MJFAS
Malaysian Journal of Fundamental and Applied Sciences

Wu *et al.* | Malaysian Journal of Fundamental and Applied Sciences, Vol. 19 (2023) 1099-1118

RESEARCH ARTICLE

Comparative Analysis of Improved Dirichlet Process Mixture Model

Lili Wu[a,b], Pei Shan Fam[a], Majid Khan Majahar Ali[a]*, Ying Tian[a], Mohd. Tahir Ismail[a], Siti Zulaikha Mohd Jamaludin[a]

[a]School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia; [b]Department of Computer Science, Xinzhou Teachers University, 034000 Xinzhou, Shanxi, China

Abstract Due to the development of information technology, large amounts of data are generated every day in various industries such as engineering, healthcare, finance, anomaly detection, image recognition, and artificial intelligence. This massive data poses the challenge of analyzing accurately and appropriate classifications. The traditional clustering methods require specifying the number of clusters and are mostly based on distance, which cannot effectively consider the correlations between different indicators of high-dimensional and multi-source data. Moreover, the number of clusters cannot automatically adjust when new data is generated. In order to improve the clustering analysis of high-dimensional and multi-source data in a big data environment, this study utilizes non-parametric mixture models based on distribution clustering, which does not require specifying the number of clusters and can auto update with the data. By combining Principal Component Analysis (PCA), t-Distributed Stochastic Neighbour Embedding (t-SNE), and the non-parametric Bayesian method called Dirichlet Process Mixture Model (DPMM), the Bayesian non-parametric PCA model (PCA-DPMM) and Bayesian non-parametric t-SNE model (TSNE-DPMM) are proposed. The Chinese restaurant process of DPMM is used for sampling by introducing a finite normal mixture distribution. The clustering results on the iris dataset are compared and analyzed. The accuracy of DPMM and TSNE-DPMM reaches 0.97, while PCA-DPMM achieves a maximum accuracy of only 0.94. When different numbers of iterations are set, TSNE-DPMM maintains an accuracy ranging from 0.92 to 0.97, DPMM ranges from 0.66 to 0.97, and PCA-DPMM ranges from 0.73 to 0.94. Therefore, the proposed TSNE-DPMM ensures accuracy and exhibits better model stability in clustering results. Future research can explore the improvement of the model by incorporating deep learning algorithms, among others, to further enhance its performance. Additionally, applying the TSNE-DPMM model to data analysis in other fields is also a future research direction. Through these efforts, we can better tackle the challenges of analyzing high-dimensional and multi-source data in a big data environment and extract valuable information from it.

Keywords: Bayesian non-parametric model; PCA; t-SNE; DPMM.

***For correspondence:**
majidkhanmajaharali@usm.my

Received: 18 June 2023
Accepted: 7 Nov. 2023

Introduction

In the recent years, clustering-based methods have been widely used in various problems such as anomaly detection and image recognition, which has led to an active research area in clustering algorithms for high-dimensional and multi-source data. In the field of clustering analysis for multi-source data, various clustering methods have been proposed, including partitioning-based methods such as K-means [1] and K-centroids clustering [2], hierarchical-based methods, density-based methods, grid-based methods, and model-based methods. Among the various proposed methods, the K-means method is often sensitive to outliers, while K-centroids clustering improves upon this issue. The Partitioning Around Medoids (PAM) algorithm [3] is a commonly used K-centroids-based clustering algorithm, but it is limited to low computational efficiency for large-scale data. Both of these partitioning-based methods can only handle spherical-shaped data and are not suitable for clustering data with

complex distribution shapes. Additionally, these methods require a priori knowledge of the number of clusters [4]. In response to these issues, model-based clustering methods have shown certain advantages, as they can cluster data from different distribution populations and determine the optimal number of clusters based on model selection criteria [5]. As a result, they have been widely applied in many fields. However, in model-based clustering methods, the number of indicators is determined by the model selection criteria, making it impossible to adaptively determine and update the number of clusters based on the data. When performing clustering analysis on multiple indicator data, the increase in data dimensionality and the inconsistency of information among indicators increase the complexity of the clustering problem, which affect the accuracy and timeliness of the clustering results.

Existing multi-indicator data clustering models have the following issues: (1) The assumption that each indicator data is independent and not considering the differences in information among indicators. (2) Slow algorithm convergence speed. (3) Requires a specified number of clusters. (4) High dimensionality leading to high clustering complexity.

To address the aforementioned issues, models based on the Dirichlet process [6] have shown certain advantages. The Dirichlet process, as a random process, is used as the prior distribution for Bayesian nonparametric statistics. In order to sample from the Dirichlet process, three different constructions have been proposed: the Polya urn scheme [7], the Chinese restaurant process (CRP) [8], and stick-breaking [9], making it possible to apply this process in various applications. Models based on the Dirichlet process can adaptively determine the number of clusters based on the data, thereby solving the problem of determining the number of clusters. However, the Dirichlet process can only cluster data with identical values, and when the data values are different, regardless of the similarity between the data clustering cannot be perform. In order to address this issue, Yao *et al*. [10] proposed the Dirichlet process mixture model (DPMM) based on the Dirichlet process, which can perform clustering analysis on data from different distribution populations. These advantages have led to the application of the DPMM model approach as a prior distribution in many fields of study. Considering the differences of information among indicators, the improved sticky hierarchical Dirichlet process method performs clustering analysis on multi-source data [11], where the stickiness parameter reflects the correlation between indicators and the overall clustering, allowing different states to have different distribution types. This means that different indicators can follow different distributions, enabling its direct application in the clustering of multi-source data and addressing the related issues in multi-source data clustering. Lai *et al*. [12] used the stick-breaking construction of the Dirichlet process as the prior distribution of the mixture weights in Gaussian mixture models, establishing the Dirichlet process mixture model and using variational methods to estimate the model parameters. The results showed that this method achieved better performance than traditional kernel principal component analysis. Peng *et al*. [13] combined the Dirichlet process mixture model with the K-means clustering algorithm and verified the outcome on public datasets. The experiments demonstrated that the improved algorithm can address the problem of indeterminate K values, and the stability, accuracy, and quality of the clustering results were significantly improved. Applying the improved model to gut microbiota OTUs data has provided a new approach to addressing issues related to type 2 diabetes clinically. The Dirichlet process mixture model can adaptively determine the optimal number of clusters based on the data. However, as the data size increases during clustering analysis, the MCMC-Gibbs sampling algorithm [14] updates only one data point at a time, leading to slow convergence. To address this issue, the Split-Merge Dirichlet Process Mixture Model (SMDPMM) [15] introduces split and merge operations during the sampling process to accelerate convergence and improve topic mining results. Most clustering methods are designed for low-dimensional data, this limitation poses a serious challenge whenever the dimensionality of the data increases. Currently, researchers use adopt the principal component analysis (PCA) approach and clustering analysis in order to simplify evaluation indicators and achieve dimensionality reduction clustering analysis [16]. However, the PCA selects principal components first, then linearly transforms the data, and finally performs clustering analysis, which does not fully capture the original data information and does not consider the correlation between clusters. The Flow Hierarchical Dirichlet Process (FHDP) [17] enhance the utilization of interrelated information between topics by incorporating flow operations into the Hierarchical Dirichlet Process (HDP) model, truncating needless information and making the hierarchical relationships of topics more explicit.

The main focal point of this work is on the inability of traditional clustering model to measure relationships between classes and directly determine the number of classes in a large complex data. In this study a Bayesian nonparametric PCA and t-SNE models is proposed; where the PCA, t-SNE, and nonparametric Bayesian methods are combined. When these models are applied for dimensionality reduction of high dimensional data clustering results are obtained which improves both the convergence speed of clustering and the determination of the number of classes.

Materials and Methods

The Dirichlet process is a stochastic process that describes the measure distribution, which is usually used in the Bayesian nonparametric mixed model to generate a priori on the mixed component when the component parameters of the mixed model are unknown. This section provides a brief introduction to the Dirichlet process, defined as follows [9]:

Suppose G_0 is a random probability distribution on the measurement space Θ, α_0 is a positive real number. For any finite partition $A_1,\ldots A_r$ of the measurement space Θ, if the random probability distribution G on the measurement space Θ satisfy the following conditions:

$$\left(G(A_1),\ldots,G(A_r)\sim Dir(\alpha_0 G_0(A_1)),\ldots,Dir(\alpha_0 G_0(A_r))\right) \quad (1)$$

Then G obeys the Dirichlet process composed of basis distribution G_0 and concentration parameters α_0, denoted as

$$G\sim DP(\alpha_0,G_0) \quad (2)$$

where, $Dir(\square)$ represents the Dirichlet distribution, α_0 indicating the degree of similarity with G. The bigger α_0, the more similar the two are. Conversely, if Equation (2) is satisfied, then Equation (1) is holds.

The above content describes the definition of the Dirichlet process, but still does not give an accurate representation, the model cannot be directly applied to the relevant algorithm, and the sampling of the Dirichlet process cannot be realized. Therefore, in the actual use process, the Dirichlet process is expressed by using three different forms of construction. The following mainly introduces the Chinese restaurant process (CRP).

The Dirichlet process is constructed as follows [18]: suppose a Chinese restaurant can accommodate infinitely many tables, and the number of tables is represented by K, θ_i means the customers entering the restaurant and ϕ_k is the table where the customers are seated. The first customer θ_1 is seated at the first table ϕ_1, the probability that the i-th customer θ_i is seated at the k-th table ϕ_k is proportional to the number of customers m_k on this table; the probability of a new table is proportional to α_0, at this time the number of tables K increases by 1, $\phi_K\sim G_0$ and $\theta_i=\phi_k$.

Figure 1 shows the structure of CRP, where the big circle represents the dining table, its unique code is ϕ_k, and the surrounding $\theta(i=1,2,3,\ldots)$ are the customers who are seated.

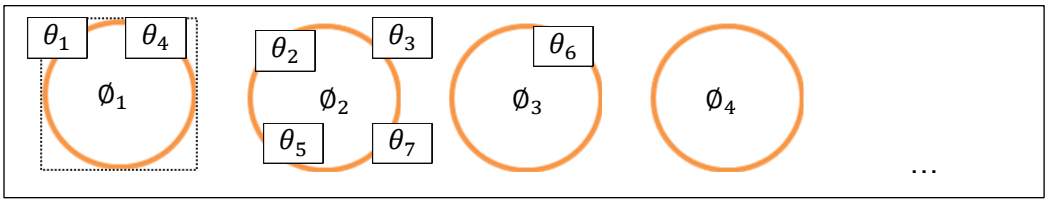


Figure 1. Chinese restaurant process

Let z_i is the indicator factor of the i-th cluster parameter variable θ_i, that is $\theta_i=\phi_{z_i}$, we can get [18]

$$z_i\,|\,z_i,\ldots,z_{i-1},\alpha_0,G_0\sim\sum_k^K\frac{m_k}{i-1+\alpha_0}\delta(z_i,k)+\frac{\alpha_0}{i-1+\alpha_0}\delta(z_i,\bar{k}) \quad (3)$$

Where $\bar{k}$ represents an empty new cluster. It can be seen from the structure of the Dirichlet process that it has good clustering properties.

The Dirichlet process clusters the data with the same value into one class, but if the two sets of data are not equal, no matter how similar they are, the Dirichlet process cannot be used to achieve clustering. For this purpose, the Dirichlet process mixture model (DPMM) [18] is introduced.

In DPMM, the Dirichlet process is used as the prior distribution of the data, so $X=\{x_1,x_2,\ldots x_n\}$ is a collection of observation data, and DPMM can cluster the observation data x_i, and each cluster is represented by a probability density function $f(\theta_i)$. DPMM can be represented by the following model:

$$G \sim DP(\alpha_0, G_0) \tag{4}$$

$$\theta_i \mid G \sim G \tag{5}$$

$$x_i \mid \theta_i \sim f(\theta_i) \tag{6}$$

G is a prior distribution of θ_i obtained by the Dirichlet process; θ_i is the cluster parameter, which is used to describe the probability distribution of each cluster $f(\theta_i)$. This is an infinite mixed model, which is different from clustering methods such as K-means. The number of parameters θ_i is not specified, but equals the number of observed data x_i. If the cluster parameters of two data are equal, that is $\theta_i=\theta_j$, then x_i and x_j belong to the same class.

The directed graph model representation of DPMM is shown in Figure 2 [11], hollow circles represent variables, shaded circles represent observations, rounded rectangles represent parameters or basic distributions, rectangles represent iterative cycles, and numbers in the lower right corner of the rectangles represent iterations times.

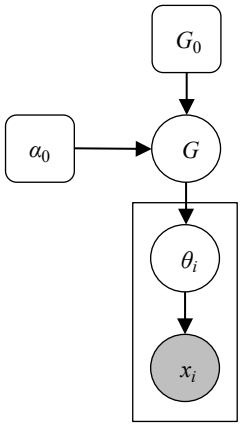


Figure 2. Graphical Model Representation of DPMM

Assuming that there is an observation data set $X=\{x_1,x_2,\ldots x_n\}$ that obeys DPMM, since the observation data is conditionally independent, the order of appearance of the observation data is not considered when performing cluster analysis on the observation data. To achieve cluster analysis is to obtain the indicator factor z_i of each data. Convention: when the upper or lower corner of a variable in the text is marked with the symbol "\", $Z_{\setminus i}$ is a set composed of the remaining data after being removed z_i from $Z=\{z_1,z_2,\ldots,z_n\}$. When the indicator factors $Z_{\setminus i}$ of other data are known, according to the Bayesian formula, the conditional distribution of z_i

$$p(z_i \mid x_1,x_2,\ldots,x_N,Z_{\setminus i},\lambda,\alpha_0) \propto (z_i \mid Z_{\setminus i},\alpha_0)\, p(x_i \mid z_1,\ldots z_N,X_{\setminus i},\lambda) \tag{7}$$

The above formula $p(z_i \mid Z_{\setminus i},\alpha_0)$ can be represented by CRP in the Dirichlet process. Since the observations are interchangeable, the i-th observation data can be regarded as the last observation. If there are already K categories about $Z_{\setminus i}$, the number of the observation data in each category is $n_{\setminus i}$. Then the first term in Equation (7) is

$$z_i \mid Z_{\setminus i},\alpha_0 \sim \sum_k^K \frac{n_k^{\setminus i}}{n-1+\alpha_0}\delta(z_i,k) + \frac{\alpha_0}{n-1+\alpha_0}\delta(z_i,\bar{k}) \tag{8}$$

The second item: if the i-th observation data chooses the k-th category, that is $z_i = k$, then there is

$$p(x_i \mid z_i = k, x_{\setminus i}, \lambda) = p\left(x_i \mid \{z_j = k, j \neq i\}, \lambda\right) = \frac{\int f(x_i \mid \theta) \prod_{Z_j = k, j \neq i} f(x_i \mid \theta) g(\theta \mid \lambda) d\theta}{\prod_{Z_j = k, j \neq i} f(x_i \mid \theta) g(\theta \mid \lambda) d\theta} \tag{9}$$

If a new class is chosen by x_i, namely $z_i = \bar{k}$,

$$p\left(x_i \mid z_i = \bar{k}, x_{\setminus i}, \lambda\right) = p(x_i \mid \lambda) = \int f(x_i \mid \theta) g(\theta \mid \lambda) d\theta \tag{10}$$

So, the right side of Equation (7) can be expressed as,

$$\sum_k^K \frac{n_k^{\setminus i}}{n - 1 + \alpha_0} \cdot p\left(x_i \mid \{z_j = k, j \neq i\}, \lambda\right) \delta(z_i, k) + \frac{\alpha_0}{n - 1 + \alpha_0} \int p(x_i \mid \theta) g(\theta \mid \lambda) d\theta \delta\left(z_i, \bar{k}\right) \tag{11}$$

Combining formulas in Equation (8) - (11), the Gibbs sampling process of DPMM can be obtained. Among them, $Z^{(t)}$ is used to describe the classification result of the observation data at the t-th round of sampling, $K(t)$ represents the number of clusters at this time, and the sampling result $Z^{(t-1)}$, $K^{(t-1)}$, $\alpha_0^{(t-1)}$ of the t-1-th is inputted, the t-th sample is based on the following process:

1. Make $\alpha_0 = \alpha_0^{(t-1)}$, $Z = Z^{(t-1)}$, for each data $x_i (i = 1, \ldots, n)$, z_i is sampled.

   $$f_k(x_i) = p(x_i \mid z_i = k, X_{\setminus i}, \lambda) \tag{12}$$

   $$f_{\bar{k}}(x_i) = p\left(x_i \mid z_i = \bar{k}, X_{\setminus i}, \lambda\right) \tag{13}$$

   a) For the existing K clusters, the likelihood estimation of the observed data is calculated for each class.

   b) The z_i is sampled according to the following distribution:

   $$p(z_i \mid x_1, \ldots, x_n, Z_{\setminus i}, \lambda, \alpha_0) \sim \frac{1}{Z_i}\left[\sum_k^K n_k^{\setminus i} f_k(x_1) \delta(z_i, k) + \alpha_0 f_{\bar{k}}(x_i) \delta\left(z_i, \bar{k}\right)\right] \tag{14}$$

   where, $Z_i = \sum_k^K n_k^{\setminus i} f_k(x_1) + \alpha_0 f_{\bar{k}}(x_i)$, $n_k^{\setminus i}$ is the amount of existing data in the k-th cluster. If $z_i = \bar{k}$, then K increases by 1.

2. Detecting the amount of observed data in each class. If the total number of observed data in a certain class is 0, remove the class and decrease K by 1.
3. If $\alpha_0 \sim \Gamma(a, b)$ is the initial sampling, update it according to the method of literature [20] , and the sampling relationship is as follows:

   $$\alpha_0^{(t)} \sim p(\alpha_0 \mid K, n, a, b) \tag{15}$$

Using CRP to describe the above sampling process is as follows: a customer comes in and distributes the table according to the probability of formula in Equation (11). If the customer chooses a new table, add a new table for the restaurant, and increase the number of tables by 1. After assigning tables to all customers, check to see if any tables are free, if so, remove that table from the restaurant first, and decrement the total number of tables seated by 1.

The following is a brief introduction to principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE).

PCA [21] is a classic data dimensionality reduction technique, which reduces the original information as much as possible. It is generally defined as: the orthogonal projection of data on a low-dimensional linear space, so that the variance of the data obtained after projection is maximized or a linear projection that minimizes the average projection cost. For the convenience of description, first, assume that the d-

dimensional observation data set is expressed as $X=\{x_1,\ldots,x_n\}, x_i \in R^d$. In the PCA method, the principal components in the low-dimensional space are represented by calculating the eigenvectors corresponding to the first *k* largest eigenvalues of the observed data *covariance* matrix. Then, using these eigenvectors, the original data is projected into the main subspace, so that the original data has a high degree of discrimination in the subspace. Algorithm flow:

1. De-average (that is, decentralize the data by column);
2. Calculate the covariance matrix $\boldsymbol{X}^*\boldsymbol{X}$;
3. Use eigenvalue decomposition or singular value decomposition (SVD) method to find the eigenvalue and eigenvector of $\boldsymbol{X}^*\boldsymbol{X}$;
4. Sort the eigenvalues from large to small, and select the largest *k* among them (that is, the top *k* with the largest variance);
5. Transform the data into the subspace constructed by *k* feature vectors.

The t-SNE algorithm uses the conditional probability distribution instead of the traditional distance representation and uses the distance similarity relationship between data points in high-dimensional and low-dimensional spaces to achieve dimensionality reduction on the premise of better maintaining the internal structure of the original data [22].

1. The similarity probability of the original data, that is, the probability of x_i being adjacent to x_j each other:

$$p_{j|i}=\frac{\exp\left(-\left\|x_i-x_j\right\|^2 \Big/ 2\sigma_i^2\right)}{\sum_{k\neq 1}\exp\left(-\left\|x_k-x_l\right\|^2 \Big/ 2\sigma_i^2\right)} \tag{16}$$

and $p_{i|i}=0$; σ_i is the variance of the Gaussian distribution; when the distance between x_i and x_j is closer, the value of $p_{j|i}$ is smaller, and the bigger the opposite.

2. In symmetric SNE, the distance between the discrete points x_i in the original data and other data points x_j is very far, so the joint probability distribution of x_i is small, and it is expressed by the following formula:

$$p_{ij}=\frac{p_{j|i}+p_{i|j}}{2n} \tag{17}$$

3. In the low-dimensional target space, the similarity probability of data is defined by t distribution of degree of Freedom 1:

$$q_{ij}=\frac{\left(1+\left\|y_i-y_j\right\|^2\right)^{-1}}{\sum_{k\neq l}\left(1+\left\|y_i-y_j\right\|^2\right)^{-1}} \tag{18}$$

where y_i is the form of the data points x_i dimensionality-reduced.

4. Use the relative entropy (KL) distance to measure whether the data distribution after dimension reduction is the same as the data distribution in the original high-dimensional space. The objective function is:

$$C=\sum_i KL\left(P_i \| Q_i\right)=\sum_i\sum_j p_{ij}\log\frac{p_{ij}}{q_{ij}} \tag{19}$$

P_i and Q_i are the joint distribution probabilities in the original data space and the dimensionally reduced data space, respectively.

5. Use the gradient descent method to optimize the objective function :

$$\frac{\delta C}{\delta y_i}=4\sum_j\left(p_{ij}-q_{ij}\right)\left(y_i-y_j\right)\left(1+\left\|y_i-y_j\right\|^2\right)^{-1} \tag{20}$$

Specific algorithm flow:

Input: $n-d$ -dimensional vectors $X=\{x_1,x_2,\ldots.x_n\}$, fixed value perplexity is prep, number of iterations is n_{iter}, learning rate is L, momentum coefficient is $\beta(t)$.

Output: low-dimensional data $Y=\{y_1,y_2,\ldots,y_n\}$.

Step 1: The variance σ_i of the point x_i is calculated by the binary search method;

Step 2: $p_{j|i}$, p_{ij} of the pairs of data points are calculated using formula as in Equation (13) and (14);

Step 3: Initialize the low-dimensional data $Y=\{y_1,y_2,\ldots,y_n\}$;

Step 4: q_{ij} of the low-dimensional data is calculated by formula in Equation (18);

Step 5: Calculate $\frac{\delta C}{\delta y_i}$;

Step 6: Update low-dimensional data, $y^t=y^{t-1}+\eta\frac{\delta C}{\delta y_i}+\beta(t)\left(y^{t-1}-y^{t-2}\right)$;

Step 7: Repeat Step 4 – Step 6 until the set number of iterations is reached.

PCA-DPMM and TSNE-DPMM

Due to the high dimensionality of the original data, DPMM is directly used for cluster analysis, and through the CRP process Gibbs sampling algorithm, each data must be randomly selected for each cycle, and the parameters of each class are updated at the same time, and the conditions for each class are selected for each data. The increased computational complexity of the probability leads to a long running time of the algorithm; while PCA and t-SNE are used to compress and simplify the data while minimizing data loss, thereby reducing the interference of noise points in the clustering process and removing redundancy; Reduce the computational complexity of the clustering process, save memory, and make the clustering more efficient and better.

Therefore, this paper uses high-dimensional data: (1) PCA for dimension reduction processing: first, the data matrix is solved, using the SVD method to solve the covariance matrix, and then generating the principal components by the eigenvalues and eigenvectors to; finally, DPMM clustering analysis was carried out for principal components. (2) The internal structure of the data as much as possible is maintain by t-SNE, the similarity in high-dimensional space is calculated by Gaussian distance, and the similarity in low-dimensional space is calculated by t-distribution to achieve dimensionality reduction. DPMM cluster analysis was performed on the dimensionally reduced data at last.

The flow chart of PCA-DPMM is as shown in Figure 3:

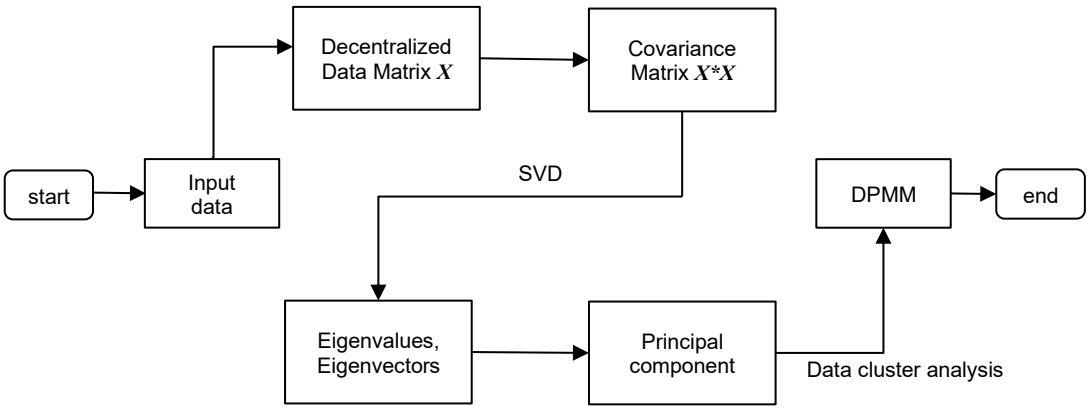


Figure 3. Flowchart for the proposed PCA-DPMM

Specific steps for PCA-DPM sampling:

Step 1: The original data extracts the principal components $Y=\{y_1,y_2,\ldots,y_n\}$ through PCA. We assume

that each cluster of the principal components obeys the Gaussian distribution:

$$y_k \sim N(\mu_k, \Sigma_k) \tag{21}$$

where, μ_k represents the parameter (mean value) of the Gaussian distribution of the k-th cluster, Σ_k is the covariance of the existing data of the k-th cluster.

Step 2: Suppose the number of initial classes in advance $K = 5$ (this data can be any positive integer, because the number of clusters is adaptively updated according to the amount of data during the operation of the algorithm). The corresponding initial indicator factors sequence Z for all data is obtained randomly, and the amount of data n_k in each cluster is counted according to Z, the division sequence of the initial cluster is obtained n; each cluster is recorded as c_i, $i = 1, 2, \ldots, K$. Set the number of iterations, loop once to get the index factor z_i of each data x_i.

Step 3: For each piece of data x_i, update z_i and n_i, the indicator factor z_i of each data x_i is selected according to the conditional probability of Equation (11), then select the existing class k, that is, the probability of its indicator factor $z_i = k$:

$$\begin{aligned} p(x_i \mid z_i = k, x_{\setminus i}, \lambda) &= \frac{\int f(x_i \mid \theta) \prod_{Z_j = k, j \neq i} f(x_i \mid \theta) g(\theta \mid \lambda) d\theta}{\prod_{Z_j = k, j \neq i} f(x_i \mid \theta) g(\theta \mid \lambda) d\theta} \\ &= \frac{1}{2\pi^{d/2} \cdot \sqrt{|\Sigma_k|}} \exp\left[-\frac{(x_i - u_k)^T \cdot \sum_i^{-1} \cdot (x_i - u_k)}{2} \right] \end{aligned} \tag{22}$$

x_i selecting a new cluster $z_i = \bar{k}$:

$$p(x_j \mid Z_j = \bar{k}, x_{\setminus i}, \lambda) = p(x_i \mid \lambda) = \int f(x_i \mid \theta) g(\theta \mid \lambda) d\theta = \frac{1}{(2\pi)^d} \exp\left(-\frac{\sum x_i^2}{4} \right) \tag{23}$$

where d is the dimension of the data. According to DPMM, each cluster of data obeys d dimensional Gaussian distribution, and the cluster parameter θ is the mean vector u_k of the Gaussian distribution, $f(x_i \mid \theta)$ is used to describe the probability distribution of each cluster. This is an infinite mixed model, and the number of parameters θ is not specified but related to the observed data x_i.

Therefore, the index factor of each data is selected according to the following formula:

$$\sum_k^K \frac{n_k^{\setminus i}}{n - 1 + \alpha_0} \cdot \frac{1}{2\pi^{d/2} \cdot \sqrt{|\Sigma_k|}} \exp\left[-\frac{(x_i - u_k)^T \cdot \sum_i^{-1} (x_i - u_k)}{2} \right] \cdot \delta(z_j, k) + \frac{\alpha_0}{n - 1 + \alpha_0} \cdot \frac{1}{(2\pi)^d} \exp\left(-\frac{\sum x_i^2}{4} \right) \cdot \delta(z_j, \bar{k}) \tag{24}$$

If the data x_i selects the k-th cluster, n_k will increase by 1; the amount of data in the original cluster will be reduced by 1; if the data x_i selects a new cluster, the number of classes K will increase by 1, and $n_{k+1} = 1$.

Update μ_k:

$$\mu_k(i) \sim N\left(\frac{u_0 + \sum c_k(i)}{1 + n_k}, \frac{1}{1 + n_k} \right) \tag{25}$$

The i-th component of μ_k is represented by $\mu_k(i)$, $c_k(i)$ is the attribute of the data in the k-th cluster, u_0 is the mean vector of the initial cluster distribution, and takes the zero vector.

Update α_0 according to [20]:

$$eta \sim Beta(\alpha_0 + 1, n) \tag{26}$$

$$s \sim Binomial\left(1, \frac{n}{\alpha_0 + n} \right) \tag{27}$$

$$\alpha_0 \sim Gamma\left(a+K-s, \frac{1}{b-\log(eta)}\right) \tag{28}$$

where eta, s, α_0 are randomly generated from the three distributions of beta distribution (Beta), binomial distribution (Binomial) and gamma distribution (Γ) respectively; initial value $a=1, b=2$.

Step 4: Repeat the above Step 2 and Step 3 until the set number of iterations is reached.
Step 5: Count the final clustering results of each run, and calculates the corresponding evaluation indicators.

The improved DPMM clustering algorithm combined with TSNE is shown in Figure 4 below.

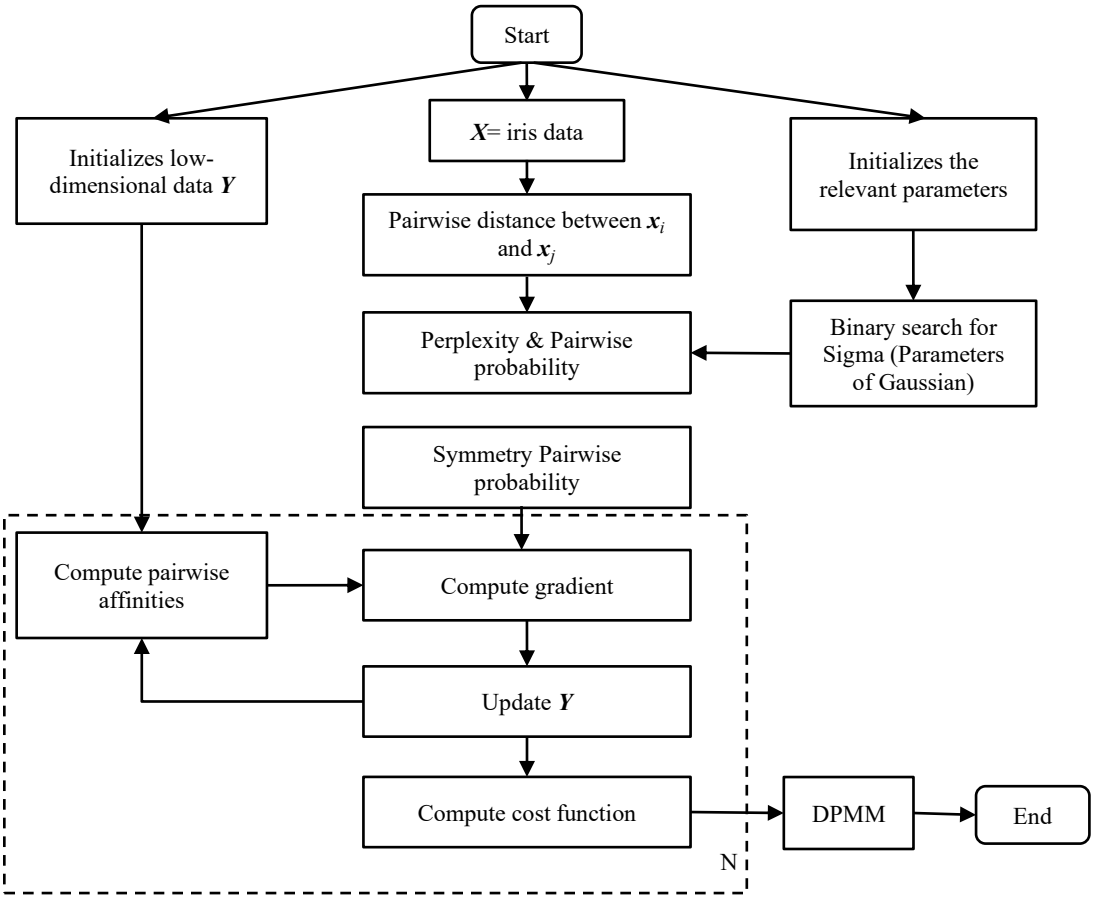


Figure 4. Implementation flow chart for the proposed DPMM

For specific sampling steps, change Step 1 of PCA-DPMM sampling in Figure 3 to use t-SNE for dimension reduction, and perform DPMM cluster analysis on the reduced dimension data.

Experimental Setup

To further confirm the validity of the models proposed in the study, PCA-DPMM, TSNE-DPMM will be evaluated based on different parameters and experimental settings. Three different simulations with different number of iterations, different learning rates in the t-SNE process, and different number of iterations. Each simulation is described in detail as follows:

- For the DPMM process, different iteration times are set. In this part, indicators such as Precision, Recall, F1 score, Specificity, and running time are used to evaluate and analyze the impact of different iterations on clustering accuracy.
- For the PCA -DPMM process, different iterations are set, and the above indicators are also used to

compare the clustering accuracy.

- In the nonlinear t-SNE dimension reduction process, set different learning rates and the number of iterations of gradient descent, evaluate and analyze the changes of the learning rate and the number of iterations of gradient descent on the data structure after dimension reduction, and further affect the clustering accuracy.
- The accuracy and time complexity of the three models are compared under different clustering iterations to evaluate the performance of the model.

The simulations described in this section are all in Python language, the experimental environment is Windows 10 operating system, the memory is 16G, and the main frequency is 3.1GHz. The data used in the experiment is public data-Iris data, which has four attributes: Sepal Length, Sepal Width, Petal Length, and Petal Width. Labels 1, 2, and 3 represent Setosa, Versicolor, and Virginical respectively, with a total of 150 pieces of data information. Table 1 shows the parameters involved in each experiment.

Table 1. Parameters for the proposed DPMM

Parameter Explanation	Parameter Value
Data volume	n =150
Each data dimension	d =4
Number of initial clusters	K = 5 (Random)
Indicator factor of i-th mixture cluster (z_i)	$Z = \{z_1, z_2, \cdots, z_K\}$
Amount of data in each cluster (nn)	$nn = \{n_1, n_2, \cdots, n_K\}$
Mean value of k-th mixture cluster	μ_k
Covariance matrix of k-th mixture cluster	Σ_k
Data of k-th mixture cluster	c_k
i-th attribute of the data in the k-th mixture cluster	$c_k(i)$
i-th cluster of μ_k	$\mu_k(i)$
Concentration parameter	α_0

Table 1 shows the parameters involved in each experiment. In Table 1, the parameters are all variables in the third part of the formula. The initial number of clusters, K can be any value as it does not affect the clustering results. The 150 data points are randomly assigned to initial clusters based on the given K, where Z represents the index set of initial clusters for each data point. nn represents the set of the number of elements in each cluster. Assuming each cluster follows a multidimensional Gaussian distribution, μ_k and Σ_k represent the mean vector and covariance matrix of the Gaussian mixture, respectively. c_k refers to the data in the k-th cluster, and correspondingly, $c_k(i)$ and $\mu_k(i)$ represent their i-th components.

Each simulation will define six types of performance metrics for evaluation. Count the Confusion Matrix of each clustering result , calculate Accuracy(A) , Average number of components(E_K) , Time complexity(T) of the entire model based on the Confusion Matrix , Precision(P) , Recall(R) , Specificity(S) , F1 score(F_1) and average value. The F1 score indicator combines the output results of Precision and Recall. The value ranges from 0 to 1. 1 represents the best output of the model, and 0 represents the worst output of the model. Tables 2-3 list the parameters involved in all evaluation metrics.

Table 2. Parameters involved in model evaluation

Parameter	Remarks
a_{ij}	The number of data c_i were predicted to be c_j
$iter$	Number of iterations of DPMM
L	Learning rate of t-SNE
n_{iter}	Number of iterations of t-SNE
c_1	Setosa
c_2	Versicolor

Parameter	Remarks
c_3	Virgin
$P(c_i)$	Precision of c_i
$R(c_i)$	Recall of c_i
$F_1(c_i)$	F1 score of c_i
$S(c_i)$	Specificity of c_i
K	Number of last clusters

In Table 2, the variables are parameters related to evaluating the model in equations (29) to (37). a_{ij} represents the number of data points that truly belong to cluster i and are classified into cluster j. When i equals j, it means the data points are correctly assigned to clusters. $iter$ denotes the number of iterations in the Gibbs sampling process of the Dirichlet Process Mixture Model (DPMM). L represents the learning rate in the t-SNE process, which controls the speed at which data points move during dimensionality reduction. Here, K represents the number of clusters in the final clustering result.

Table 3. Confusion Matrix

Confusion Matrix		Predicted Value			
		c_1	c_2	…	c_K
Actual Value	c_1	a_{11}	a_{12}	…	a_{1K}
	c_2	a_{21}	a_{22}	…	a_{2K}
	…	…	…	…	…
	c_K	a_{K1}	a_{K2}	…	a_{KK}

In the Confusion Matrix, $c_1,\cdots c_k$ represent the cluster name, the sum of each row represents the number of real samples of this cluster, the sum of each column represents the number of samples predicted to be of this cluster, and the data a_{ij} has the same meaning as in Table 2. The specific expressions of the relevant indicators are as follows, A represents the accuracy of the clustering results, while $P(c_i)$、$R(c_i)$、$F_1(c_i)$ and $S(c_i)$ respectively indicate the Precision, Recall, F1 score, and Specificity of cluster label c_i.

$$A = \frac{\sum_{i=1}^{K} a_{ii}}{n} \tag{29}$$

$$P(c_i) = \frac{a_{ii}}{\sum_{i=1}^{K} a_{ji}} \tag{30}$$

$$R(c_i) = \frac{a_{ii}}{\sum_{i=1}^{K} a_{ij}} \tag{31}$$

$$F_1(c_i) = \frac{2P(c_i)R(c_i)}{P(c_i)+R(c_i)} \tag{32}$$

$$S(c_i) = \frac{n - a_{ii} - \sum_{j=2}^{K}(a_{ij} + a_{ji})}{n - \sum_{j=1}^{K} a_{ij}} \tag{33}$$

Due to the multi-class problem, the precision, recall, F1 score, and specificity of each cluster evaluation metric are used to represent the overall clustering results with their average values. The following formulas for calculating the mean of the indicators are given:

$$P_{mean} = \frac{1}{K}\sum_{i=1}^{K} P(c_i) \tag{34}$$

$$R_{mean} = \frac{1}{K}\sum_{i=1}^{K} R(c_i) \tag{35}$$

$$F_{1mean} = \frac{1}{K}\sum_{i=1}^{K} F_1(c_i) \tag{36}$$

$$S_{mean} = \frac{1}{K}\sum_{i=1}^{K} S(c_i) \tag{37}$$

Where $i = 1, 2, \cdots, K$.

Results and Discussion

To evaluate the effectiveness of the proposed models, different models will be evaluated based on different perspective: different of iterations on DPMM, different learning rate on t-SNE, different of iterations on t-SNE, and different of iterations on DPMM, PCA-DPMM, TSNE-DPMM. These perspectives will be evaluated based on various performance metrics. After finding the best setting from the next four sections, the proposed PCA-DPMM and TSNE-DPMM will be compared with existing model.

The Clustering Results of DPMM

When using DPMM's CRP to cluster iris data, if the data is not standardized, the clustering results are quite different, indicating that the clustering model is invalid when multiple dimension indicators are at different scale levels, which further shows that the complexity of multisource data clustering and the different indicators have a great impact on the clustering effect. Therefore, the multidimensional data is first standardized before clustering, and then different iterations are set to compare the clustering results.

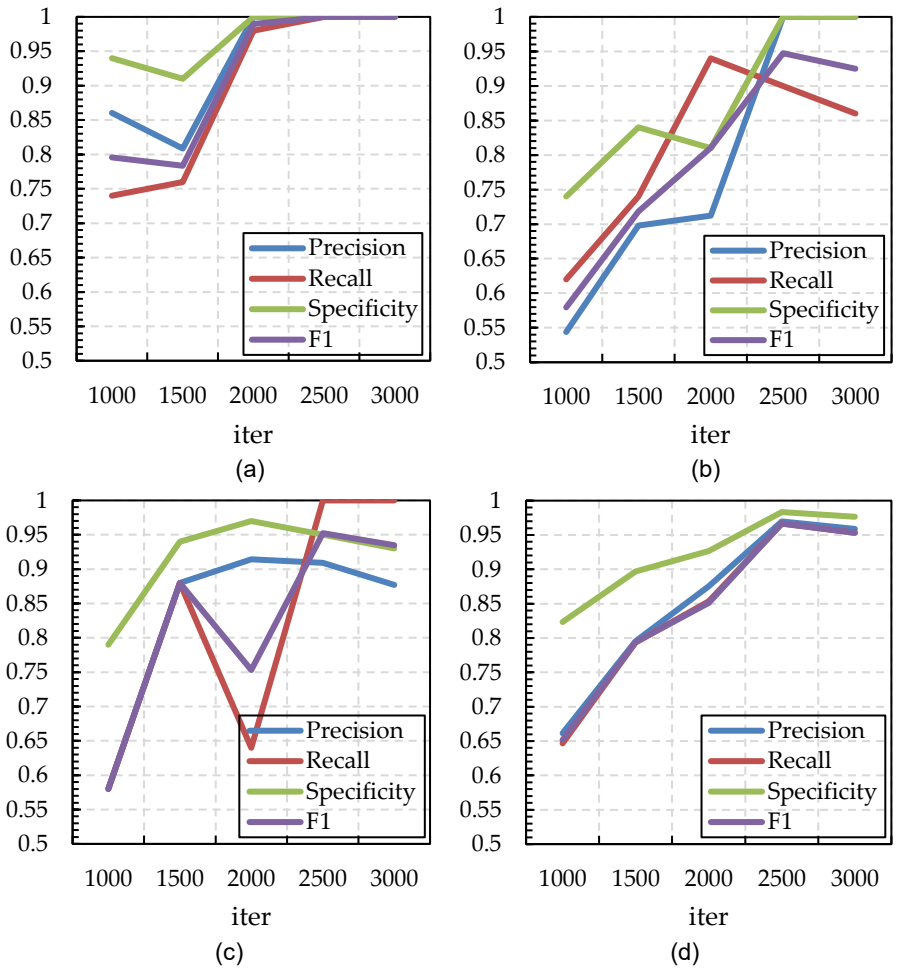


Figure 5. The indicators for (a) Setosa, (b) Versicolor, (c) Virginical and (d) the average value of DPMM with different iterations.

Figure 5 shows that the classification accuracy of DPMM for Setosa, Versicolor, and Virginal is improved within a certain range as the number of iterations increases. When the number of iterations is between 2500-3000, Precision, Recall, F1 score, and Specificity reach Highest. Below we give the clustering results of different iteration times, the average number of clusters, the running time (Seconds) from the list, and the timing is in seconds here, and Accuracy.

Table 4. DPMM clustering results with different iterations.

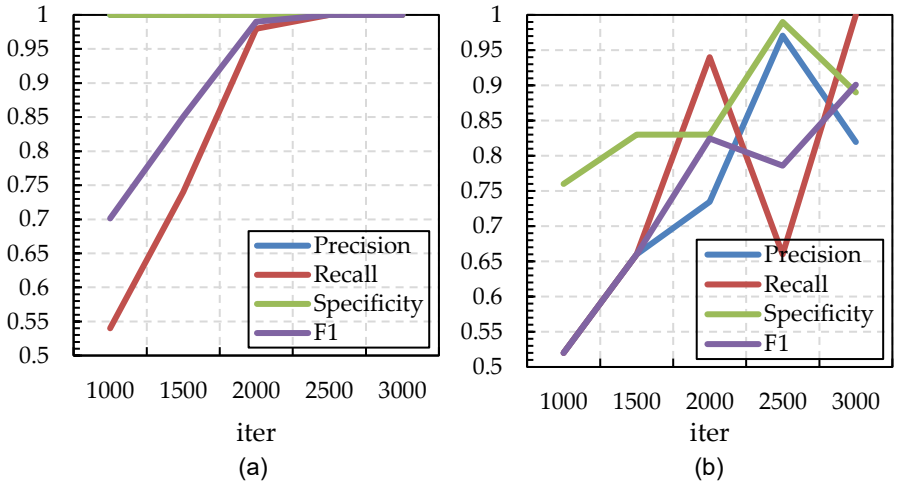
iter	Clustering results	Average of clusters	Accuracy	Time (seconds)
1000	[43,57,50]	3.54	0.65	82.31
1500	[47,53,50]	3.96	0.79	124.69
2000	[49,66,35]	3.01	0.85	162.40
2500	[50,45,55]	3.11	0.97	210.79
3000	[50,43,57]	3.01	0.95	256.54

In Table 4, the iterations of the DPMM clustering algorithm (*iter*) are set to 1000, 1500, 2000, 2500, and 3000. The second column represents the clustering results. For example, [43, 57, 50] indicates that there are three clusters in the result, with the respective numbers of data points in each cluster being 43, 57, and 50. Since the number of clusters is obtained after each iteration, the third column shows the average number of clusters. The fourth and fifth columns represent the accuracy of the clustering results and the running time for *iter* iterations, respectively. The time unit is in seconds. Overall, it can be observed that when the number of iterations exceeds 2500, the DPMM achieves the highest accuracy in clustering the iris data. The average number of clusters stabilizes around 3. However, as the number of iterations increases, the running time also increases. The accuracy reaches its peak between 2500 and 3000 iterations.

The clustering results of PCA-DPMM

Below we will give the clustering results of iris data using PCA-DPMM. In the PCA-DPMM clustering process, the Iris data is reduced to 2 dimensions by PCA (more than 90% of the cumulative contribution rate has been included), Variance contribution rate of each variable:

c_0 = 0.7296244541329989
c_1 = 0.2285076178670174
c_2 = 0.036689218892828786
c_3 = 0.005178709107154802 5
The cumulative contribution rate of the second component exceeds 90%.



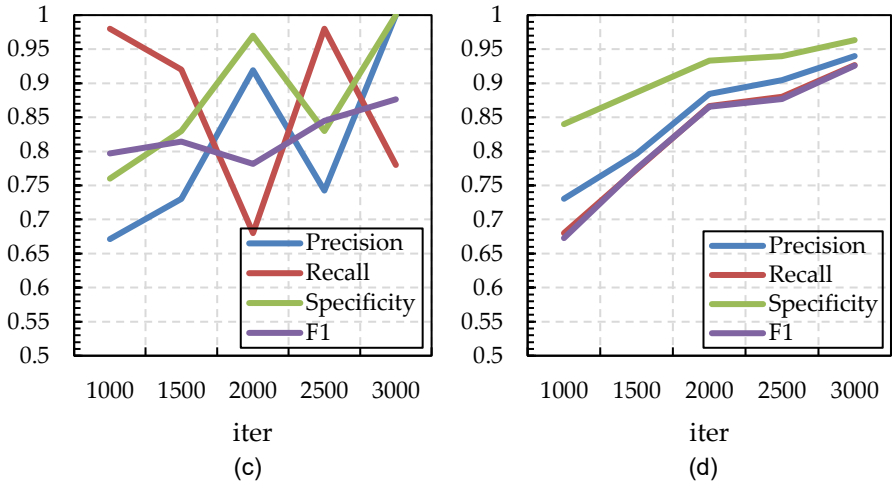


Figure 6. The indicators for (a) Setosa, (b) Versicolor, (c) Virginical and (d) the average value of PCA-DPMM with different iterations

Figure 6 shows that PCA- DPMM has improved the accuracy of Setosa and average classification as the number of iterations increases. Accuracy, Precision, Recall, F1 score, Specificity are all on the rise. As for the unstable classification accuracy of Versicolor and Virginal, we will give the clustering results of different iterations, the average number of clusters, running time and Accuracy from the list below.

Table 5. PCA-DPMM clustering results with different iterations

iter	Clustering results	Average of clusters	Accuracy	Time (seconds)
1000	[27,50,73]	2.65	0.68	92.49
1500	[50,63,37]	3.96	0.77	114.84
2000	[49,35,66]	3.26	0.87	166.82
2500	[50,66,34]	3.00	0.88	209.75
3000	[50,39,61]	3.11	0.93	332.97

The variables in Table 5 have the same meanings as in Table 4, but they represent the results of PCA-DPMM clustering. From Table 5, it can be observed that as the number of iterations increases, the accuracy of PCA-DPMM clustering on the iris data gradually improves, reaching the highest value at 3000 iterations. The average number of clusters remains stable at around 3. However, similar to the DPMM, the running time of PCA-DPMM also increases with the number of iterations. Overall, the findings suggest that increasing the number of iterations can improve the accuracy of PCA-DPMM clustering on the iris data. However, it is important to note that the running time also increases as more iterations are performed.

The Clustering results of TSNE-DPMM

Below, we will present the clustering results of the iris dataset using TSNE-DPMM. The algorithm consists of two stages. In the first stage, t-SNE is used for dimensionality reduction with different settings for the learning rate (L) and the maximum number of iterations (n_{iter}) in the gradient descent. The learning rate controls the speed at which data points move during the dimensionality reduction process, while the n_{iter} determines the number of repeated optimization steps for stable embedding. In this case, t-SNE is applied to reduce the iris data to 2 dimensions, followed by standardization of the reduced data. In the second stage, DPMM (Dirichlet Process Mixture Model) is applied to cluster the standardized data. The parameters used are the same as before.

Tables 6 and 7 present the maximum iteration count and learning rate for t-SNE, which were adjusted using the method of controlled variables. In the TSNE-DPMM clustering process, with t-SNE reducing the iris data to 2 dimensions, we observe that the sparser the distribution of the reduced data, the better the clustering results. Additionally, larger gaps between data distributions result in more accurate

clustering. Generally, a perplexity value between 30-32 is recommended for the best performance. Based on the comparison results, the DPMM clustering stage is conducted with 2500 iterations. The clustering results of TSNE-DPMM can be compared accordingly.

Table 6. TSNE-DPMM clustering results with different iterations of t-SNE (L=1200)

n_{iter}	Precision	Recall	F1 score	Accuracy	Specificity	Time (seconds)
800	0.8883	0.8733	0.8808	0.8733	1	221.33
900	0.9268	0.9267	0.9267	0.9267	1	222.45
1000	0.9678	0.9667	0.9672	0.9667	1	243.34
1100	0.9231	0.9	0.9114	0.9	1	203.72

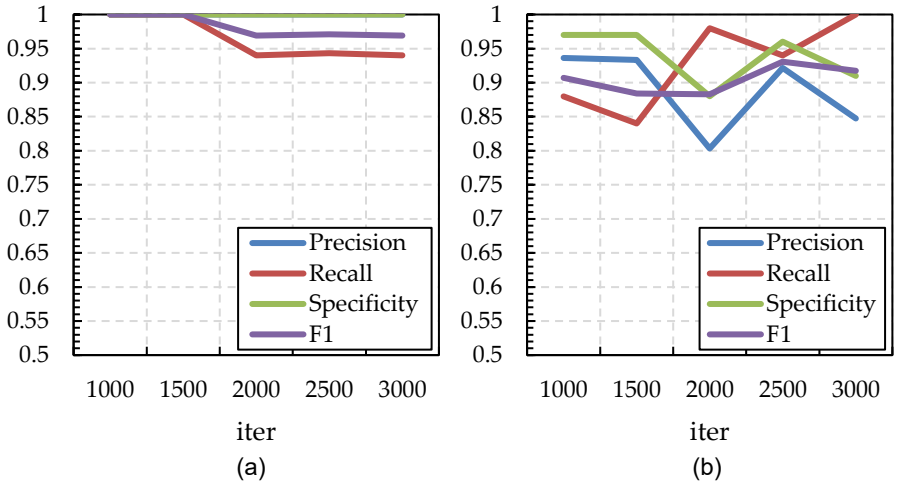
In Table 6, the learning rate (L) for the t-SNE stage was set to 1200, and the maximum iteration count (n_{iter}) was varied to analyze the experimental results. It was found that the maximum iteration count affects the distribution of the reduced data, thus impacting the clustering accuracy. However, it has little effect on the runtime. When the maximum iteration count for t-SNE is set to 1000, the various indicators are relatively high, with an accuracy of 0.9667. The runtime is slightly longer, but the difference is not significant.

Table 7. TSNE-DPMM clustering results with different learning rate of t-SNE (n_{iter} =1000)

L	Precision	Recall	F1 score	Accuracy	Specificity	Time (seconds)
1000	0.8902	0.8733	0.8808	0.8733	1	197.63
1100	0.9045	0.9267	0.9267	0.9267	1	197.63
1200	0.9678	0.9667	0.9672	0.9667	1	2 43.34
1300	0.8576	0.8467	0.8521	0.8467	0.9882	256.02

According to the results of Table 6, in Table 7, let's set the maximum iteration count (n_{iter}) for the t-SNE stage to 1000 and vary the learning rate (L). It is observed that the learning rate in the t-SNE dimensionality reduction process also affects the distribution of the reduced data, consequently impacting the clustering accuracy. However, it does not have a significant impact on the runtime. The highest performance in terms of various indicators and accuracy (0.9667) is achieved when the learning rate is set to 1200.

Figure 7 below, the t-SNE learning rate is set to 1200, and the maximum number of iterations is set to 1000.



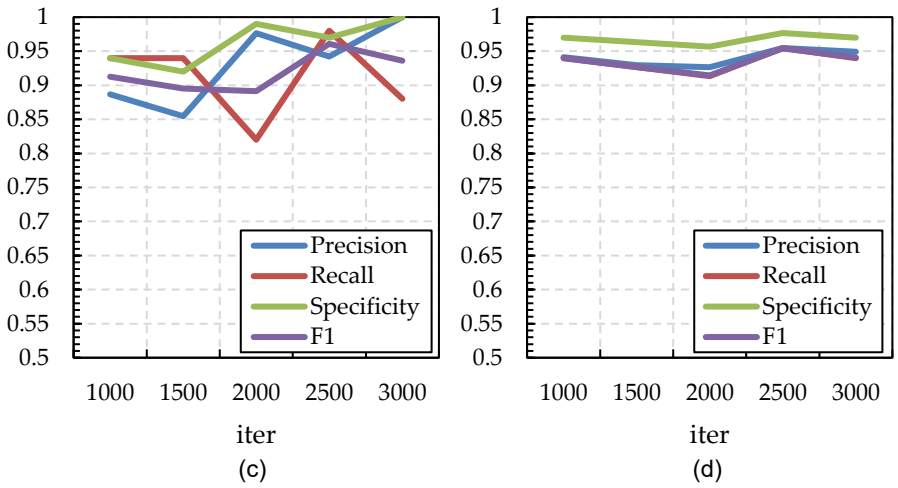


Figure 7. The indicators for (a) Setosa, (b) Versicolor, (c) Virginical and (d) the average value of TSNE-DPMM with different iterations

Figure 7 shows that when the t-SNE learning rate is 1200 and the number of iterations is 1000, the four indicators of TSNE-DPMM for Setosa, Versicolor and Virginical are basically stable above 0.8, and the average value exceeds 0.9, especially when the number of DPMM iterations is 2500, it reaches the highest; among them, the Specificity remains above 0.95. The following t-SNE learning rate is 1200, the maximum number of iterations is 1000, and the clustering results, average number of clusters, running time and accuracy of different clustering iterations are given from the list.

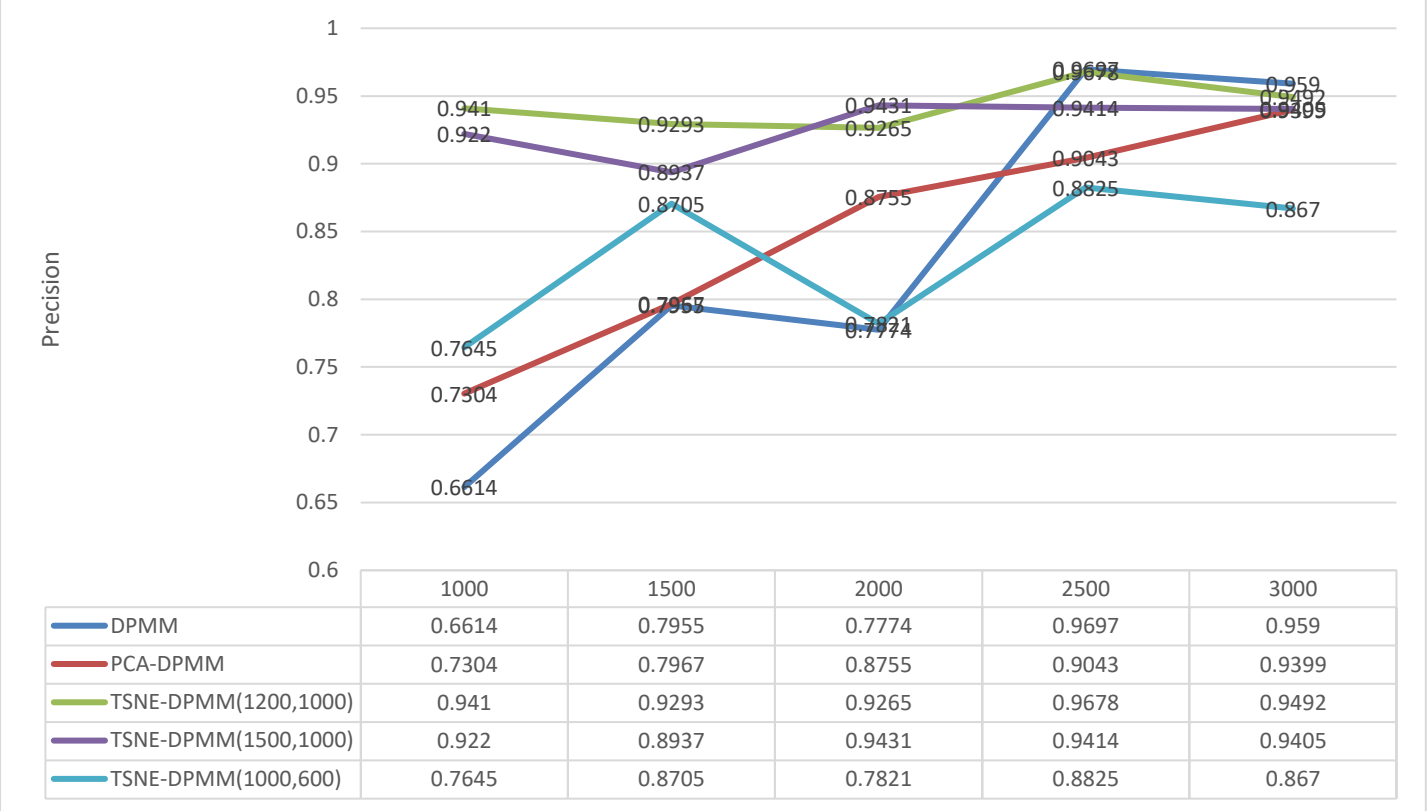
Table 8. TSNE-DPMM clustering results with different iterations

$iter$	Clustering results	Average of clusters	Accuracy	Time(Seconds)
1000	[50,47,53]	3.54	0.94	101.93
1500	[50,45,55]	3.33	0.93	143.48
2000	[47,61,42]	3.21	0.91	176.74
2500	[50,41,49]	3.19	0.95	228.77
3000	[47,56,44]	3.13	0.94	257.76

In Table 8, the learning rate for the t-SNE stage was set to 1200 and the iteration count was set to 1000. The iteration count ($iter$) for the DPMM clustering stage was varied to examine its impact on the clustering accuracy of TSNE-DPMM on the iris dataset. The results show that as the clustering iteration count increases, the Accuracy of TSNE-DPMM remains above 0.91, reaching a peak of 0.95 at 2500 iterations. The average number of clusters stabilizes at around 3. Additionally, as the iteration count increases, the runtime of the algorithm also shows an increasing trend.

The Clustering Results of Three Models

Using the improved DPMM clustering algorithm for cluster analysis of iris data, the results of the cluster analysis have an important relationship with the number of iterations, so it is necessary to comprehensively consider the clustering effect under different $iter$ values. The following compares the stability and classification average precision of the three models from the line chart.



	1000	1500	2000	2500	3000
DPMM	0.6614	0.7955	0.7774	0.9697	0.959
PCA-DPMM	0.7304	0.7967	0.8755	0.9043	0.9399
TSNE-DPMM(1200,1000)	0.941	0.9293	0.9265	0.9678	0.9492
TSNE-DPMM(1500,1000)	0.922	0.8937	0.9431	0.9414	0.9405
TSNE-DPMM(1000,600)	0.7645	0.8705	0.7821	0.8825	0.867

Figure 8. Precision of three models

For the three models, the number of iterations is respectively 1000, 1500, 2000, 2500, and 3000 times, and the corresponding precision line chart is given according to Figure 8. It can be seen from the figure that the classification precision of the three models has an upward trend as the number of iterations increases. Among them, the accuracy of the clustering results of DPMM increases the fastest as the number of iterations increases, reaching the highest at 2500 iterations; PCA-DPMM shows Straight up; in the TSNE-DPMM model, when the parameter perplexity of t-SNE dimensionality reduction is 30, the learning rate is 1200-1500, and the maximum number of iterations is 1000, when performing DPMM clustering on the data after dimensionality reduction, the number of iterations is 1000-3000, the accuracy is basically stable at 0.92-0.97. For TSNE-DPMM, by changing the dimensionality reduction parameters learning rate and the maximum number of iterations, it is found that when L=1200, n_{iter}=1000, the clustering effect of the model is the best, and it is relatively stable.

Table 9, the three commonly used clustering algorithms are K-means, DBSCAN, and hierarchical clustering. In this article, we will compare the clustering accuracy of these algorithms and perform clustering on the iris dataset.

Table 9. Comparison of clustering accuracy on iris data set

Model	Accuracy	Model	Accuracy
K-means	0.8933	DPMM	0.9697
DBSCAN	0.6667	PCA-DPMM	0.9387
Hierarchical Clustering	0.2333	TSNE-DPMM (L=1200,n_{iter}=1000)	0.9678

According to Table 9, the clustering algorithm proposed in this article has a higher accuracy compared to the other three. It does not require the pre-determination of the number of clusters, unlike the other three algorithms which require an accurate cluster count. Hierarchical clustering is clearly not suitable for classifying the iris dataset, and the density-based clustering algorithm DBSCAN also does not

perform well. Only K-means shows relatively higher accuracy. This further emphasizes that the proposed algorithm in this article has advantages in clustering high-dimensional datasets with multiple indicators.

Consider the effects of $iter$ =1000, 1500, 2000, 2500, 3000, and finally get the histograms of different indicators of the clustering results under different $iter$ values, as shown in Figure 9.

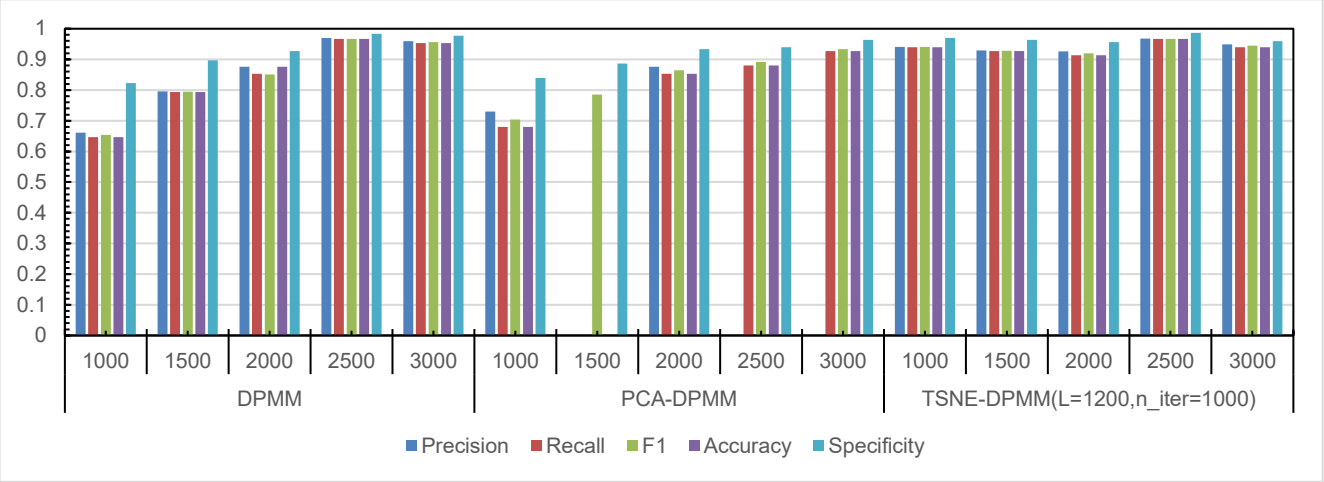


Figure 9. TSNE-DPMM clustering results with different iterations

In general, the clustering results of the iris data set are compared. From the comparison of the four indicators in Figure 9, it can be seen that when the number of iterations is 2500, the clustering accuracy of the three models is relatively high. Among them, the accuracy rate of PCA-DPMM is the lowest under the set number of iterations. Compared with the stability of the clustering results under different iterations, TSNE-DPMM is the most stable, and each index basically exceeds 0.92; at the same time, after comparing the running time in Table 10, we found that the running time of the three models is similar to that of The number of clustering iterations set is positively correlated, so that when the amount of data is very large, a lower number of iterations can be selected to reduce the running time and the clustering accuracy will not be greatly affected.

Table 10. The running time of Iris data with different $iter$ values

Model	1000	1500	2000	2500	3000
DPMM	82.31	124.69	162.40	210.79	256.54
PCA-DPMM	92.49	114.84	193.14	209.75	332.97
TSNE-DPMM (L=1200,n_{iter}=1000)	101.93	162.74	187.80	243.34	256.76

In table 10, the learning rate is 1200 and the maximum number of t-SNE iterations is 1000.

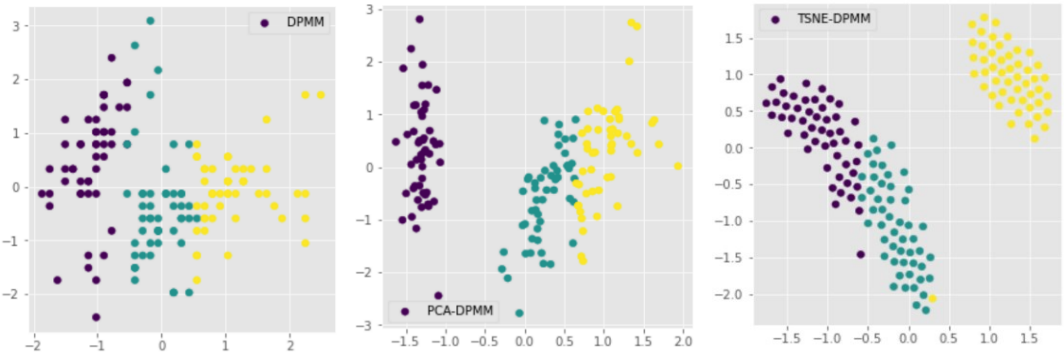


Figure 10. The scatter plot of three models ($iter$ =2500)

From the comparison in Figure 10, it can be seen that the visual effect of direct classification using DPMM will be worse, the data is scattered, and the division between classes is not obvious; after the introduction of PCA, the classification effect will be better, and the division between classes is relatively clear. It can be seen that TSNE-DPMM has the best classification effect.

Conclusions

In order to bridge the gap of low computational efficiency for large-scale data and to develop a more efficient and accurate clustering method for multi-source high-dimensional data analysis, in this study, based on DPMM, improved PCA-DPMM and TSNE-DPMM are proposed. By analyzing the non-parametric Bayesian model DPMM, it is found that DPMM does not require setting the number of clusters and the clustering data can adaptively update based on the data volume. Through clustering comparison on the iris dataset, it was observed that as the number of iterations increases, the clustering accuracy is relatively high. However, as the data volume increases, achieving high clustering accuracy requires increasing the number of iterations, which leads to a linear increase in algorithm running time. Different parameter settings and iteration numbers are used. When the number of iterations is set to 1000, the accuracies of DPMM and PCA-DPMM are 0.6614 and 0.7304 respectively, while TSNE-DPMM has reached 0.941. The corresponding running times are 82 seconds, 92 seconds and 102 seconds respectively. When the number of iterations increases to 2500, the accuracy of the three models reaches the highest values: 0.9697, 0.9043 and 0.9678. The running times are 210 seconds, 209 seconds and 243 seconds respectively. Through comparison of accuracy and running time, it is found that compared with the other two models, TSNE-DPMM exhibits the most stable clustering accuracy and superior visualization effect, and the category division is clearer. This allows us to set a lower number of iterations for large data volumes, thereby improving algorithm efficiency while ensuring accuracy. In future research, the time complexity of TSNE-DPMM needs to be further improved to better apply it to actual scenarios such as real-time monitoring of data anomalies and IoT anomaly detection.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

Acknowledgment

The author would like to thank his mentor and all his helpers for their continued support.

References

[1] Gao, M. Y., Wang, J. & Yang, J. (2023). Research into the relationship between personality and behavior in video games, based on mining association rules. *Mathematics*, *11*(3), 1-13.
[2] Li, W. P., Cao, Y. & Li, L. J. (2021). Orthogonal wavelet transform KCA in fault diagnosis. *Journal of Vibration and Shock*, *40*(07), 291-296.
[3] Badhera, U., Verma, A. & Nahar, P. (2022). Applicability of K-medoids and K-means algorithms for segmenting students based on their scholastic performance. *Journal of Statistics and Management Systems*, *25*(7), 1621-1632.
[4] Li, T. & Ma, J. W. (2023). Dirichlet process mixture of Gaussian process functional regressions and its variational EM algorithm. *Pattern Recognition*, *134*, 109129.
[5] Huang, Y. G., Zhang, S. S. & Liu, H. J. (2022). Urban road traffic state identification based on Gaussian mixture model clustering algorithm. *Modern Electronics Technique*, *45*(07), 168-173.
[6] Liu, Y. & Nandram, B. (2022). Sampling methods for the concentration parameter and discrete baseline of the Dirichlet Process. *Statistics in Transition New Series*, *23*(4), 21-36.
[7] Saraiva, E. F., Suzuki, A. K. & Milan, L. A. (2017). Identifying differentially expressed genes using the Polya urn scheme. *Communications for Statistical Applications and Methods*, *24*(6), 627-640.
[8] Rogers, D. & Winkel, M. (2022). A Ray–Knight representation of up-down Chinese restaurants. *Bernoulli*, *28*(1), 689-712.
[9] Bhattacharya, I. & Ghosal, S. (2021). Bayesian multivariate quantile regression using Dependent Dirichlet Process prior. *Journal of Multivariate Analysis*, *185*, 104763.
[10] Yao, Y., Li, Z. Q., Zhao, J. H. & Wu, L. N. (2019). Adaptive chaotic MIMO radar based on DPMM clustering and Kalman filtering technique. *Chaos: An Interdisciplinary Journal of Nonlinear*

Science, *29*(11).
[11] Li, H. & Zhang, N. (2019). A Sticky Hierarchical Dirichlet Process Clustering Method. *Statictics & Information Forum*, *34*(08), 20-26.
[12] Lai, Y. P., Guan, W. B., Luo L. J., Ruan, Q., Ping, Y., Song, H. P., Meng, H. Y. & Pan, Y. (2021). Extended variational inference for Dirichlet process mixture of Beta‐Liouville distributions for proportional data modeling. *International Journal of Intelligent Systems*, *37*(7), 4277-4306.
[13] Peng, X. & He, J. F. (2023). Flora analysis based on Dirichlet polynomial process model and k-means. *Chinese Journal of Bioinformatics*, 1-16.
[14] Chen, Y. M., Liu, W. F., Kong, M. X. & Zhang, G. L. (2020). A modeling and tracking algorithm of finite mixture models for multiple extended target based on the GLMB filter and Gibbs sampler. *Acta Automatica Sinica*, *46*(07), 1445-1456.
[15] Duan, T., Pinto, J. P. and Xie, X. (2019). Parallel clustering of single cell transcriptomic data with split-merge sampling on Dirichlet process mixtures. *Bioinformatics*, *35*(6), 953-961.
[16] Xu, X., Lin, H. J. Liu, Y. Y., & Hu, B. (2022). On-line fault detection method of hydraulic turbine combining PCA and adaptive K-Means clustering. *Journal of Electronic Measurement and Instrumentation*, *36*(03), 260-267.
[17] Han, Z. M., Zhang, M. M., Li, M. Q., Duan, D. G. & Chen, Y. (2019). Flow hierarchical dirichlet process for complex topic modeling, *Chinese Journal of Computers*, *42*(07), 1539-1552.
[18] Li, Y., Schofield, E., & Gönen, M. (2019). A tutorial on Dirichlet process mixture modeling. *Journal of Mathematical Psychology*, *91*, 128-144.
[19] Zhou, Z. M. & Gao, S. Y. (2014). A Survey on Hierarchical Dirichlet Process Principle and its Application. *Computer Applications and Software*, *31*(08), 1-5+41.
[20] Teh, Y., Kurihara, K., & Welling, M. (2007). Collapsed variational inference for HDP. *Advances in Neural Information Processing Systems*, 20.
[21] Khoo, T. H., Pathmanathan, D., & Dabo-Niang, S. (2023). Spatial autocorrelation of global stock exchanges using functional areal spatial principal component analysis. *Mathematics*, *11*(3), 674.
[22] Lu, W. P. & Yan, X. F. (2022). Industrial process data visualization based on a deep enhanced t-distributed stochastic neighbor embedding neural network. *Assembly Automation*, *42*(2), 268-277.