaussian densities $g(x | \mu_i, \Sigma_i)$, $i = 1, 2, 3$. In the first experiment, we perform a sensitivity analysis to the choices of hyperpriors θ and ξ in the W-CRP formulation. Recall from Section 3.3.1 that (1) θ controls the strength of association between elements within clusters; (2) ξ is the concentration parameter for the CRP which controls the prior probability of producing new clusters. In this experiment, we test sensitivity to θ and ξ , keeping all the remaining parameters fixed ($d_{EB} = 2$, $r = 3$, $s = 4$). Fig. 7 shows the clustering results on three different 2D Euclidean datasets with different θ 's and ξ 's. Observations in Dataset 1 are clearly separable into three classes; Dataset 2 contains observations that can either be clustered into two or three classes; and Dataset 3 contains observation that shows no clear congregation of observations.

The histograms in Fig. 7 show posterior distributions of cluster number K . In the upper left panel, results are provided for $\xi = 0.2$ and $\theta \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. In the upper right panel, we use $\xi = 10$ and the same $\theta \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. Clearly, when $\xi = 10$, we have more clusters with small sizes, although the dominant clusters appear to be similar to the case of $\xi = 0.2$. In the lower left panel, we set $\xi = 0.2$ and $\theta \in \{1000, 2000, 3000, 4000, 5000\}$. Comparing this with the result in the upper left panel, we can see that larger θ values lead to more but tighter clusters. This conforms to our intuition about the role of θ . In the lower right panel, we used large $\xi = 10$ and large $\theta \in \{1000, 2000, 3000, 4000, 5000\}$. Comparing this with the one in upper right panel, we find that when θ is large, the estimate of the number of clusters is less sensitive to ξ than with a smaller value of θ .

The next experiment presents a more detailed sensitivity analysis to the choices of ξ and θ . We perform the cluster analysis using our method with all possible combinations of $\xi \in \{0.1, 0.2, 0.4, 0.8, 1, 2, 5, 10, 12, 15\}$ and $\theta \in \{0.1, 0.2, 0.5, 1, 5, 10, 20, 50, 100, 200, 500, 1000\}$. Fig. 8 shows a heat-map of the matrix of the estimated number of clusters for different values of ξ (across the rows) and θ (across the columns). To conclude: (1) The posterior of cluster number K is robust to the choice of ξ and θ in a large range. (2) For a fixed θ , larger values of ξ tend to produce more clusters (along with smaller clusters with only a few observations); and a small ξ tends to produce less clusters consistent with the data. (3) For a fixed ξ , a small θ tends to produce more clusters and a large θ tends to produce less clusters. The conclusions are in line with the roles of ξ and θ : (1) ξ controls the probability of introducing new clusters. Hence, for a very large ξ , the posterior tends to have more smaller extraneous clusters. We recommend choosing $\xi = \tilde{K}/(\log(n))$, \tilde{K} is a rough estimate of the number of clusters. Such a choice does not overestimate the value of ξ and produces less extraneous clusters. (2) θ controls the strength of association between elements within clusters. A large θ indicates a strong association, and tends to have more and tighter classes. However, θ tends to have a weak influence on the clustering result if ξ is not very large.

In a very recent technical report (Miller and Harrison, 2013), it is shown that Dirichlet process mixture (DPM) of Gaussians leads to an inconsistent estimation of the number of clusters. Instead, mixture of finite mixtures model (MFM) (Miller and Harrison, 2013) can guarantee the consistent estimation of the number of clusters. In next experiment, we compare with mixture of finite mixtures (MFM) prior on the same data as in Fig. 7. For a given finite mixture of Gaussians with number of components K , the MFM model assumes a Dirichlet distribution, $\text{Dir}(\alpha, \dots, \alpha)$ conditionally on K which assigned a $\text{Poisson}(\lambda)$ distribution. Fig. 9 shows the clustering results with MFM prior, where we set $\lambda = 2$. The left panel shows results using $\theta \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and the right panel shows results using $\theta \in \{1000, 2000, 3000, 4000, 5000\}$. The clustering results of CRP and MFM are very similar.

For comparisons in the Euclidean case, we use the Dirichlet process mixture (DPM) of Gaussians

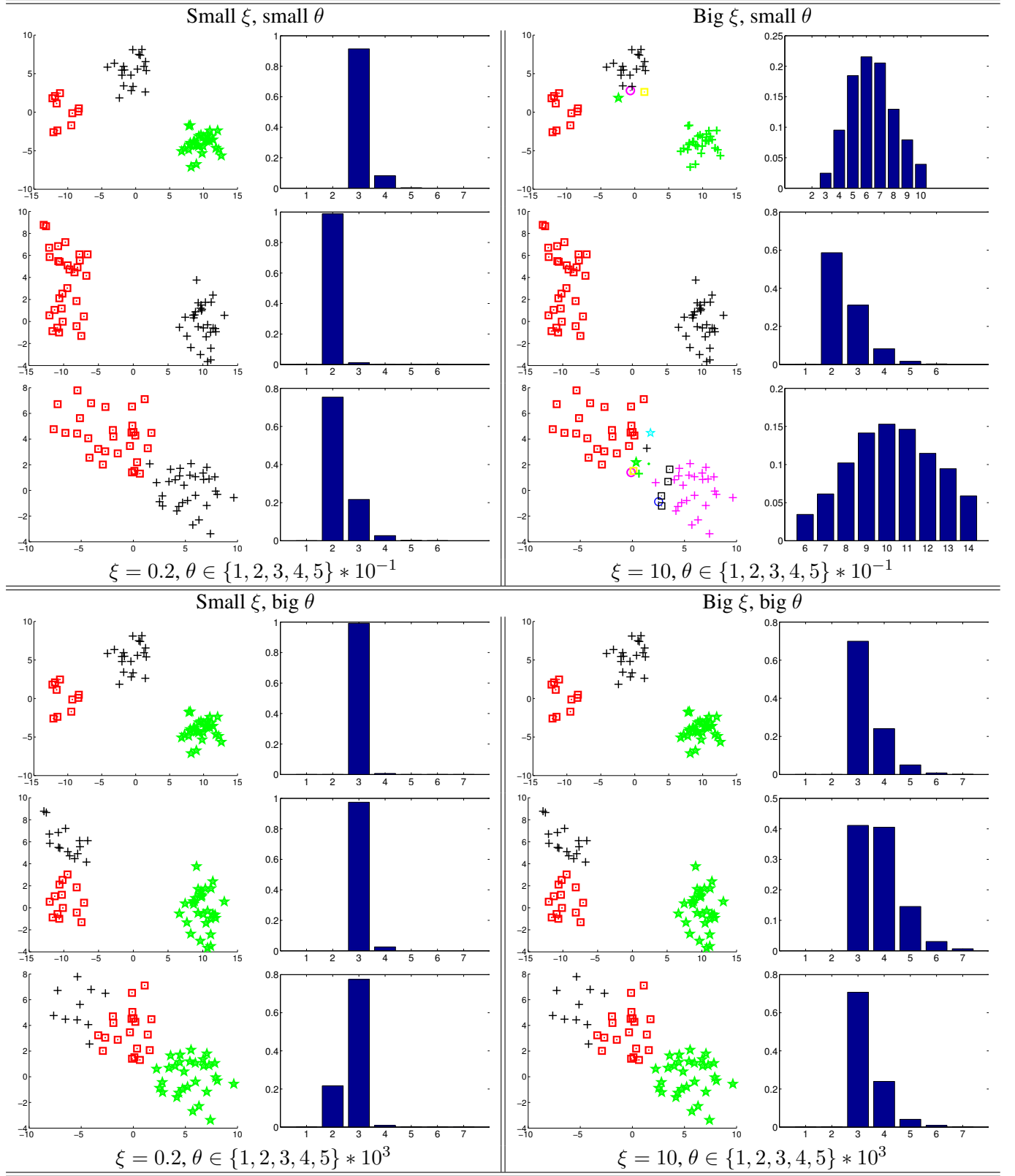


Figure 7: Euclidean data clustering results under different priors on θ and ξ .

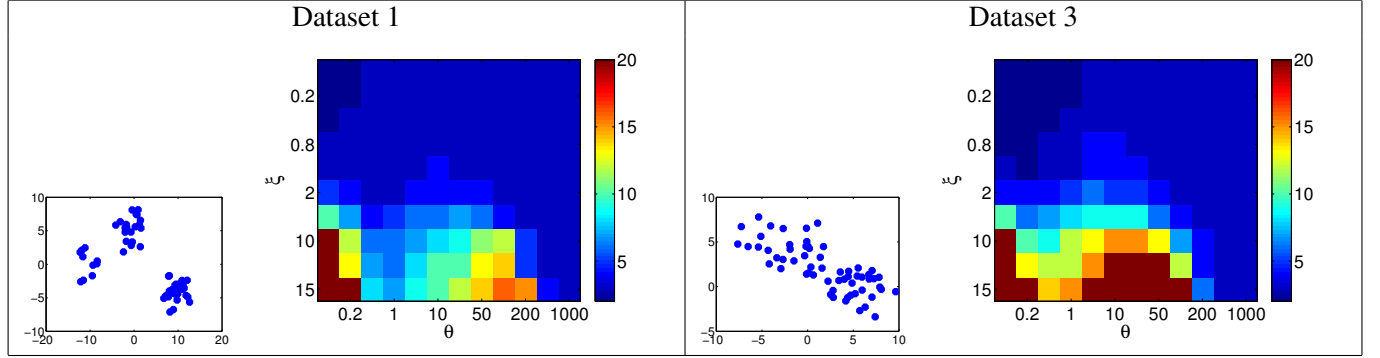


Figure 8: Number of clusters under different θ and ξ . The columns of the heat-map matrix are corresponding to θ ($\theta \in \{0.1, 0.2, 0.5, 1, 5, 10, 20, 50, 100, 200, 500, 1000\}$) and the rows are corresponding to ξ ($\xi \in \{0.1, 0.2, 0.4, 0.8, 1, 2, 5, 10, 12, 15\}$).

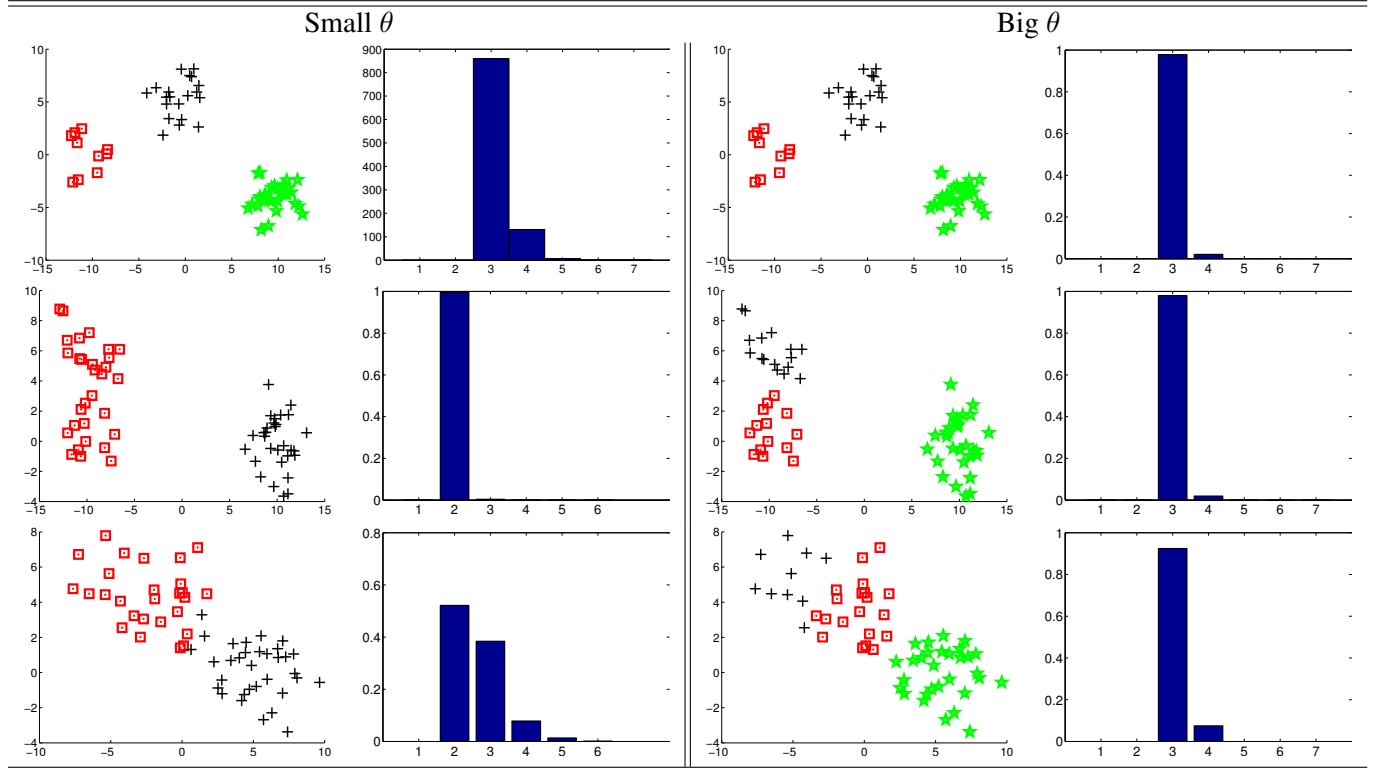


Figure 9: Euclidean data clustering results using our model with MFM prior. We use $\lambda = 2$ and different θ in this experiment. The left part results are using $\theta \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, and the right part results are using $\theta \in \{1000, 2000, 3000, 4000, 5000\}$.

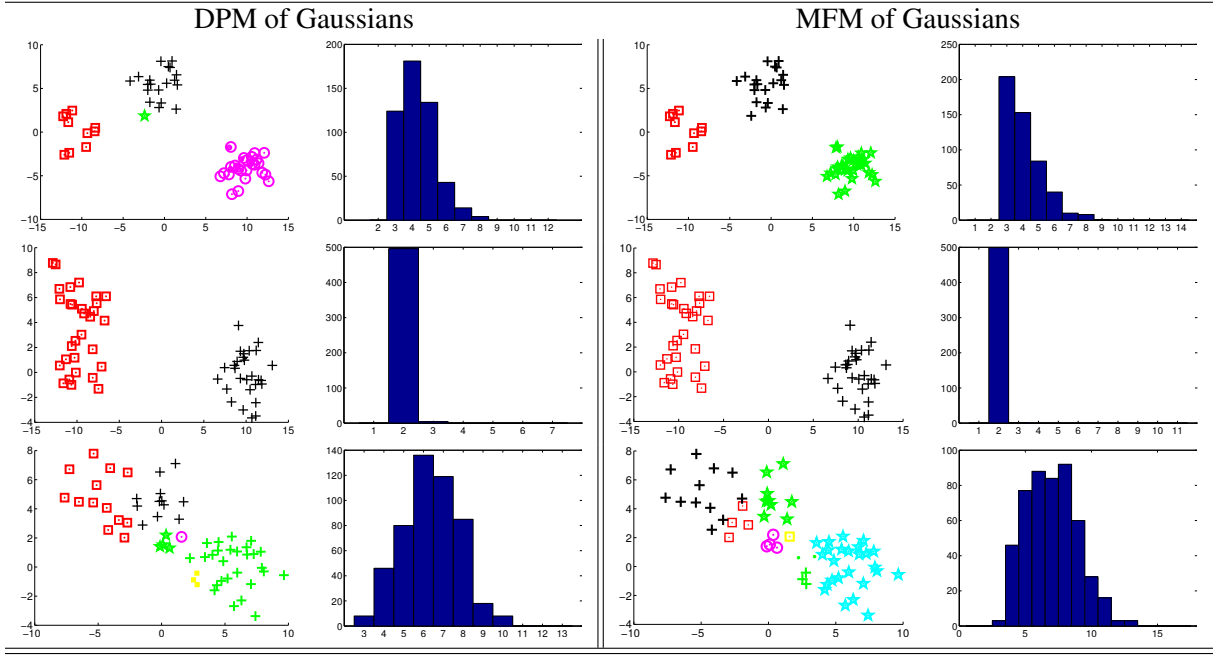


Figure 10: Clustering with DPM of Gaussians and MFM of Gaussians. The left panel shows the results from DPM of Gaussians and the right panel shows the results from MFM of Gaussians.

(Heller and Ghahramani, 2005) and MFM of Gaussians directly on the observations $y_i, i = 1, \dots, n$. Note that, unlike in the non-Euclidean case, the mean and covariance matrix can be efficiently obtained in the Euclidean case. Fig. 10 shows the results for the same dataset as in Fig. 7. The left panel shows the results with DPM of Gaussians and the right panel shows the results with MFM of Gaussians. One can observe from Fig. 10 that both the methods tend to produce extraneous clusters compared to W-CRP. This confirms the inconsistency of DPM as in (Miller and Harrison, 2013) and also demonstrates that the convergence of MFM is slow, although theoretically it might produce consistent estimates of the number of clusters when the sample size is large enough.

Non-Euclidean Shape Data:

In this experiment, we study shapes taken from the MPEG-7 database (Jeannin and Bober, 1999). The full database has 1400 shape samples, 20 shapes for each class. We first choose 100 shapes to form a subset of 10 classes with 10 shapes from each class. The observations are randomly permuted and the inner-product matrix S is calculated using Definition 1. Then we perform our clustering method on S . We impose a prior on α with Inv-Gamma(3, 4), and ξ is estimated by $\tilde{K}/\log(100)$, where \tilde{K} is an estimate of number of clusters ($\tilde{K} = 15$ in this case). The clustering result is shown in Fig. 11, where (a) and (b) shows the inner product (I-P) matrix before and after clustering, (c) shows the final clustering result, and (d) shows the histogram of cluster number K obtained from 4000 MCMC samples of B . From the result, one can see that our algorithm clusters these 100 shapes well other than splitting one class.

In next experiment, we analyze the sensitivity of the cluster number K to the parameter d , degrees of freedom of the Wishart distribution. Note that in the Euclidean case, d can be easily estimated since d (in the case of $d < n$) is the dimension of the data. Fig. 12 shows the estimated cluster number K versus the value of parameter d in the dataset shown in Fig. 11. It is evident that the estimates of K are robust to different choices of d .

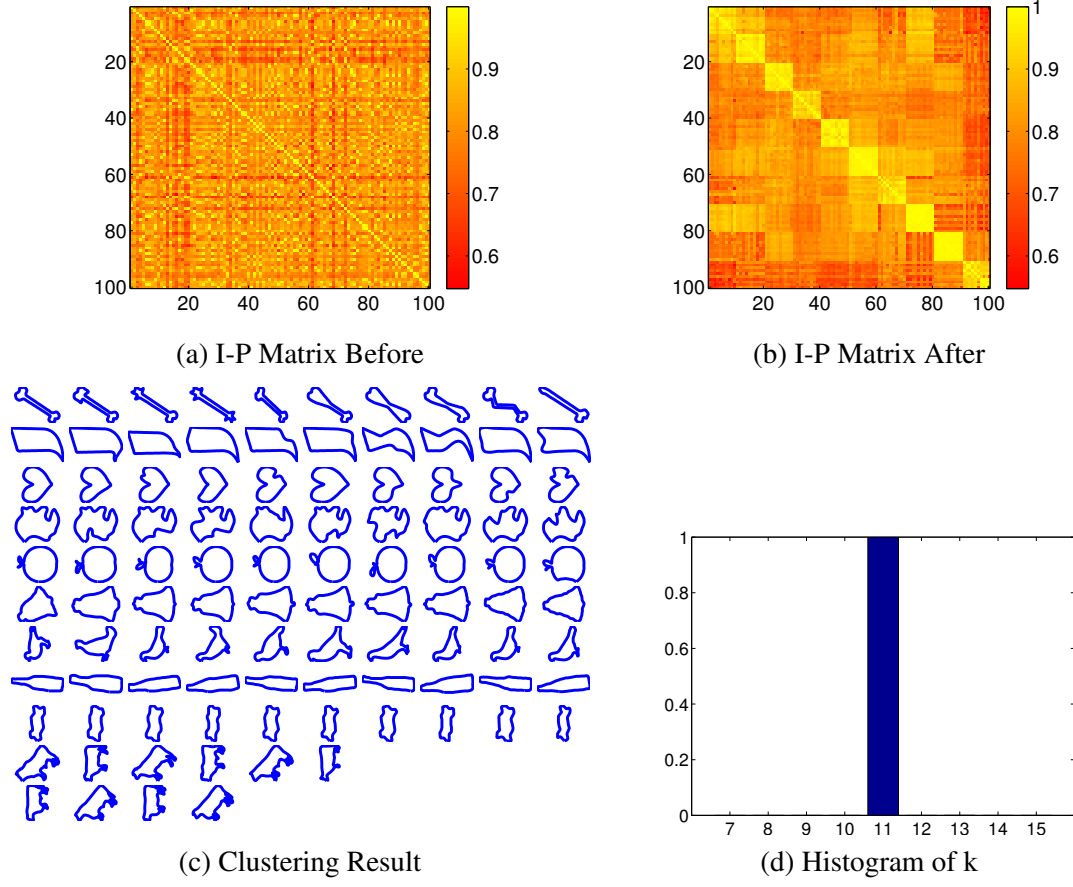


Figure 11: Clustering process for 100 shapes. The histogram shows the posterior distribution of the cluster number k obtained from 4000 MCMC samplings of B without any burn-in.

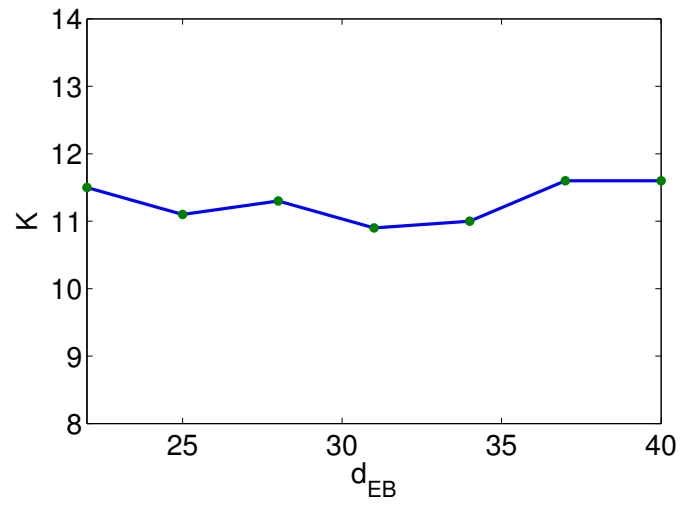


Figure 12: Clustering sensitivity analysis of parameter d in the dataset shown in Fig. 11.

Table 2: Comparison of the classification rate on MPEG-7 dataset

Classifier	FD+ SVM	HMM+ WtL	HMM+ OPC	K- medians	ESA+ PW	EIP+ W-CRP
Classification rate (%)	94.29	96.43	97.4	81.5	96.67	100.00
Rand index	-	-	-	0.91	0.98	1.00
Time (seconds)	-	-	-	648.5	707.4	774.2

To compare with existing methods in the shape domain, we test our method on another subset of MPEG-7 dataset that was used in (Bicego and Murino, 2004; Bicego et al., 2004; Bicego and Murino, 2007). The dataset contains 6 classes of shapes with 20 shapes per class. To quantify the clustering result, we use the “classification rate” defined in (Jain and Dubes, 1988). For each cluster, we note the predominant shape class, and for those shapes assigned to the cluster which do not belong to the dominant class are recognized to be misclassified. The classification rate is the total number of dominant shapes for all classes divided by the total number of shapes. However, this measure is known to be sensitive towards larger clusters. The Rand index (Torsello et al., 2007) is an alternative measure of the quality of classification which measures the similarity between the clustering result and the ground truth, defined as $RI = a / \binom{n}{2}$. Here a is the number of the “agreements” between the clustering and the ground truth, which is defined as the sum of two quantities: (1) the number of pairs of elements belonging to the same class that are assigned to the same cluster; (2) the number of pairs of elements belonging to different sets that are assigned to different classes. If the clustering result is the same as the ground truth, $RI = 1$, otherwise $RI < 1$. The Rand index penalizes the over-segmentation while the classification rate does not. Table 2 compares the overall classification rate and Rand index of our method with other methods, such as Fourier descriptor combined with support vector machine based classification (FD + SVM), hidden Markov model (HMM + WtL) with weighted likelihood classification (Bicego and Murino, 2007), HMM with OPC approach (HMM + OPC) (Bicego et al., 2004), elastic shape analysis (ESA) (Srivastava et al., 2011) with k-medians (K-medians), ESA with pairwise clustering method (ESA + PW) (Srivastava et al., 2005). Our model, with Wishart-CRP applied on the elastic inner product (EIP) matrix is denoted by EIP + W-CRP. The classification rate, Rand index and the computational time of K-medians, ESA + PW and our method are obtained based on the average of 5 runs on a laptop with a i5-2450M CPU and 8GB memory. The computational time of our approach (EIP + DW) includes the cost of calculating the inner-product matrix S (642.6 s) and generating the 4000 MCMC samples (131.6 s). A faster approach for calculating the elastic inner product matrix defined in our paper is available in (Huang et al., 2014). For ESA + PW and K-medians method, we set $K = 6$ since we know the true K in this case. The classification rates for FD+ SVM, HMM + WtL, and HMM + OPC are reported from (Bicego and Murino, 2007), and these rates are based on the 1-nearest neighbor classification. As evident from the results, our model can automatically find the cluster number $K = 6$, and the classification rate is better than the competitors.

4.2. Real Data Study

In this section, we cluster cell and protein shapes introduced in Section 2. Automated clustering is a crucial goal in real data applications where it is hard to provide a rough estimate of the number of clusters visually. In the examples where the ground truth labels are available, our methods provide higher classification rates.

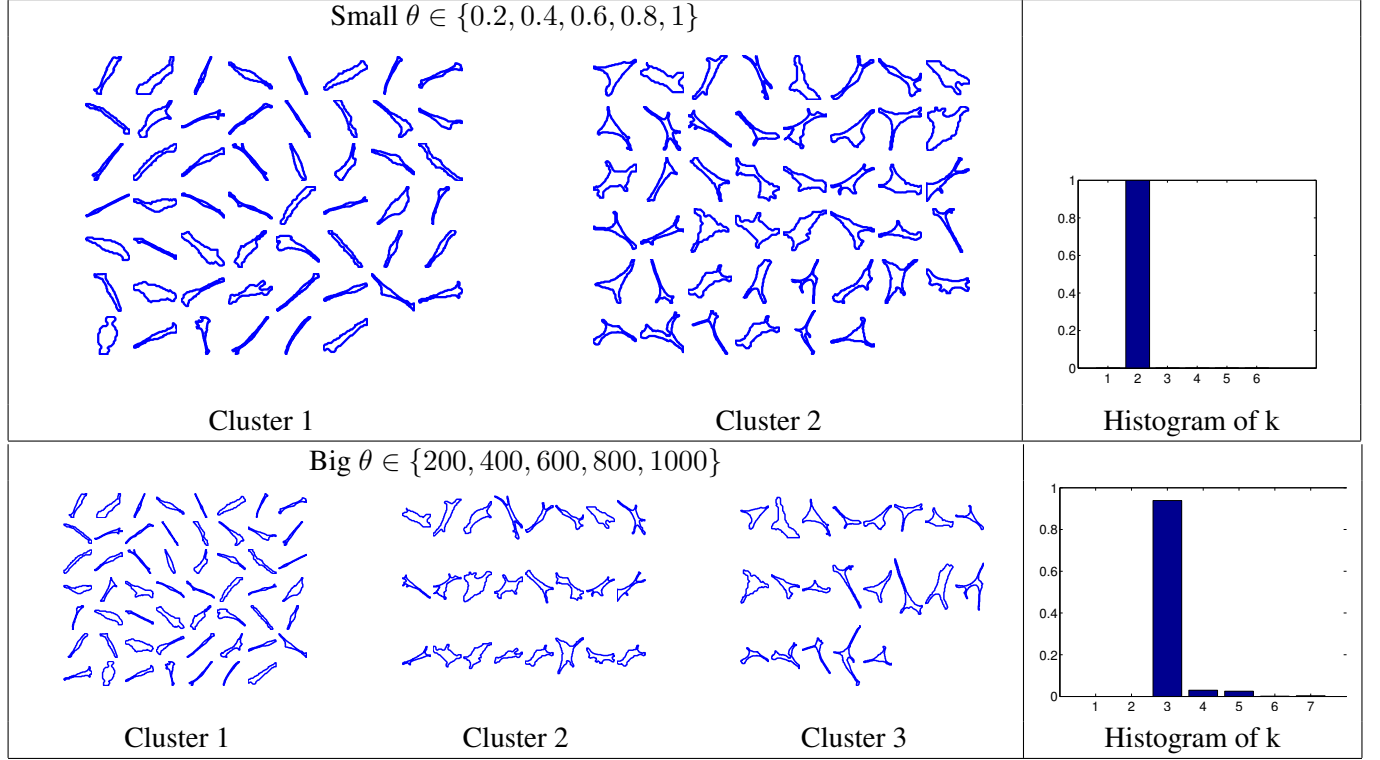


Figure 13: Clustering process for NIH-3T3 cell shapes with different θ values.

Cell shapes clustering: We first cluster the cell shape data introduced in Section 2.2. The cell shape dataset includes cell shapes from different cell types: DLEX-p46 cells, and NIH-3T3 cells. DLEX-p46 cells are round, whereas NIH-3T3 cells have an elongated, spindly appearance.

In the first experiment, we select 100 shapes from NIH-3T3 cell shapes to form a subset, and cluster them with different priors on θ : a set of small θ ($\theta \in \{0.2, 0.4, 0.6, 0.8, 1\}$) and a set of large θ ($\theta \in \{200, 400, 600, 800, 1000\}$). Fig. 13 shows the clustering result. The first row shows the clustering result with a small θ and the second row shows the result with a big θ . One can see that with a small θ , our method cluster the data into 2 classes: the first class of shapes only have two corners, and the second class has three or more corners. For a large θ , we cluster the shapes into three classes: one with shapes have two corners, one with shapes have four or more corners and one with shapes have three corners.

We pooled together 100 shapes of DLEX-P46 cells and 100 shapes of NIH-3T3 cells to form a set of 200 cells. Our goal is to cluster this pooled dataset into two classes: one with the DLEX-p46 cell shapes and the other with NIH-3T3 cell shapes. Since we expect a smaller number of clusters here, we use a set of small θ , $\theta \in \{0.1, 0.2, 0.3, 0.4\}$, $\xi = 1$ as our model parameters. The clustering result are shown in Fig. 14 first row, where each plot is a cluster. The 200 cell shapes are automatically clustered into 3 classes. Cell shapes of DLEX-p46 are plotted with thick blue lines, and NIH-3T3 cells are plotted with thin pink lines. The NIH-3T3 cell shapes have a large variance, thus our method separates them into 2 classes, one with long strip shapes, and the other with star shapes. Setting $K = 2$, we compared with ESA + PW method in the left panel of the bottom row and the K-medians method in the right. From the result, one can see that our method identifies 3 meaningful clusters instead of 2, and the clustering quality (both the classification rate and Rand index) also is higher than ESA + PW and K-median method. The classification rates for our

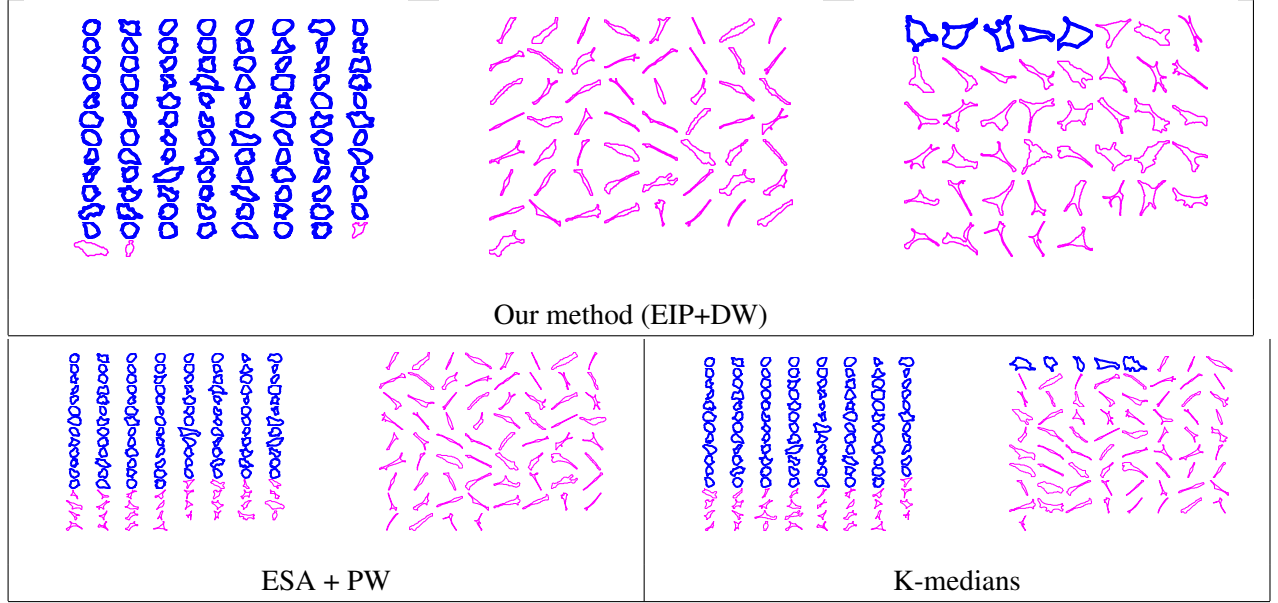


Figure 14: Clustering result for a subset of cell shapes containing both DLEX-P46 (thick blue) and NIH-3T3 (thin pink). The first row is the result of our method. The bottom left panel shows the result of ESA + PW and the bottom right shows the result of K-medians.

method, ESA+PW and K-medians are 96%, 83.5%, and 81.5%, respectively, and the Rand indexes are 0.82, 0.73, and 0.69, respectively.

Protein structure data clustering: In the following experiments, we will use our model to cluster the protein structure data introduced in Section 2.1.

In the first experiment, we choose a small protein structure dataset obtained from SCOP with only 88 proteins. Based on SCOP, these proteins are from 4 classes (SCOP provides the ground truth). Those proteins are pre-processed similar to an earlier study (Liu et al., 2011). To have a good estimate of the SRVFs from the raw data, we smooth the resampled protein structures with a Gaussian kernel. We also added one residue at both N and C terminal of each protein chain by extrapolating from the two terminal residues to allow some degrees of freedom on matching boundary residues. The added residues are removed after matching. Note that these smoothed SRVFs will only be used for searching optimal re-parameterizations γ and rotations $SO(3)$ to get the inner-product between protein structures. Then we apply our mixture of Wisharts model to the inner-product matrix S and get the clustering result, where we use parameters $\theta \in \{0.1, 0.2, 0.3, 0.4\}$ and $\xi = 1$. The final clustering results are shown in Fig. 15. The clustering rate is 100% compare with the ground truth provided by SCOP.

In next experiment, we choose 20 classes with at least 10 elements in each class from SCOP dataset to form a subset with 602 proteins. The final clustering result shows some clusters with only a few elements which we consider as *outliers*. In this experiment, our model identifies 17 outliers (7 small clusters). After removing these outliers, the remaining 585 proteins are clustered into 38 classes. The clustering rate is 84.1%. The first row in Fig. 16 shows the inner product matrix corresponding to the 585 protein structures (after putting elements in the same cluster together), and the posterior estimate of the partition matrix B . The second row shows first four clusters of the clustering result after the alignment (removing shape-preserving transformations). One can see that inside each cluster, the shapes of these protein structures are

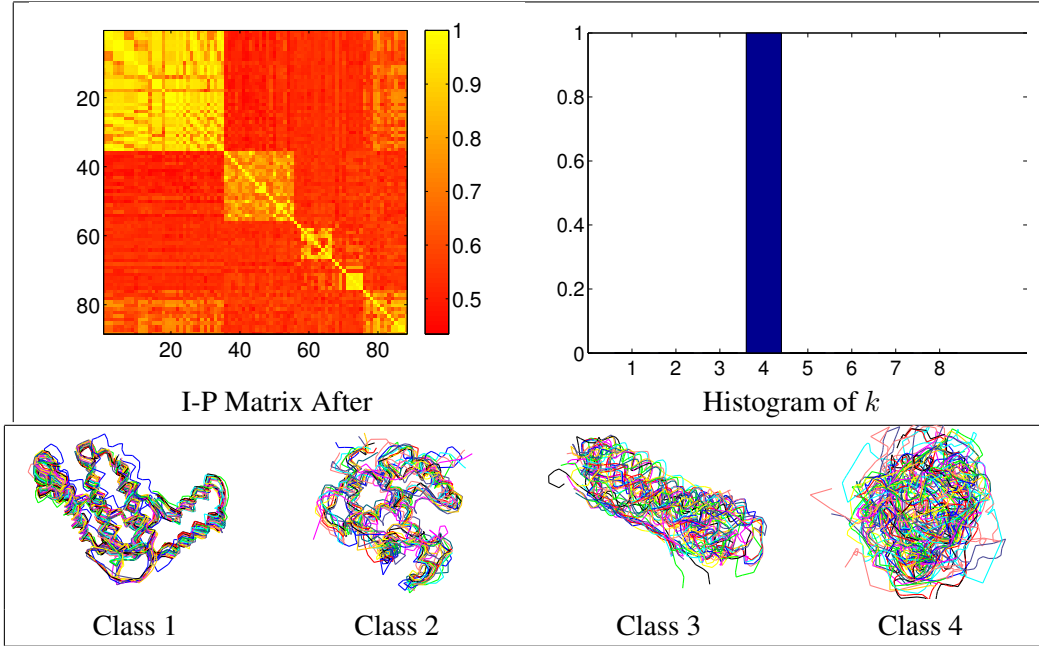


Figure 15: Protein structure classification on a SCOP subset containing 88 proteins. The first row shows the inner-product matrix between protein structures after clustering and the histogram of cluster number k . The second row shows the four clusters.

very similar to each other. As comparisons, we remove the outliers detected by our method, then apply ESA + PW and K-medians method to cluster the left 585 proteins by setting $K = 20$. ESA + PW gets 75.99% of classification rate and K-medians gets 63.42%. The Rand indexes for our model, ESA + PW and K-medians are 0.95, 0.93, and 0.91 respectively. As evident, we obtain a good clustering result based on only the shape of the proteins.

5. Conclusion

We have presented a Bayesian approach for clustering of shape data that does not require the number of clusters a priori. Instead, it assumes a flexible prior on the space of data partitions and studies the resulting posterior distribution on the clustering configuration. This prior is derived from a Dirichlet process (realized using the Chinese restaurant process) and the likelihood is given by the Wishart distribution. The Bayesian inference provides a reasonable solution for each of the simulated and real shape datasets studied in this paper. This fully automated method provides a way for shape-based partitioning of large datasets even in situations where visualization-based tools are not feasible.

References

- J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, volume 1, California, USA, 1967, p. 14.
- J. H. Ward, Hierarchical grouping to optimize an objective function, *Journal of the American Statistical Association* 58 (1963) 236–244.
- A. Y. Ng, M. I. Jordan, Y. Weiss, et al., On spectral clustering: analysis and an algorithm, *Advances in Neural Information Processing Systems* 2 (2002) 849–856.
- A. K. Jain, R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

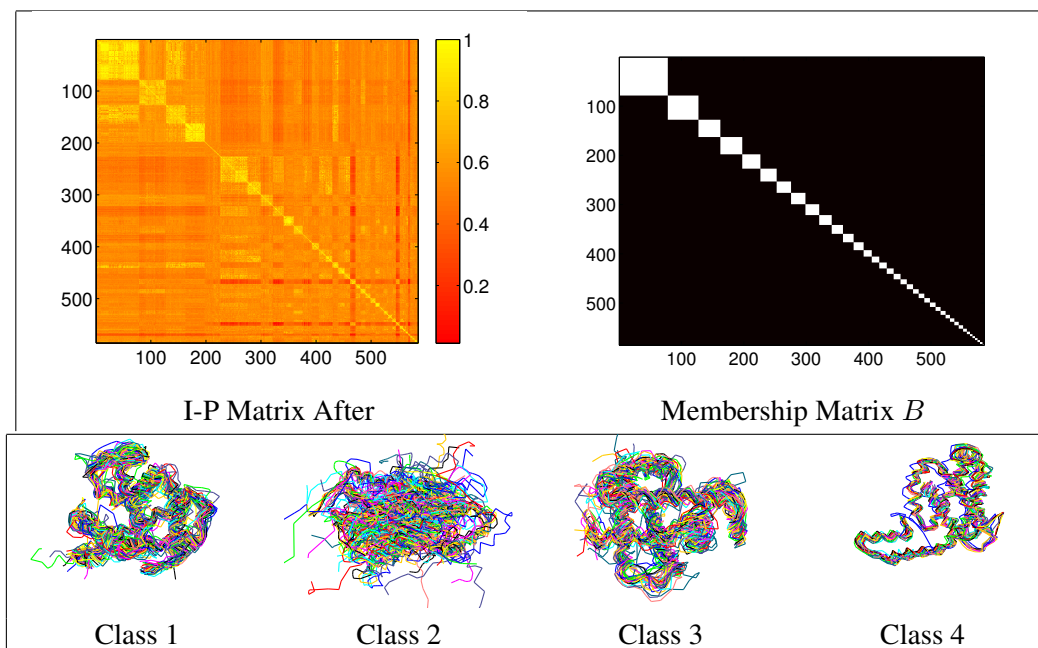


Figure 16: Protein structure classification on a SCOP subset of 602 proteins. The first row shows the inner-product matrix between protein structures after clustering and the corresponding inferred B , respectively. The second row shows the first four clusters.

- K. Ozawa, A stratificational overlapping cluster scheme, *Pattern Recognition* 18 (1985) 279 – 286.
- J. D. Banfield, A. E. Raftery, Model-based Gaussian and non-Gaussian clustering, *Biometrics* (1993) 803–821.
- C. Fraley, A. E. Raftery, How many clusters? Which clustering method? Answers via model-based cluster analysis, *The Computer Journal* 41 (1998) 578–588.
- C. Fraley, A. E. Raftery, Model-based clustering, discriminant analysis, and density estimation, *Journal of the American Statistical Association* 97 (2002) 611–631.
- C. Fraley, A. E. Raftery, MCLUST version 3: an R package for normal mixture modeling and model-based clustering, Technical Report, DTIC Document, 2006.
- P. MacCullagh, J. Yang, How many clusters?, *Bayesian Analysis* 3 (2008) 1–19.
- S. Kurtek, A. Srivastava, E. Klassen, Z. Ding, Statistical modeling of curves using shapes and related features, *Journal of the American Statistical Association* 107 (2012) 1152–1165.
- S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002) 509–522.
- M. Liu, B. C. Vemuri, S.-I. Amari, F. Nielsen, Shape retrieval using hierarchical total Bregman soft clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (2012) 2407–2419.
- D. Yankov, E. Keogh, Manifold clustering of shapes, in: *Proceedings of ICDM*, Washington, DC, USA, 2006, pp. 1167–1171.
- B. Auer, A. Fischer, Projection-based curve clustering, *Journal of Statistical Computation and Simulation* 82 (2012) 1145–1168.
- A. Srivastava, S. Joshi, W. Mio, X. Liu, Statistical shape analysis: clustering, learning, and testing, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2005) 590–602.
- S. Gaffney, P. Smyth, Joint probabilistic curve clustering and alignment, in: *Neural Information Processing Systems (NIPS)*, MIT Press, 2005, pp. 473–480.
- J. E. Vogt, S. Prabhakaran, T. J. Fuchs, V. Roth, The translation-invariant Wishart-Dirichlet process for clustering distance data, 2010, pp. 1111–1118.
- A. G. Murzin, S. E. Brenner, T. Hubbard, C. Chothia, SCOP: A structural classification of proteins database for the investigation of sequences and structures, *Journal of Molecular Biology* 247 (1995) 536 – 540.
- G. K. Rohde, A. J. S. Ribeiro, K. N. Dahl, R. F. Murphy, Deformation-based nuclear morphometry: capturing nuclear shape variation in hela cells, *Cytometry Part A* 73A (2008) 341–350.
- C. H. Thomas, J. H. Collier, C. S. Sfeir, K. E. Healy, Engineering gene expression and protein synthesis by modulation of nuclear

- shape, *Proc Natl Acad Sci* 99 (2002) 1972–1977.
- C. Hagwood, J. Bernal, M. Halter, J. Elliott, Evaluation of segmentation algorithms on cell populations using cdf curves, *IEEE Transactions on Medical Imaging* 31 (2012) 380–390.
- C. Hagwood, J. Bernal, M. Halter, J. Elliott, T. Brennan, Testing equality of cell populations based on shape and geodesic distance, *IEEE Transactions on Medical Imaging* 32 (2013) 2230–2237.
- J. D. Tucker, W. Wu, A. Srivastava, Generative models for functional data using phase and amplitude separation, *Computational Statistics & Data Analysis* 61 (2013) 50–66.
- D. Adametz, V. Roth, Bayesian partitioning of large-scale distance data, in: *Neural Information Processing Systems (NIPS)*, 2011, pp. 1368–1376.
- A. Srivastava, E. Klassen, S. H. Joshi, I. H. Jermyn, Shape analysis of elastic curves in Euclidean spaces, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33 (2011) 1415–1428.
- P. McCullagh, Marginal likelihood for distance matrices, *Statistica Sinica* 19 (2009) 631–649.
- J. Pitman, *Combinatorial stochastic processes*, volume 1875, Springer-Verlag, 2006.
- T. S. Ferguson, A Bayesian analysis of some nonparametric problems, *The Annals of Statistics* (1973) 209–230.
- T. S. Ferguson, Prior distributions on spaces of probability measures, *The Annals of Statistics* (1974) 615–629.
- K. Bringmann, K. Panagiotou, Efficient sampling methods for discrete distributions, volume 7391 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2012.
- J. W. Miller, M. T. Harrison, Inconsistency of Pitman-Yor process mixtures for the number of components, *arXiv preprint arXiv:1309.0024* (2013).
- K. A. Heller, Z. Ghahramani, Bayesian hierarchical clustering, in: *Proceedings of the International Conference on Machine learning (ICML)*, Bonn, Germany, 2005, pp. 297–304.
- S. Jeannin, M. Bober, Shape data for the MPEG-7 core experiment ce-shape-1 @ONLINE, 1999. URL: <http://www.cis.temple.edu/~latecki/TestData/mpeg7shapeB.tar.g>.
- M. Bicego, V. Murino, Investigating hidden Markov models’ capabilities in 2D shape classification, *IEEE Trans. Pattern Anal. Mach. Intell* 26 (2004) 281–286.
- M. Bicego, V. Murino, M. A. Figueiredo, Similarity-based classification of sequences using hidden Markov models, *Pattern Recognition* 37 (2004) 2281 – 2291.
- M. Bicego, V. Murino, Hidden Markov model-based weighted likelihood discriminant for 2D shape classification, *IEEE Trans. Pattern Anal. Mach. Intell* 16 (2007) 2707–2719.
- A. Torsello, A. Robles-Kelly, E. Hancock, Discovering shape classes using tree edit-distance and pairwise clustering, *International Journal of Computer Vision* 72 (2007) 259–285.
- W. Huang, K. Gallivan, A. Srivastava, P.-A. Absil, Riemannian optimization for elastic shape analysis, *Mathematical theory of Networks and Systems* (2014).
- W. Liu, A. Srivastava, J. Zhang, A mathematical framework for protein structure comparison., *PLoS Computational Biology* 7 (2011).