entropy

MDPI

Article

A Nonparametric Model for Multi-Manifold Clustering with Mixture of Gaussians and Graph Consistency

Xulun Ye, Jieyu Zhao * and Yu Chen

Institute of Computer Science and Technology, Ningbo University, Ningbo 315211, China; yexlwh@163.com (X.Y.); chenyu_cycy@126.com (J.Z.)
* Correspondence: zhao_jieyu@nbu.edu.cn; Tel.: +86-876-005-93

Received: 28 September 2018; Accepted: 23 October 2018; Published: 29 October 2018

Abstract: Multi-manifold clustering is among the most fundamental tasks in signal processing and machine learning. Although the existing multi-manifold clustering methods are quite powerful, learning the cluster number automatically from data is still a challenge. In this paper, a novel unsupervised generative clustering approach within the Bayesian nonparametric framework has been proposed. Specifically, our manifold method automatically selects the cluster number with a Dirichlet Process (DP) prior. Then, a DP-based mixture model with constrained Mixture of Gaussians (MoG) is constructed to handle the manifold data. Finally, we integrate our model with the *k*-nearest neighbor graph to capture the manifold geometric information. An efficient optimization algorithm has also been derived to do the model inference and optimization. Experimental results on synthetic datasets and real-world benchmark datasets exhibit the effectiveness of this new DP-based manifold method.

Keywords: multi-manifold clustering; Dirichlet process mixture model; mixture of Gaussians; graph theory

1. Introduction

Over the past decades, clustering has been the most fundamental task in many computer vision and data mining applications [1,2], e.g., image/motion segmentation [3,4], community detection [5], feature selection [6] and biological/network information analysis [7,8]. However, most of the conventional clustering methods assume that data samples are scattered in the feature space, which ignores the intrinsic underlying data structure that many real datasets have [3,9]. To overcome this problem, various manifold-based clustering (multi-manifold clustering) methods have been proposed and developed. Compared to the conventional clustering method, which regards the cluster as the data points with small distances between cluster members or dense areas of the feature space, the multi-manifold approach aims to gather the given data points into disparate groups, which come from different underlying submanifolds [10].

Unlike the conventional clustering methods [11,12], multi-manifold clustering can be classified into two different categories, the linear method and the nonlinear method [13]. In the first category, linear methods (also known as subspace clustering) construct the multi-manifold clustering by assuming that the underlying cluster can be well approximated by a union of low dimensional linear manifolds [14]. For example, Gholami [14] and Vidal [15] used a linear function to fit the underlying submanifold and cluster the clusters with the mixture model. Sparse Subspace Clustering (SSC)- [16], Low-Rank Representation (LRR)- [17] and Least Squares Regression (LSR)-based [18] methods approach the linear manifold clustering problem by finding a sparse representation of each point in terms of other data points. After forming a similarity graph with the learned sparse representation, spectral clustering methods are used to cluster data into distinctive clusters. As an expanding framework of the linear multi-manifold clustering methods, non-linear algorithms can be naturally applied to linear and/or nonlinear manifolds. For example, the K-manifold clusters the nonlinear subspace dataset by expanding the conventional K-means with geodesic distance [19]. Spectral Multi-Manifold Clustering (SMMC) integrates the local geometric information within the subspace clustering framework to handle the manifold structure [13], Multi-Manifold Matrix Decomposition for Co-clustering (M3DC) handles the manifold dataset by considering the geometric structures of both the sample manifold and the feature manifold simultaneously [20]. Recently, the state-of-the-art method may be deep subspace clustering, which assembles the deep framework and the conventional subspace clustering method [21,22].

However, a drawback of most conventional manifold clustering methods is that the clustering accuracy depends on the cluster number, which is always unavailable in advance [23]. To overcome this model selection problem, one category of the most widely-studied methods is that equipping the conventional methods with a Dirichlet process prior, e.g., Dirichlet Process Mixture (DPM) models [24,25], Multilevel Clustering with Context (MC^2) [26] and Dirichlet Process Variable Clustering (DPVC) [27]. Since the distributions adopted in these nonparametric models are defined in the Euclidean space, those conventional Dirichlet process clustering methods suffer difficulty when dealing with the manifold data. To overcome this problem, many manifold DP clustering models have been proposed. Wang [28] and Gholami [14] assumed that the submanifold is lying on the linear manifold and can be fitted with the hyperplane. Straub et al. [29,30] defined the Gaussian distribution on the sphere surface and introduced an auxiliary indicator vector zwith a DP prior. More than the sphere manifold, Simo et al. [31] expanded the distribution to the manifold space with the logarithmic and exponential mapping. Although these models are quite powerful and have been widely studied in many applications, they have their drawbacks when the manifold structure is not prespecified [31]. For example, the DP-space and temporal subspace clustering model is an expanding method of the linear manifold clustering method. It lacks the capability to handle a non-linear manifold dataset. In the geodesic mixture model, the logarithmic and exponential mapping algorithms [32,33] used in this model depend mainly on the pre-defined geometric structure, which is always unavailable. For the sphere mixture model, the sphere manifold has not been extended to arbitrary manifolds [31].

In this paper, we investigate the manifold clustering method with no prespecified manifold structure and cluster number in the DPM framework. In order to model the complicated manifold cluster distributions, we integrate the original DPM with the conventional Mixture of Gaussians (MoG) [34,35] to handle the manifold distribution (Figure 1a). Furthermore, we also notice that an unconstrained MoG distribution fails to capture the manifold geometrical information (Figure 1b). Inspired by the previous study [23,36], we regularize our model with a k-nearest neighbor graph. To form a meaningful cluster, in which samples from the same cluster are closed and related, we constrain the MoG mean with a Mahalanobis distance.

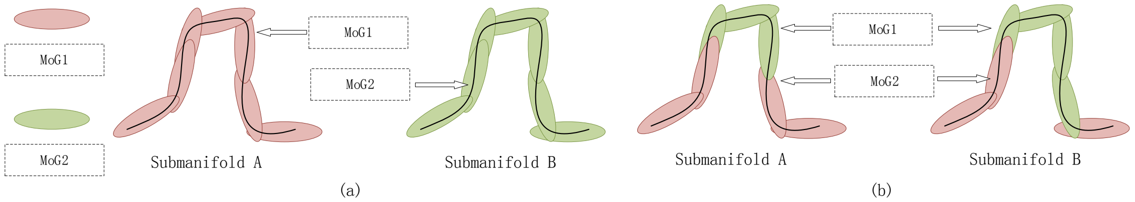


Figure 1. Illustration of manifold modeling. Mixture of Gaussians (MoG) distributions are demonstrated to model the submanifolds. (**a**) demonstrates two ideal results using two MoG distributions to model the submanifold; (**b**) demonstrates a result where there is no geometric information and mean constraint, in which Gaussian distributions in the two MoGs may be scattered into two submanifolds.

The main contributions are as follows:

- A constrained MoG distribution has been applied to model the non-Gaussian manifold distribution.
- We integrate the graph theory with DPM to capture the manifold geometrical information.
- The variational inference-based optimization framework is proposed to carry out the model inference and learning.

The organization of our paper proceeds as follows. In Section 2, we review the background knowledge of the Dirichlet process mixture model. Simultaneously, we present the generation procedure of the proposed manifold Dirichlet process mixture model and give the variational expectation maximization inference algorithm. Experimental comparisons will be presented in Section 3. In Section 4, we give the detailed discussions and present the limitations and advantages. Section 5 concludes the paper.

2. Materials and Methods

In this section, we firstly review the basic concept of the Dirichlet Process Mixture (DPM) model. Then, we propose the multi-manifold clustering method by equipping DPM with MoG and the k-nearest neighbor graph. In our method, the Dirichlet process is used to generate the suitable cluster number. MoG and the k-nearest neighbor graph are applied to model the non-Gaussian manifold distribution and capture the manifold geometric information. Finally, variational inference is derived to do the model inference and learning.

2.1. Dirichlet Process Mixture Model

The Dirichlet Process Mixture (DPM) model is an approach that extends the mixture model by introducing a Dirichlet process prior within the Bayesian framework. In DPM, we firstly sample a prior distribution G from the Dirichlet process and then sample the likelihood parameters $\{\boldsymbol{\theta}_n\}_{n=1}^{N}$ from G. With the sampled likelihood parameters, observation data x_n can be generated from the likelihood distribution $F(x|\boldsymbol{\theta}_n)$. This procedure can be concluded as follows:

$$\begin{aligned} & G|G_0(\boldsymbol{\lambda}) \sim DP(G_0(\boldsymbol{\lambda}), \alpha) \\ & \boldsymbol{\theta}_n|G \sim G \qquad n = 1,2,3,...,N \\ & \boldsymbol{x}_n \sim F(\boldsymbol{x}|\boldsymbol{\theta}_n) \qquad n = 1,2,3,...,N, \end{aligned} \tag{1}$$

where $F(\boldsymbol{x}|\boldsymbol{\theta}_n)$ is a likelihood distribution and G_0 is a base distribution. x_n is the observation sample.

By integrating out G, the joint distribution of the likelihood parameters $\{\boldsymbol{\theta}_n\}_{n=1}^{N}$ exhibits a clustering effect. Suppose that we have $N-1$ parameters $\{\boldsymbol{\theta}_n\}_{n=1}^{N-1}$ sampled from our Dirichlet process. We then have the following probability for the N-th value of $\boldsymbol{\theta}$.

$$p(\boldsymbol{\theta}_N|\{\boldsymbol{\theta}_n\}_{n=1}^{N-1}) = \frac{\alpha G_0}{\alpha+N-1} + \sum\nolimits_{i=1}^{I} \frac{n_i \delta(i)}{\alpha+N-1}, \tag{2}$$

where n_i denotes the $\boldsymbol{\theta}$ frequency of occurrence in $\{\boldsymbol{\theta}_n\}_{n=1}^{N-1}$ and $\delta(j)$ represents the delta function. I denotes the number of unique values in $\{\boldsymbol{\theta}_n\}_{n=1}^{N}$. (2) reveals the fact that a new sample $\boldsymbol{\theta}_n$ is either generated from a new cluster with probability G_0 or extracted from the existing clusters $\{\boldsymbol{\theta}_n\}_{n=1}^{N-1}$ with probability $n_i/(\alpha+N-1)$.

2.2. Our Proposed Method

In this section, we expand the original DPM model to a multi-manifold clustering framework named the Similarity Dirichlet Process Mixture (SimDPM) model. The main notations and descriptions used in our method are summarized in Table 1.

Table 1. The main notations and descriptions.

Notations	Descriptions
$\boldsymbol{\lambda}$	Hyper parameter $\boldsymbol{u}_0, c_0, \boldsymbol{W}_0, \nu_0$ of normal-Wishart
M	The mixture number
N	Number of the observation samples
$\boldsymbol{\theta}_k^*$	Gaussian parameter $\boldsymbol{u}, \delta$
$\tilde{\boldsymbol{\theta}}_k$	MoG parameter $\tilde{\boldsymbol{u}}_k, \tilde{\delta}_k, \tilde{\boldsymbol{\pi}}_k$
s	Tradeoff parameter
K	The maximum cluster number
α	Parameter of the Beta distribution
$z_n^{(1)}$	Class indicator of a Gaussian distribution
$z_n^{(2)}$	Class indicator of the MoG distribution
X	Unlabeled dataset
γ_k	Variational parameter $\gamma_{k,1}, \gamma_{k,2}$
$\boldsymbol{\tau}_k$	Variational parameter of normal-Wishart
$\boldsymbol{\Phi}_n$	Variational parameter of categorical distribution
A	An auxiliary parameter that equals $\boldsymbol{\Phi}$
$\boldsymbol{\lambda}_A$	Penalty parameter used in the graph Laplacian
L	Graph Laplacian
L^k	k-nearest neighbor graph
D^k	Diagonal matrix whose entries are column sums of L^k
R	Posterior penalty term with graph Laplacian
r	Neighbor number used in the graph Laplacian

As we have debated, DPM is unable to model the manifold dataset since the conventional likelihood distribution $F(\boldsymbol{x}|\boldsymbol{\theta}_n)$ is defined in the Euclidean space or prespecified manifold. To overcome this problem, we approximate the manifold distribution with MoG (Figure 1a). Then, we construct the sample generation process with two phases, a single Gaussian distribution and a mixture of Gaussians distribution. The reason we generate the data with both the single Gaussian distribution and the MoG distribution is that some simple submanifolds and non-manifold clusters can be modeled by the single Gaussian distribution.

Suppose that we are given N observation samples $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N$ where $x_n \in \mathbb{R}^D$. Given the additional parameters of the MoG distribution, we assume the following generative process for each observation data $\boldsymbol{x}_i$:

1. For $i = 1, 2, 3, ...$; draw $v_i|\alpha \sim Beta(1, \alpha)$
2. For $i = 1, 2, 3, ...$; draw $\boldsymbol{\theta}_i^*|G_0 \sim G_0$
3. For every data point i:
   (a) Choose $z_i^{(1)}|\boldsymbol{v} \sim mult(s\boldsymbol{\pi}(\boldsymbol{v}))$
   (b) Choose $z_i^{(2)}|\boldsymbol{v} \sim mult((1-s)\pi(\boldsymbol{v}))$
   (c) Draw $\boldsymbol{x}_i|z_i^{(1)} \sim N(\boldsymbol{x}|\boldsymbol{\theta}^*_{z_i^{(1)}})$
   (d) Draw $\boldsymbol{x}_i|z_i^{(2)} \sim MoG(\boldsymbol{x}|\tilde{\boldsymbol{\theta}}_{z_i^{(2)}})$

where $Beta(1, \alpha)$ is a beta distribution with parameter one and α, $mult(\boldsymbol{\pi}(\boldsymbol{v}))$ is a categorical distribution parameterized by $\boldsymbol{\pi}(\boldsymbol{v})$, $\boldsymbol{v}$ and $\boldsymbol{\pi}(\boldsymbol{v})$ are vectors with $\boldsymbol{v} = \{v_k\}_{k=1}^{\infty}$ and $\boldsymbol{\pi}(v_i) = v_i \prod_{j=1}^{i-1}(1 - v_j)$ and G_0 is a normal-Wishart distribution with parameter $\boldsymbol{\lambda} = (\boldsymbol{u}_0, c_0, \boldsymbol{W}_0, v_0)$ where

$\boldsymbol{u}_0 \in \mathbb{R}^D$ $\boldsymbol{W}_0 \in \mathbb{R}^{D\times D}$, $\boldsymbol{\theta}^*_{z_i^{(1)}}$ is a Gaussian distribution parameterized by $\boldsymbol{u}_{z_i^{(1)}} \in \mathbb{R}^D, \boldsymbol{\delta}_{z_i^{(1)}} \in \mathbb{R}^{D\times D}$. $\tilde{\boldsymbol{\theta}}_{z_i^{(2)}}$ is the MoG distribution with parameter $\tilde{\boldsymbol{u}}_{z_i^{(2)}} \in \mathbb{R}^D, \tilde{\boldsymbol{\delta}}_{z_i^{(2)}} \in \mathbb{R}^{D\times D}, \tilde{\boldsymbol{\pi}}_{z_i^{(2)}} \in \mathbb{R}^M$ where M denotes the mixture number in every $\tilde{\boldsymbol{\theta}}_{z_i^{(2)}}$. Tradeoff parameter s denotes how likely the observation sample x_n is sampled from a single Gaussian distribution. The corresponding probability graph model representation of manifold DPM can be described as Figure 2.

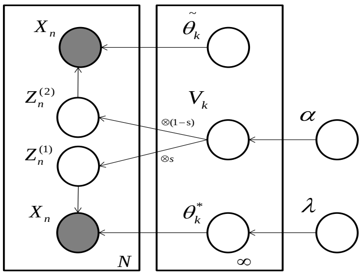


Figure 2. Probability Graph Model (PGM) representation of the Similarity Dirichlet Process Mixture (SimDPM) model. Nodes denote the random variables. In our framework, observations are generated from two phases, a fully-Bayesian procedure and a constrained MoG model.

To form a meaningful cluster (samples from the same cluster are closely related) and respect the manifold geometrical information, we constrain the MoG mean with:

$$\frac{1}{2}(\tilde{\boldsymbol{u}}_{k,m} - \tilde{\boldsymbol{u}}_{k,m-1})^T \tilde{\boldsymbol{\delta}}_{k,m}^{-1} (\tilde{\boldsymbol{u}}_{k,m} - \tilde{\boldsymbol{u}}_{k,m-1}) < \epsilon,\ m > 1,$$

and use a k-nearest neighbor graph to regularize the posterior probability inspired by [23], in which the graph Laplacian is used to capture the geometric information that has been missed by the MoG distribution.

$$R = \sum_{k=1}^{\infty} p(k|\boldsymbol{X})^T L p(k|\boldsymbol{X}), \tag{3}$$

where $p(k|\boldsymbol{X}) = \{p(k|\boldsymbol{x_n})\}_{n=1}^N$ is the posterior probability. L is the graph Laplacian constructed by the k-nearest neighbor graph [13]. Note that the constraint of $\tilde{\boldsymbol{u}}_{k,m}$ depends only on the previous $\tilde{\boldsymbol{u}}_{k,m-1}$, but not on $\tilde{\boldsymbol{u}}_{k,m+1}$. Below, we characterize the k-nearest neighbor graph L^k. Given the unlabeled data X, for any point x_i, we sort the rest of the data samples and select the top-k nearest neighbors. If node x_j is in the top-k nearest points of node x_i, we set:

$$L_{i,j}^k = e^{\frac{-||x_i - x_j||^2}{Te}}.$$

Here, we define the L as the equation $L = D^k - L^k$. D^k is a diagonal matrix whose entries are column (or row, since Sis symmetric) sums of L^k. For convenience, the neighbor number used in our graph is denoted as r.

2.3. Variational Expectation Maximization Inference

Our scheme for estimating the data cluster depends mainly on our capability to infer the posterior distribution. We solve this using variational expectation maximization inference.

Unlike the conventional expectation maximization algorithm, the posterior probability in our model will be estimated via the variational inference, and then, we optimize the MoG parameter by maximizing the lower bound with the fixed variational parameter. Following the general variational inference framework, we firstly give the Evidence Lower BOund (ELBO) for the SimDPM with the truncated stick-breaking process (when applying this process, the maximum cluster number ∞ is truncated to K) [37].

$$\begin{aligned} \log(p(\boldsymbol{X}|\alpha,\boldsymbol{\lambda},\tilde{\boldsymbol{\Theta}}) - \lambda_R R \geq & E_q[\log p(\boldsymbol{\Theta}^*|\boldsymbol{\lambda})] + \sum_{n=1}^{N} E_q[\log p(z_n^{(1)}, z_n^{(2)}|\boldsymbol{V})] + \sum_{n=1}^{N} E_q[\log p(x_n|\boldsymbol{\Theta}^*, z_n^{(1)})] \\ & - E_q[\log q(\boldsymbol{v},\boldsymbol{\Theta}^*,\boldsymbol{z}^{(1)},\boldsymbol{z}^{(2)})] + \sum_{n=1}^{N} E_q[\log p(\boldsymbol{x}_n|\tilde{\boldsymbol{\Theta}}, z_n^{(2)})] \\ & + E_q[\log p(\boldsymbol{v}|\alpha)] - \boldsymbol{\lambda}_R R, \end{aligned} \tag{4}$$

where $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N$ is the observation sample, $p(\boldsymbol{\Theta}^*|\boldsymbol{\lambda})$ is a normal-Wishart distribution with hyper parameters $\boldsymbol{\lambda} = (\boldsymbol{u}_0, c_0, \boldsymbol{W}_0, v_0)$ and $p(\boldsymbol{x}_n|\tilde{\boldsymbol{\Theta}})$ is the constrained MoG distribution parameterized by $\tilde{\boldsymbol{\Theta}} = \{\tilde{\boldsymbol{\theta}}_k\}_{k=1}^K$ where $\tilde{\boldsymbol{\theta}}_k = \{\tilde{\boldsymbol{u}}_k, \tilde{\boldsymbol{\delta}}_k, \tilde{\boldsymbol{\pi}}_k\}$, $p(\boldsymbol{x}_n|\boldsymbol{\Theta}^*)$ is a single Gaussian distribution. $\boldsymbol{z}^{(1)} = \{z_n^{(1)}\}_{n=1}^N$ and $\boldsymbol{z}^{(2)} = \{z_n^{(2)}\}_{n=1}^N$ are the indicator variables sampled from the categorical distribution $p(z_n^{(1)}, z_n^{(2)}|\boldsymbol{v})$. Following the factorized family variational inference [37], which can make the posterior distribution computable, q can be expressed as:

$$q(\boldsymbol{v},\boldsymbol{\Theta}^*,\boldsymbol{z}^{(1)},\boldsymbol{z}^{(2)}) = \prod_{k=1}^{K-1} q_{\gamma_k}(v_k) \prod_{k=1}^{K} q_{\boldsymbol{\tau}_k}(\boldsymbol{\theta}_k^*) \prod_{n=1}^{N} q_{s\boldsymbol{\phi}_n}(z_n^{(1)}) \prod_{n=1}^{N} q_{(1-s)\boldsymbol{\phi}_n}(z_n^{(2)}), \tag{5}$$

where $q_{\gamma_k}(v_k)$ is the Beta distribution with $\gamma_k = \{\gamma_{k,1}, \gamma_{k,2}\}$ and $q_{\boldsymbol{\tau}_t}(\boldsymbol{\theta}_t^*)$ is a normal-Wishart distribution with the parameter $\boldsymbol{\tau}_k = \{\boldsymbol{u}_k, c_k, \boldsymbol{W}_k, v_k\}$. For $q_{s\boldsymbol{\phi}_n}(z_n^{(1)})$ and $q_{(1-s)\boldsymbol{\phi}_n}(z_n^{(2)})$, we denote it as two categorical distributions with parameter $\boldsymbol{\phi}_n = \{\phi_{n,k}\}_{k=1}^K$ ($\boldsymbol{\Phi} = \{\boldsymbol{\phi}_n\}_{n=1}^N \in \mathbb{R}^{N\times K}$). s is the tradeoff parameter.

For derivation convenience, we denote ELBO as $L(\boldsymbol{\gamma}, \boldsymbol{\tau}, \boldsymbol{\Phi}, \tilde{\boldsymbol{\Theta}})$. By using this inequality relaxation, we note that learning the model and estimating the model parameters are altered to maximize the following equation.

$$\begin{aligned} & \underset{\{\gamma_k,\boldsymbol{u}_k,c_k,\boldsymbol{W}_k,v_k,\tilde{\boldsymbol{u}}_k,\tilde{\boldsymbol{\delta}}_k,\tilde{\boldsymbol{\pi}}_k\}}{\arg\max} \quad L(\boldsymbol{\gamma},\boldsymbol{\tau},\boldsymbol{\Phi},\tilde{\boldsymbol{\Theta}}) - \lambda_R R \\ & s.t. \frac{1}{2}(\tilde{\boldsymbol{u}}_{k,m} - \tilde{\boldsymbol{u}}_{k,m-1})^T \tilde{\boldsymbol{\delta}}_{k,m}^{-1}(\tilde{\boldsymbol{u}}_{k,m} - \tilde{\boldsymbol{u}}_{k,m-1}) < \epsilon,\ m > 1, \\ & R = \sum_{k=1}^{\infty} p(k|\boldsymbol{X})^T L p(k|\boldsymbol{X}). \end{aligned}$$

We also notice that, since we have truncated the maximum cluster number to K, the penalty term R is altered to be $R = \sum_{k=1}^{K} p(k|\boldsymbol{X})^T L p(k|\boldsymbol{X})$.

Variational E-step: In the variational inference framework, the variational parameter can be estimated by maximizing the lower bound of likelihood function $\log p(\boldsymbol{X}|\alpha,\boldsymbol{\lambda})$ with the coordinate ascent algorithm.

For $\phi_{n,k}$ in $\{\phi_{n,k}\}_{k=1}^K$, note that this is a constrained maximization since $\sum_{k=1}^K \phi_{n,k} = 1$, and the probability $p(k|\boldsymbol{X})$ can be approximated by the variational parameter $\boldsymbol{\Phi}_{.,k}$. To solve this problem, we use an auxiliary variable $\boldsymbol{A}_k = \boldsymbol{\Phi}_{.,k}$ where $A_k \in \mathbb{R}^N$ and form the Lagrangian by isolating the terms in ELBO, which contain $\phi_{n,k}$ as:

$$\begin{aligned} & \underset{\phi_n}{\arg\max} L(\boldsymbol{\gamma},\boldsymbol{\tau},\boldsymbol{\Phi},\tilde{\boldsymbol{\Theta}}) + \boldsymbol{\lambda}_L(\sum_{k=1}^{K} \phi_{n,k} - 1) - \boldsymbol{\lambda}_R \sum_{k=1}^{K} \boldsymbol{A}_k^T L \boldsymbol{A}_k, \\ & \quad s.t.\ \boldsymbol{A}_k = \boldsymbol{\Phi}_{.,k}, \end{aligned} \tag{6}$$

where $\boldsymbol{\lambda}_L$ is a Lagrangian multiplier and $\boldsymbol{\lambda}_R$ is a penalty parameter.

Fix $\boldsymbol{A}_k$ to update $\boldsymbol{\Phi}_{.,k}$. The updating rule for $\phi_{n,k}$ can be achieved by taking the derivation.

$$\begin{aligned}\log \phi_{n,k} \propto & s E_q[\log p(\boldsymbol{x}_n|\boldsymbol{\theta}_k^*)] + (1-s)\log \mathrm{p}(\boldsymbol{x}_n|\tilde{\boldsymbol{\theta}}_k) \\ & + \sum_{j<k,n} (\Psi(\gamma_{j,2}) - \Psi(\gamma_{j,1}+\gamma_{j,2})) \\ & + \Psi(\gamma_{k,1}) - \Psi(\gamma_{k,1}+\gamma_{k,2}).\end{aligned} \tag{7}$$

We now fix $\boldsymbol{\Phi}$ to update $\boldsymbol{H}_k$.

$$\begin{aligned}& \arg\min_{\boldsymbol{A}_k} \boldsymbol{A}_k^T L \boldsymbol{A}_k + \boldsymbol{\lambda}_A ||\boldsymbol{A}_k - \boldsymbol{\Phi}_{.,k}^{old}||_2^2 \\ & \Longrightarrow \boldsymbol{A}_k = \boldsymbol{\Phi}_{.,k} = \boldsymbol{\lambda}_H(\lambda_A I + L)^{-1}\boldsymbol{\Phi}_{.,k}^{old},\end{aligned} \tag{8}$$

where $\boldsymbol{\lambda}_A$ is the penalty parameter. For the other variational parameter, we can attain the following closed-form solutions when taking the derivation of the previous proposed ELBO function and setting it to zero:

$$\begin{aligned}\gamma_{k,1} &= 1 + \sum_{i=1}^{N} \phi_{i,k}, \\ \gamma_{k,2} &= \alpha + \sum_{i=1}^{N} \sum\nolimits_{j>k} \phi_{i,j},\end{aligned} \tag{9}$$

$$\begin{aligned}c_k &= c_0 + N_k, \\ v_k &= v_0 + N_k,\end{aligned} \tag{10}$$

$$\begin{aligned}\boldsymbol{u}_k &= \frac{1}{c_k}(c_0\boldsymbol{u}_0 + N_k\,\overline{\boldsymbol{x}}_k), \\ \mathbf{W}_k^{-1} &= \mathbf{W}_0^{-1} + N_k\mathbf{S}_k + \frac{c_0 N_k}{c_0 + N_k}(\overline{\boldsymbol{x}}_k - \boldsymbol{u}_0)(\overline{\boldsymbol{x}}_k - \boldsymbol{u}_0)^T\end{aligned} \tag{11}$$

where N_k, $\mathbf{S}_k$ and $\bar{\boldsymbol{x}}_k$ can be estimated as follows:

$$\begin{aligned}N_k &= \sum_{n=1}^{N} s\phi_{n,k}, \\ \overline{\boldsymbol{x}}_k &= \frac{s}{N_k}\sum_{n=1}^{N} \phi_{n,k}\boldsymbol{x}_n, \\ \mathbf{S}_k &= \frac{s}{N_k}\sum_{n=1}^{N} \phi_{n,k}(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_k)(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_k)^T.\end{aligned} \tag{12}$$

For the prior parameters $\boldsymbol{u}_0, c_0, \boldsymbol{W}_0, v_0$, we use them in a non-informative manner to make them influence as little as possible the inference of the variational posterior distributions. For the other variational parameters, we initialize them in a random way.

Variational M-step: To optimize the lower bound parameter $\tilde{\boldsymbol{\theta}}$, we apply the EM framework again, in which we introduce an auxiliary posterior variable $q(k,m|\boldsymbol{x}_n)$ and Jensen's inequality [37].

$$\begin{aligned}& L(\gamma, \boldsymbol{\tau}, \boldsymbol{\Phi}, \tilde{\boldsymbol{\Theta}}) + H\sum_{m=2}^{M}\left\{(\tilde{\boldsymbol{u}}_{k,m} - \tilde{\boldsymbol{u}}_{k,m-1})^T\tilde{\boldsymbol{\delta}}_{k,m}^{-1}(\tilde{\boldsymbol{u}}_{k,m} - \tilde{\boldsymbol{u}}_{k,m-1})\right\} \\ & \geq C + \sum_{n=1}^{N}\sum_{k,m=1}^{K,M}\left\{(1-s)\phi_{n,k}q(k,m|\boldsymbol{x}_n)\log\frac{\tilde{\boldsymbol{\pi}}_{k,m}N(\boldsymbol{x}_n|\tilde{\boldsymbol{u}}_{k,m}, \tilde{\boldsymbol{\delta}}_{k,m})}{q(k,m|\boldsymbol{x}_n)}\right\} \\ & + H\sum_{m=2}^{M}\left\{(\tilde{\boldsymbol{u}}_{k,m} - \tilde{\boldsymbol{u}}_{k,m-1})^T\tilde{\boldsymbol{\delta}}_{k,m}^{-1}(\tilde{\boldsymbol{u}}_{k,m} - \tilde{\boldsymbol{u}}_{k,m-1})\right\},\end{aligned} \tag{13}$$

where C is a constant value with no respect to $\tilde{\boldsymbol{\Theta}}$ in $L(\boldsymbol{\gamma}, \boldsymbol{\tau}, \boldsymbol{\Phi}, \tilde{\boldsymbol{\Theta}})$. By using the inequality relaxation, the variational M-step can be reformulated as the optimization problem:

$$\begin{aligned} \max\Big\{ \sum_{n=1}^{N} \phi_{n,k} q(k,m|x_n) \log \frac{\tilde{\boldsymbol{\pi}}_{k,m} N(x_n|\tilde{\boldsymbol{u}}_{k,m}, \tilde{\boldsymbol{\delta}}_{k,m})}{q(k,m|x_n)} \\ + H(\tilde{\boldsymbol{u}}_{k,m} - \tilde{\boldsymbol{u}}_{k,m-1})^T \tilde{\boldsymbol{\delta}}_{k,m}^{-1} (\tilde{\boldsymbol{u}}_{k,m} - \tilde{\boldsymbol{u}}_{k,m-1}) \\ + \boldsymbol{\lambda}_L \big(\textstyle\sum_{m=1}^{M} q(k,m|\boldsymbol{x}_n) - 1\big)\Big\}, \end{aligned} \tag{14}$$

where $\boldsymbol{\lambda}_L$ and H are the Lagrangian multipliers. We therefore achieve the following closed-form solution by taking the derivative and setting the lower bound of (14) to zero:

$$\tilde{\boldsymbol{u}}_{k,1} = \frac{\sum_{n=1}^{N} \phi_{n,k} q(k,1|\boldsymbol{x}_n) \boldsymbol{x}_n}{\sum_{n=1}^{N} \phi_{n,k} q(k,1|\boldsymbol{x}_n)}; \tag{15}$$

when m is greater than one, we have:

$$\tilde{\boldsymbol{u}}_{k,m} = \frac{\sum_{n=1}^{N} \phi_{n,k} q(k,m|\boldsymbol{x}_n) \tilde{\boldsymbol{\delta}}_{k,m-1} \boldsymbol{x}_n - H \tilde{\boldsymbol{u}}_{k,m-1}}{\sum_{n=1}^{N} \phi_{n,k} q(k,m|\boldsymbol{x}_n) - H}. \tag{16}$$

Similar to the mean parameter $\tilde{\boldsymbol{u}}_{k,m}$, for $\tilde{\boldsymbol{\delta}}_{k,m}$, we have:

$$\tilde{\boldsymbol{\delta}}_{k,m} = \frac{T_{k,m}}{\sum_{n-1}^{N} \phi_{n,k} q(k,m|\boldsymbol{x}_n)}, m > 1, \tag{17}$$

where:

$$\begin{aligned} T_{k,m} = -H(\tilde{\boldsymbol{u}}_{k,m} - \tilde{\boldsymbol{u}}_{k,m-1})(\tilde{\boldsymbol{u}}_{k,m} - \tilde{\boldsymbol{u}}_{k,m-1})^T \\ + \sum_{n=1}^{N} \phi_{n,k} q(k,m|\boldsymbol{x}_n)(\boldsymbol{x}_n - \tilde{\boldsymbol{u}}_{k,m})(\boldsymbol{x}_n - \tilde{\boldsymbol{u}}_{k,m})^T, \end{aligned}$$

since the constraint does not exist in the components where $m < 2$, the updating rule for $m = 1$ is a little different.

$$\tilde{\boldsymbol{\delta}}_{k,m} = \frac{\sum_{n=1}^{N} \phi_{n,k} q(k,m|\boldsymbol{x}_n)(\boldsymbol{x}_n - \tilde{\boldsymbol{u}}_{k,m})(\boldsymbol{x}_n - \tilde{\boldsymbol{u}}_{k,m})^T}{\sum_{n=1}^{N} \phi_{n,k} q(k,m|\boldsymbol{x}_n)}. \tag{18}$$

For the computation of $\boldsymbol{\pi}_{k,m}$, we have:

$$\tilde{\boldsymbol{\pi}}_{k,m} = \frac{\sum_{n=1}^{N} \phi_{n,k} q(k,m|\boldsymbol{x}_n)}{\sum_{m}^{M} \sum_{n=1}^{N} \phi_{n,k} q(k,m|\boldsymbol{x}_n)}. \tag{19}$$

The computation of $q(k,m|\boldsymbol{x}_n)$ will be identical to the standard mixture of Gaussians model learning algorithm [37].

2.4. Agorithm

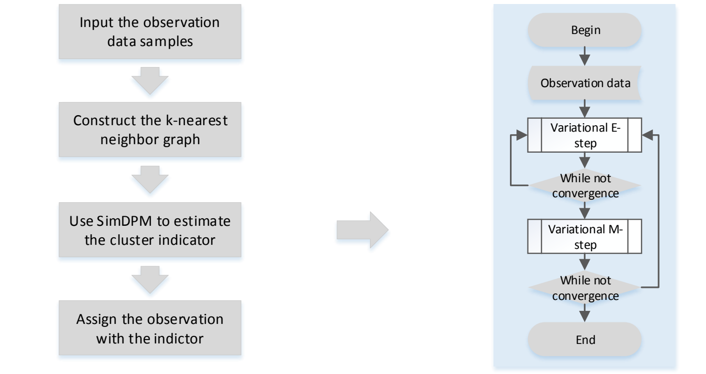
The full learning and inference algorithm is summarized in Algorithm 1. The flowchart of our proposed framework is demonstrated in Figure 3. Below, we analyze the computational complexity.

Algorithm 1 Semi-supervised DPM clustering algorithm.

```
Require: unlabeled dataset $\boldsymbol{X}_u$.
Ensure: variational parameters, $\{\boldsymbol{\gamma_k}, \boldsymbol{\tau_k}, \boldsymbol{\phi_k}\}_{k=1}^{K}$, and model parameter, $\{\tilde{\boldsymbol{\theta}_k}\}_{k=1}^{K}$.
Construct the $k$-nearest neighbor graph $L^k$ and $L$. Initialize the variational parameter randomly.
while not convergent do
    Expectation step:
    while not convergent do
        for all $n, k$ do
            Update the variational parameters $\{ \phi_{n,k}^{old}\}$ using (7).
            Update the variational parameters $\{\nu_k,\ \phi_k,\ c_k,\ \boldsymbol{u}_k,\ \mathbf{W}_k^{-1},\ v_k\}$ using (9), (10) and (11).
        end for
        Update the variational parameters $\{ \phi_{n,k}\}$ using (8).
    end while
    Maximization step:
    for $k = 1; k \leq K; k = k + 1$ do
        Update $\tilde{\boldsymbol{u}}_{k,1}$ and $\tilde{\delta}_{k,1}$ with (15) and (18).
        for $m = 2; m \leq M; m = m + 1$ do
            Update $\tilde{\boldsymbol{u}}_{k,m}$, $\tilde{\delta}_{k,m}$ and $\tilde{\boldsymbol{\pi}}_{k,m}$ using (16), (17) and (19).
        end for
    end for
    $q(k, m|x_n) \leftarrow \frac{\tilde{\boldsymbol{\pi}}_{k,m} N(x_n|\tilde{\boldsymbol{u}}_{k,m}, \tilde{\delta}_{k,m})}{\sum_{m=1}^{M} \tilde{\boldsymbol{\pi}}_{k,m} N(x_n|\tilde{\boldsymbol{u}}_{k,m}, \tilde{\delta}_{k,m})}$
end while
```

Algorithm complexity: Suppose that we have N samples, each sample has D dimensions. The maximum cluster number in our experiment is K. Expectation step converges after running T_e times. The whole algorithm converges after T times. From the derivation, we know that the main computation lies on the Equations (7), (8) and (11), in which we need to calculate the inverse and determinant of the matrix. For Equations (7) and (11), we need $O(K \cdot D^3)$. For Equation (8), we need $O(N^3)$. Another major computation is the Equations (16) and (17), which takes the computational complexity of $O((M-1) \cdot K \cdot N \cdot D^2)$. According to the debates, we know that the whole algorithm computational complexity is $O(T \cdot (T_e \cdot (N^3 + K \cdot D^3) + (M-1) \cdot K \cdot N \cdot D^2))$.

For the space complexity, the main cost is the variational parameters which takes $O(K \cdot D^2 + N \cdot K)$. Another cost is the MoG parameters which needs $O(M \cdot K \cdot D^2)$. Then, the total space complexity is $O(K \cdot D^2 + N \cdot K + M \cdot K \cdot D^2)$.


Input the observation data samples
Construct the k-nearest neighbor graph
Use SimDPM to estimate the cluster indicator
Assign the observation with the indictor
Begin
Observation data
Variational E-step
While not convergence
Variational M-step
While not convergence
End

(a) Flowchart of the full framework

(b) Flowchart of the SimDPM

Figure 3. Illustration of the framework flowchart. (**a**) shows the flowchart of the framework; (**b**) demonstrates the flowchart of SimDPM.

3. Results

To demonstrate the usefulness of the proposed manifold model, we tested our method on both synthetic and real-world datasets and compared it with the following methods:

1. Original Dirichlet Process Mixture (DPM) model [38].
2. Affinity Propagation (AP) clustering [39].
3. A Dirichlet process-based linear manifold clustering method, DP-space [28].
4. Density-based Clustering algorithm by Fast Search and Find of Density Peaks (CFSFDP) [40].
5. Another category is the clustering method, which needs to specify the class number, K-means, LRR [17] and LatLRR [41].

Clustering accuracy in our experiment was measured through Normalized Mutual Information (NMI) [32]. Suppose $U = \{U_1, U_2, U_3, ..., U_{|U|}\}$ denotes the real cluster labels obtained from the ground truth and $V = \{V_1, V_2, V_3, ..., V_{|V|}\}$ obtained from a clustering algorithm. $|U|$ and $|V|$ denote the cluster number. Then, a mutual information metric between U and V can be defined as:

$$MI(U,V) = \sum_{i=1}^{|U|}\sum_{j=1}^{|V|} P(i,j) log(\frac{P(i,j)}{P(i)P(j)}) \tag{20}$$

where $P(i)$ and $P(j)$ are the probability that a sample picked at random falls into class U_i or V_j and $P(i,j)$ denotes the probability that a sample falls into both classes U_i and V_j. The Normalized Mutual Information (NMI) then can be defined as:

$$NMI(U,V) = \frac{MI(U,V)}{\sqrt{(H(U)H(V))}} \tag{21}$$

where $H(U)$ and $H(V)$ denote the entropy.

Experimental setup: In our experiment, we ran every algorithm 10 times and report the average accuracy. The parameters of the SimDPM algorithm were selected using the ground-truth labels of less than 40% according to the clustering accuracy. The default value for α and the maximum cluster number K were set at 20 and 30. The other variational parameters were initialized randomly except $\boldsymbol{u}_k$ and $\mathbf{W}_k$, for which we used the mean and covariance of the observation data to initialize. All our algorithms were implemented in MATLAB R2016a on a DELL Precision Workstation with 8.00 G RAM and a Xeon(R) E3 CPU.

For the original DPM, we used the $\alpha = 20$ and set the other variational parameters randomly. When operating the DP-space, we used $\boldsymbol{\lambda}$ and s from the values, as this was suggested in the original codes, and we selected this by using 30% ground-truth labels. The parameters used in LatLRR and LRR were that $\alpha = 1$, $\beta = 1.4$ and $\boldsymbol{\lambda} = 4$. For CFSFDP, we chose the determination points that were significantly different from the other points in the decision graph. In the setting of AP, we used the preference value as a scalar one. Both CFSFDP and AP used the K-nearest neighbor graph as the similarity matrix.

3.1. Synthetic Dataset

In this section, we evaluate our SimDPM model on a synthetic dataset. We show the results in Figure 4. Clearly, there are two patterns.

Visual comparison on synthetic hybrid data shows that SimDPM performed better than the traditional DPM model. In our result, Figure 4b shows the result using the original variational DPM. As can be seen, the original DPM tended to partition the synthetic dataset in a hard manner. Our manifold method yielded an ideal clustering result. The reason is that our model handled the dataset with a Gaussian expanding distribution and reserved the local geometrical structure of the data space by applying a k-nearest neighbor graph.

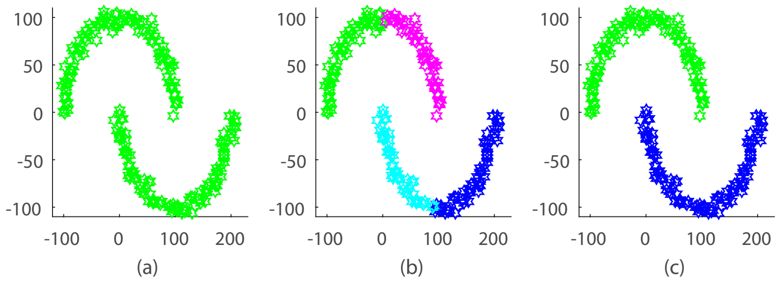


Figure 4. Illustration of SimDPM and the original DPM clustering result on a synthetic dataset. (**a**) demonstrates the original dataset with no label; (**b**,**c**) are the original DPM clustering result and SimDPM clustering result. Different color means different cluster.

3.2. Real Dataset

(a) Motion segmentation: Motion segmentation usually refers to the task of separating the movements of multiple rigid-body objects from video sequences. Linear manifold clustering methods are popular in this task [42]. In our experiment, we used the Hopkin155 dataset [43] and cast it into a general multi-manifold clustering task. We show some samples in Figure 5. According to the dataset itself, we divided the universal set into checkerboard and others [43], in which each contained 26 and nine subsets. For the checkerboard dataset, we separated it into the Linear manifold dataset (L) and Non-Linear manifold dataset (Non-L) according to the 3D projection of PCA. When applying our algorithm, we projected point trajectories into 10D features. The clustering result and the estimated cluster number are presented in Tables 2 and 3. As can be seen, our proposed method performed the best on the Non-L dataset. On the others and L dataset, DP-space was the first best, and our method was the second best. For the estimated cluster number, we can observe that our model could produce the suitable cluster size compared with the ground truth.

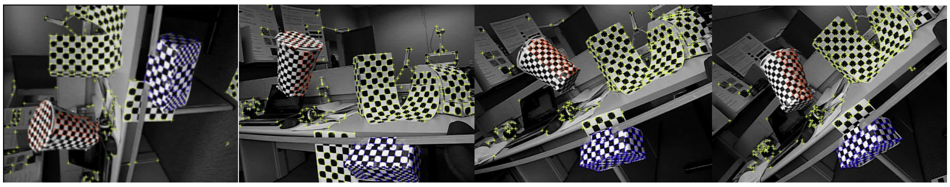


Figure 5. Illustration of the Hopkin155 dataset.

Table 2. Clustering accuracy on the Hopkin155 dataset with 3 motions. AP, Affinity Propagation; CFSFDP, Density-based Clustering algorithm by Fast Search and Find of Density Peaks; LRR, Low-Rank Representation. Bolded numbers denote the highest clustering accuracy.

Method	**Checkerboard**			**Others**
	L	**Non-L**	**Average**	
SimDPM	0.80	**0.73**	**0.79**	0.83
DPM [38]	0.42	0.37	0.41	0.45
DP-space [28]	**0.84**	0.48	0.78	**0.94**
AP [39]	0.29	0.32	0.29	0.31
CFSFDP [40]	0.40	0.19	0.36	0.47
K-means	0.48	0.48	0.49	0.47
LRR [17]	0.51	0.33	0.48	0.33
LatLRR [41]	0.52	0.31	0.47	0.34

Table 3. The estimated cluster number on the Hopkin155 dataset with 3 motions. L, Linear.

Method	**Checkerboard**			**Others**
	L	**Non-L**	**Average**	
Ground truth	3.00	3.00	3.00	3.00
The estimated cluster number	3.33	3.60	3.10	3.09

(b) Coil20 image dataset: The coil20 [44] image database is a popular manifold database containing 20 objects from the Columbia university image library. Some image samples are demonstrated in Figure 6. Each image is taken from five degrees apart as the object is rotated on a turntable. Thus, each object in coil20 has 72 images. The size of the object is 128 × 128, with 256 grey levels per pixel. In our experiment, each image was firstly represented by a 128 × 128 dimensional vector, and then, we projected it into a 10D feature using the PCA method. To test the general clustering performance, we used five coil20 subsets. For the overall testing, we also gave the universal dataset (Dataset 20). The clustering result and the estimated cluster number are demonstrated in Tables 4 and 5. From the result, we know that our method consistently outperformed the DP-based algorithms such as DP-space and DPM. When comparing with the other methods, our method was the first or the second best, especially compared with the approaches that do not need to specify the cluster number.

Figure 6. Illustration of the coil20 dataset.

Table 4. Clustering accuracy on the coil20 dataset. Bolded numbers denote the highest clustering accuracy.

Method	**Subdataset**					**20**
	2	**4**	**6**	**8**	**10**	
SimDPM	**0.30**	0.55	0.56	**0.60**	**0.69**	0.72
DPM [38]	0.29	0.42	0.50	0.53	0.57	0.69
DP-space citewang2015dp	0.01	0.34	0.10	0.19	0.10	0.26
AP [39]	0.12	0.22	0.18	0.11	0.25	0.36
CFSFDP [40]	0	0.57	0.57	0.53	0.46	0.42
K-means	0	0.52	0.46	0.57	0.59	**0.73**
LRR [17]	0.11	**0.62**	0.56	0.47	0.52	0.70
LatLRR [41]	0	0.57	**0.57**	0.48	0.50	0.58

Table 5. The estimated cluster number on the coil20 dataset.

Method	**Subdataset**					**20**
	2	**4**	**6**	**8**	**10**	
Ground truth	2.0	4.0	6.0	8.0	10.0	20.0
The estimated cluster number	3.1	5.0	6.3	6.7	11.3	21.7

(c) Swedish leaf image dataset: The Swedish dataset introduced in [45] consists of 1125 leaves of 15 species with 75 images per species. In this dataset, we firstly extracted the outer contour and then achieved the contour features by applying the Fourier transform [46]. Every leaf in our experiments was represented as a 10-dimensional feature. Some samples are shown in Figure 7.

Similar to the coil20 dataset, we demonstrated the efficiency on five subsets and the universal dataset. We ran every algorithm in our experiment 10 times, and took the accuracy by averaging the 10 results. The experimental results are demonstrated in Tables 6 and 7, which present some observations: (1) compared with the original DPM, the improvement of the clustering accuracy (average 0.07) was lower than the improvement in the coil20 dataset (average 0.05); (2) the cluster number was consistently increasing as the ground truth cluster number was increasing.

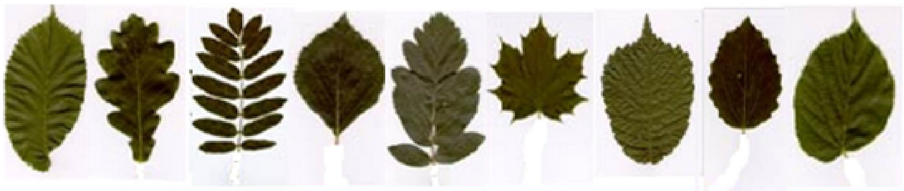


Figure 7. Leaf samples from the leaf dataset.

Table 6. Clustering accuracy on the leaf dataset. Bolded numbers denote the highest clustering accuracy.

Method	Subdataset					15
	2	4	6	8	10	
SimDPM	0.29	**0.62**	0.46	**0.50**	**0.54**	**0.38**
DPM [38]	0.26	0.43	0.45	0.49	0.51	0.34
DP-space [28]	0.49	0.33	0	0	0	0.03
AP [39]	0	0.03	0.11	0	0.02	0
CFSFDP [40]	0.17	0.56	0.24	0.28	0.42	0
K-means	0.45	0.39	**0.61**	0.50	0.44	0.22
LRR [17]	**0.76**	0.45	0.48	0.33	0.40	0.22
LatLRR [41]	0.65	0.44	0.32	0.20	0.41	0.23

Table 7. The estimated cluster number on the leaf dataset.

Method	Subdataset					15
	2	4	6	8	10	
Ground truth	2.0	4.0	6.0	8.0	10.0	15.0
The estimated class number	4.7	6.3	10.2	14.5	16.2	20.2

From the experimental results in Tables 2–7, we can draw some points as follows.

- The proposed method obtained the highest clustering accuracy especially on the Non-L and coil20 dataset compared with the non-prespecified cluster number methods, which validates the effectiveness of our non-linear assumption.
- DP-space performed better than our method on the L and others dataset, the reason being that DP-space has a prior structure assumption, which introduces additional manifold geometric information.
- LRR, LatLRR and K-means outperformed our algorithm on some coil20 and leaf subdatasets, the reason being that our method needed to estimate the cluster number along with clustering. This made our algorithm hard to optimize.
- Compared to the coil20 and leaf dataset, our method achieved an all-around performance boosting on the motion segmentation dataset; this is because the simple clustering task (the linear manifold has only three classes) was easy for our algorithm to optimize and model.

- Compared to the leaf dataset, our method achieved a better clustering performance boosting on the coil20 dataset. The reason is that coil20 is a well-defined manifold dataset, in which the structure among samples is easy to capture by the graph Laplacian.
- Our manifold model consistently produced the suitable cluster number with the increasing of the data cluster size, which indicates that our model could provide a flexible model size when fitting different datasets.

3.3. The Effect of the Algorithm Parameters

In this section, we firstly investigate the effects of the parameters λ_A and s on the Non-L dataset. More specifically, in the experiment, when one parameter is being tuned, the value of the other parameter is fixed. The parameters λ_A and s were sampled from $\{1000, 100, 80, 60, 40, 20\}$ and $\{1, 0.9, 0.8, 0.7, 0.6, 0.5\}$. We show the clustering accuracy in Figure 8.

As we can see in Figure 8, experimental results indicate that the proposed model was sensitive to λ_A. Empirically, the best clustering accuracy was achieved when $\lambda_A = 100$. We also observed that our method achieved the best clustering accuracy when $s = 0.7, 0.8$. This reveals that the MoG had improved the clustering accuracy. Besides, we measured the clustering accuracy with the different M and the neighbor number r using the k-nearest neighbor graph. The clustering accuracy is demonstrated in Figures 10 and 9. As can be seen, the clustering accuracy achieved the best performance on the leaf subdataset and the coil20 dataset when the neighbor number $r = \{5, 6, 7, 8, 9, 10\}$. Clustering accuracy increased along with the increasing of the M in the subdataset of the coil20 dataset and leaf dataset. Unlike the subdataset, parameter M and the neighbor number r had little effect on the full dataset of coil20 and leaf. The reason is that our model is a non-convex model, and the complicated dataset led to a much more complicated optimization.

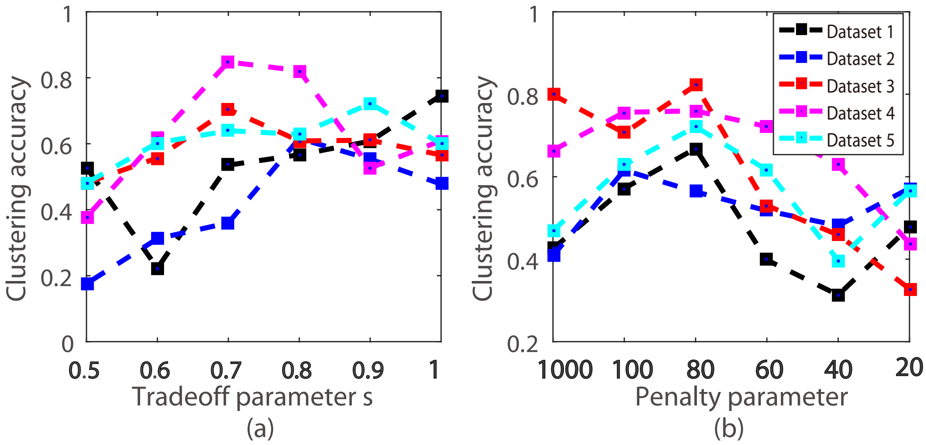


Figure 8. Illustration of the clustering accuracy with different s and λ_A on five hopkin155 datasets. (**a**) is the clustering accuracy with different s. (**b**) is the clustering accuracy with different λ_A.

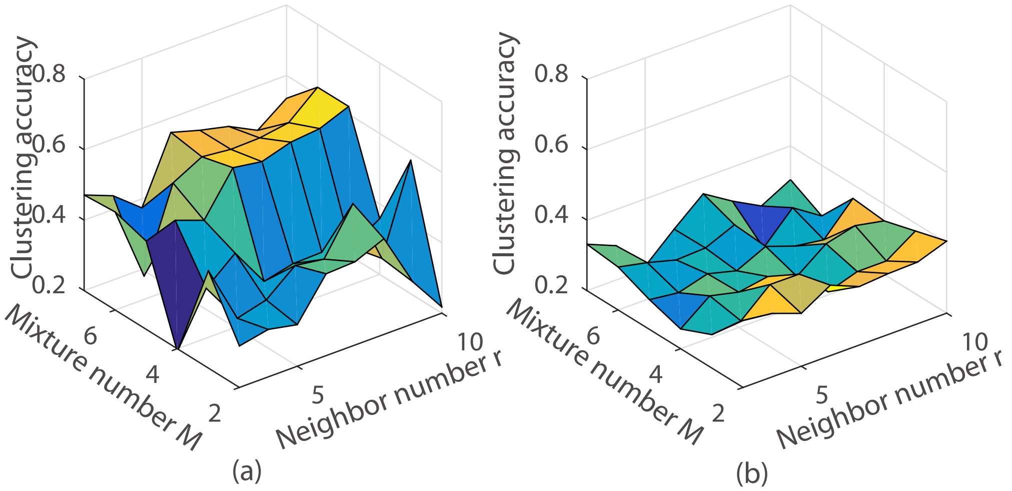


Figure 9. Illustration of the clustering accuracy with different M and the neighbor number r using the k-nearest neighbor graph on the leaf dataset. (**a**) Subdataset of leaf with 6 classes; (**b**) The leaf full dataset.

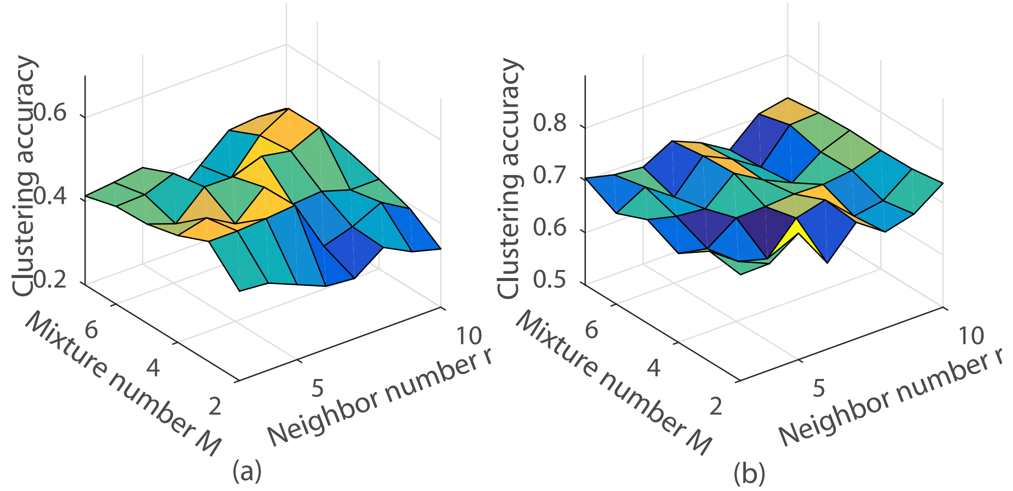


Figure 10. Illustration of the clustering accuracy with different M and the neighbor number r using the k-nearest neighbor graph on the coil20 dataset. (**a**) Subdataset of coil20 with 6 classes; (**b**) The coil20 full dataset.

4. Discussion

Compared to the previous linear manifold and geodesic mixture models, our theoretical analysis has shown that our method is a prespecified manifold and cluster number-free model. This is because we use a DP prior to generate the cluster indicator with the suitable cluster number and use the MoG and K-nearest neighbor graph to capture the submanifold rather than using a predefined manifold. Additionally, compared with different multi-manifold clustering methods with prespecified manifolds and cluster numbers like DP-space, LRR and LatLRR, our method has shown superior performance on the general manifold clustering task (coil20 and leaf dataset). This indicates our method can fill the research gap we have mentioned in the Introduction.

Although our method can handle the problems we have mentioned (estimating the cluster number and handling the general manifold clustering task), limitations still exist. That is, our method is not a full Bayesian framework. Thereby, some parameters should be tuned manually. This may be unacceptable in some real applications. Moreover, we note that MoG and the Gaussian distribution are sensitive to the dimension of the data. In future work, we will explore a full generative model, in which the parameters can be generated by using some Bayesian priors. Since our approach is sensitive to the dimension, we will also explore certain methods to integrate the dimension reduction method and the manifold clustering.

5. Conclusions

In this paper, we have proposed a nonparametric generative model to handle the manifold dataset with no prespecified cluster number and manifold distribution. In the course of the theoretical and experimental analysis, we have demonstrated that MoG can extend the application scope of the original DPM and can significantly improve the clustering accuracy compared to the previous proposed method. However, to be frank, the proposed method can only partially handle the problem we state in the Introduction due to the facts that: (1) MoG, the mean constraint and K-nearest neighbor graph are hard to optimize when we incorporate them into the DP framework; this can be observed when we use it in the coil20 and full leaf dataset; (2) the DP prior has a limitation when generating the suitable cluster number.

Author Contributions: X.Y. and J.Z.: investigation and original draft preparation. Y.C.: review and editing.

Funding: This research was funded by National Natural Science Foundation of China grant number 61571247, 61603202, the National Natural Science Foundation of Zhejiang Province grant number LZ16F030001, the International Cooperation Projects of Zhejiang Province grant number NO.2013C24027, and the K. C. Wong Magna Fund in Ningbo University. The APC was funded by the National Natural Science Foundation of China grant number 61571247.

Acknowledgments: This work was supported by the National Natural Science Foundation of China under Grant Nos. 61571247 and 61603202, the National Natural Science Foundation of Zhejiang Province under Grant No. LZ16F030001, the International Cooperation Projects of Zhejiang Province under Grant No. 2013C24027 and the K. C. Wong Magna Fund at Ningbo University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Akogul, S.; Erisoglu, M. An Approach for Determining the Number of Clusters in a Model-Based Cluster Analysis. *Entropy* **2017**, *19*, 452. [CrossRef]
2. Zang, W.; Zhang, W.; Zhang, W.; Liu, X. A Kernel-Based Intuitionistic Fuzzy C-Means Clustering Using a DNA Genetic Algorithm for Magnetic Resonance Image Segmentation. *Entropy* **2017**, *19*, 578. [CrossRef]
3. Wang, Y.; Liu, Y.; Blasch, E.; Ling, H. Simultaneous Trajectory Association and Clustering for Motion Segmentation. *IEEE Signal Process. Lett.* **2017**, *25*, 145–149. [CrossRef]
4. Bansal, S.; Aggarwal, D. Color Image Segmentation Using CIELab Color Space Using Ant Colony Optimization. *Int. J. Comput. Appl.* **2011**, *29*, 28–34. [CrossRef]
5. Chen, P.Y.; Hero, A.O. Phase Transitions in Spectral Community Detection. *IEEE Trans. Signal Process.* **2015**, *63*, 4339–4347. [CrossRef]
6. Sun, J.; Zhou, A.; Keates, S.; Liao, S. Simultaneous Bayesian Clustering and Feature Selection through Student's *t* Mixtures Model. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 1187–1199. [CrossRef] [PubMed]
7. Wei, H.; Chen, L.; Guo, L. KL Divergence-Based Fuzzy Cluster Ensemble for Image Segmentation. *Entropy* **2018**, *20*, 273.
8. Mo, Y.; Cao, Z.; Wang, B. Occurrence-Based Fingerprint Clustering for Fast Pattern-Matching Location Determination. *IEEE Commun. Lett.* **2012**, *16*, 2012–2015. [CrossRef]
9. Cai, D.; Mei, Q.; Han, J.; Zhai, C. Modeling hidden topics on document manifold. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, CA, USA, 26–30 October 2008; pp. 911–920.
10. Li, B.; Lu, H.; Zhang, Y.; Lin, Z.; Wu, W. Subspace Clustering under Complex Noise. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, 1. [CrossRef]
11. Rahkar Farshi, T.; Demirci, R.; Feiziderakhshi, M.R. Image Clustering with Optimization Algorithms and Color Space. *Entropy* **2018**, *20*, 296. [CrossRef]

12. Men, B.; Long, R.; Li, Y.; Liu, H.; Tian, W.; Wu, Z. Combined Forecasting of Rainfall Based on Fuzzy Clustering and Cross Entropy. *Entropy* **2017**, *19*, 694. [CrossRef]
13. Wang, Y.; Jiang, Y.; Wu, Y.; Zhou, Z.H. Spectral clustering on multiple manifolds. *IEEE Trans. Neural Netw.* **2011**, *22*, 1149–1161. [CrossRef] [PubMed]
14. Gholami, B.; Pavlovic, V. Probabilistic Temporal Subspace Clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3066–3075.
15. Vidal, R.; Ma, Y.; Sastry, S. Generalized Principal Component Analysis (GPCA). *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1945–1959. [CrossRef] [PubMed]
16. Elhamifar, E.; Vidal, R. Sparse Subspace Clustering: Algorithm, Theory, and Applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2765–2781. [CrossRef] [PubMed]
17. Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; Ma, Y. Robust Recovery of Subspace Structures by Low-Rank Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 171–184. [CrossRef] [PubMed]
18. Lu, C.; Min, H.; Zhao, Z.; Zhu, L.; Huang, D.; Yan, S. Robust and Efficient Subspace Segmentation via Least Squares Regression. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 347–360.
19. Souvenir, R.; Pless, R. Manifold Clustering. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV 2005), Beijing, China, 17–20 October 2005; pp. 648–653.
20. Allab, K.; Labiod, L.; Nadif, M. Multi-Manifold Matrix Decomposition for Data Co-Clustering. *Pattern Recognit.* **2017**, *64*, 386–398. [CrossRef]
21. Peng, X.; Xiao, S.; Feng, J.; Yau, W.Y.; Yi, Z. Deep Subspace Clustering with Sparsity Prior. In Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016), New York, NY, USA, 9–15 July 2016; pp. 1925–1931.
22. Ji, P.; Zhang, T.; Li, H.; Salzmann, M.; Reid, I. Deep Subspace Clustering Networks. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 24–33.
23. He, X.; Cai, D.; Shao, Y.; Bao, H.; Han, J. Laplacian Regularized Gaussian Mixture Model for Data Clustering. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 1406–1418. [CrossRef]
24. Neal, R.M. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *J. Comput. Graph. Stat.* **2000**, *9*, 249–265.
25. Wei, X.; Yang, Z. The Infinite Student's T-Factor Mixture Analyzer for Robust Clustering and Classification. *Pattern Recognit.* **2012**, *45*, 4346–4357. [CrossRef]
26. Nguyen, T.V.; Phung, D.; Nguyen, X.; Venkatesh, S.; Bui, H. Bayesian Nonparametric Multilevel Clustering with Group-Level Contexts. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 288–296.
27. Palla, K.; Ghahramani, Z.; Knowles, D.A. A Nonparametric Variable Clustering Model. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3-6 December 2012; pp. 2987–2995.
28. Wang, Y.; Zhu, J. DP-Space: Bayesian Nonparametric Subspace Clustering with Small-Variance Asymptotics. In Proceedings of the 32nd International Conference on Machine Learning (ICML 2015), Lille, France, 6–11 July 2015; pp. 862–870.
29. Straub, J.; Campbell, T.; How, J.P.; Fisher III, J.W. Efficient Global Point Cloud Alignment using Bayesian Nonparametric Mixtures. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 2941–2950.
30. Straub, J.; Freifeld, O.; Rosman, G.; Leonard, J.J.; Fisher, J.W., III. The Manhattan Frame Model—Manhattan World Inference in the Space of Surface Normals. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 235–249. [CrossRef] [PubMed]
31. Simo-Serra, E.; Torras, C.; Moreno-Noguer, F. 3D human pose tracking priors using geodesic mixture models. *Int. J. Comput. Vis.* **2017**, *122*, 388–408. [CrossRef]
32. Sommer, S.; Lauze, F.; Hauberg, S.; Nielsen, M. Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations. In Proceedings of the 11th European Conference on Computer Vision (ECCV 2010), Crete, Greece, 5–11 September 2010; pp. 43–56.
33. Huckemann, S.; Hotz, T.; Munk, A. Intrinsic shape analysis: Geodesic PCA for Riemannian manifolds modulo isometric lie group actions. *Stat. Sin.* **2010**, *20*, 1–58.

34. Cao, X.; Chen, Y.; Zhao, Q.; Meng, D.; Wang, Y.; Wang, D.; Xu, Z. Low-Rank Matrix Factorization under General Mixture Noise Distributions. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 1493–1501.
35. Zhao, Q.; Meng, D.; Xu, Z.; Zuo, W.; Zhang, L. Robust Principal Component Analysis with Complex Noise. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 55–63.
36. Liu, J.; Cai, D.; He, X. Gaussian Mixture Model with Local Consistency. In Proceedings of the 24th AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 11–15 July 2010; pp. 512–517.
37. Bishop, C.M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*; Springer: New York, NY, USA, 2006; p. 049901.
38. Blei, D.M.; Jordan, M.I. Variational inference for Dirichlet process mixtures. *Bayesian Anal.* **2006**, *1*, 121–143. [CrossRef]
39. Frey, B.J.; Dueck, D. Clustering by passing messages between data points. *Science* **2007**, *315*, 972–976. [CrossRef] [PubMed]
40. Rodriguez, A.; Laio, A. Machine learning. Clustering by fast search and find of density peaks. *Science* **2014**, *344*, 1492. [CrossRef] [PubMed]
41. Liu, G.; Yan, S. Latent Low-Rank Representation for subspace segmentation and feature extraction. In Proceedings of the 13th International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 1615–1622.
42. Zhang, Y.; Luo, B.; Zhang, L. Permutation Preference based Alternate Sampling and Clustering for Motion Segmentation. *IEEE Signal Process. Lett.* **2017**, *25*, 432–436. [CrossRef]
43. Tron, R.; Vidal, R. A benchmark for the comparison of 3D motion segmentation algorithms. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
44. Nene, S.A.; Nayar, S.K.; Murase, H. *Columbia Object Image Library (Coil-20)*; Technical Report; Columbia University: New York, NY, USA, 1996.
45. Söderkvist, O. Computer Vision Classification of Leaves from Swedish Trees. Master's Thesis, Linköping University, Linköping, Sweden, September 2001. Available online: http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A303038&dswid=-9927 (accessed on 26 October 2018).
46. Zahn, C.T.; Roskies, R.Z. Fourier descriptors for plane closed curves. *IEEE Trans. Comput.* **1972**, *100*, 269–281. [CrossRef]