

---

# FLEXIBLE BIVARIATE BETA MIXTURE MODEL: A PROBABILISTIC APPROACH FOR CLUSTERING COMPLEX DATA STRUCTURES

---

Yung-Peng Hsu, Hung-Hsuan Chen\*  
National Central University  
yungpeng1998@gmail.com, hhchen1105@acm.org

February 28, 2025

## ABSTRACT

Clustering is essential in data analysis and machine learning, but traditional algorithms like  $k$ -means and Gaussian Mixture Models (GMM) often fail with nonconvex clusters. To address the challenge, we introduce the Flexible Bivariate Beta Mixture Model (FBBMM), which utilizes the flexibility of the bivariate beta distribution to handle diverse and irregular cluster shapes. Using the Expectation Maximization (EM) algorithm and Sequential Least Squares Programming (SLSQP) optimizer for parameter estimation, we validate FBBMM on synthetic and real-world datasets, demonstrating its superior performance in clustering complex data structures, offering a robust solution for big data analytics across various domains. We release the experimental code at <https://github.com/yung-peng/MBMM-and-FBBMM>.

## 1 Introduction

Clustering is a fundamental task in data analysis and machine learning that aims to group data points into clusters such that the points in the same cluster are more similar than those in other clusters. This unsupervised learning method is widely used in various applications, including image analysis, information retrieval, text analysis, bioinformatics, and many more [1, 2, 3, 4]. Clustering helps uncover the underlying structure of the data, facilitates data summarization, and sometimes serves as a preprocessing step for other algorithms [2].

Despite its widespread use, one of the primary challenges many traditional clustering algorithms face is that they often assume that the data points form clusters with convex shapes. For example, centroid-based algorithms like  $k$ -means and distribution-based models like Gaussian Mixture Models (GMM) typically produce clusters that are hyperspherical or ellipsoidal [5]. Although this assumption simplifies the clustering process, it restricts the flexibility of these models to handle complex data distributions that do not conform to convex shapes.

This convexity constraint can lead to suboptimal clustering results, especially when the data inherently possesses nonconvex structures. Examples include data points that form concentric circles, crescent shapes, or other intricate patterns. Traditional methods may fail to correctly group these points, leading to less optimal clustering results and loss of valuable structural information.

We propose the Flexible Bivariate Beta Mixture Model (FBBMM) to address these limitations. Unlike conventional models, FBBMM leverages the flexibility of the bivariate beta distribution, which can accommodate a wide range of shapes, including convex, concave, and other irregular forms. This adaptability is crucial for accurately capturing the proper structure of complex datasets.

The FBBMM offers several advantages over traditional clustering algorithms. First, versatile cluster shapes: FBBMM can model clusters with various shapes using the bivariate beta distribution, providing a better fit for nonconvex data structures. Second, soft clustering: like GMM, FBBMM assigns a probability to each data point to belong to different

---

\*Corresponding author

Table 1: Comparison of Clustering Algorithms

Algorithm	Type	Shape	Assignment	Noise Robustness	Generative
<i>k</i> -means	Centroid-based	Convex	Hard	Low	No
DBSCAN	Density-based	Arbitrary	Hard	High	No
Agglomerative	Hierarchical	Arbitrary	Hard	Medium	No
GMM	Distribution-based	Convex	Soft	Low	Yes
MBMM	Distribution-based	Flexible	Soft	Medium	Yes
FBBMM	Distribution-based	Flexible	Soft	Medium	Yes

clusters, offering a better and more flexible representation of data point memberships. Third, generative capability: FBBMM, being a generative model, can generate new data points that resemble the original data, which is helpful in data augmentation and simulation tasks.

In this paper, we detail the formulation of FBBMM, describe its probability function and the parameter estimation process using the Expectation Maximization (EM) algorithm, and demonstrate its effectiveness through experiments on synthetic and real-world datasets. The results indicate that FBBMM outperforms traditional models in handling nonconvex clusters and provides a robust framework for flexible and accurate data clustering.

The rest of the paper is organized as follows. In Section 2, we review famous clustering algorithms of various types. Section 3 presents the bivariate beta distribution, the FBBMM model, and the parameter learning process. Section 4 compares FBBMM with famous clustering algorithms using both synthetic and open datasets. Finally, we conclude our work and discuss the limitations of FBBMM and future work in Section 5.

## 2 Related Work

Clustering algorithms can be categorized into four types: centroid-based, density-based, hierarchical, and distribution-based methods. Each has its strengths and limitations, as discussed below, followed by a comparison with our proposed Flexible Bivariate Beta Mixture Model (FBBMM).

Centroid-based methods like *k*-means [6] are computationally efficient but assume convex clusters, making them unsuitable for nonconvex data. Density-based methods like DBSCAN [7] identify clusters of arbitrary shapes and are robust to noise but depend heavily on hyperparameter tuning. Hierarchical methods, such as agglomerative clustering [8], build a tree-like structure and do not require pre-specifying cluster numbers but are computationally expensive and struggle with large datasets. Distribution-based models like Gaussian Mixture Models (GMM) [9] handle soft clustering but are limited to elliptical cluster shapes. MBMM [5] addresses this by assuming multivariate beta distributions, allowing nonconvex clusters but restricting correlations to be positive.

FBBMM overcomes these limitations by employing the flexible bivariate beta distribution, enabling it to model both convex and nonconvex clusters and handle positive and negative correlations. It supports soft clustering and is generative, capable of producing new data points for tasks like data augmentation. Although FBBMM handles bivariate data, this limitation can be mitigated using dimension reduction techniques such as PCA or autoencoders.

As shown in Table 1, FBBMM’s flexibility in cluster shapes and ability to handle positive and negative correlations make it a more versatile and effective clustering method compared to traditional approaches.

## 3 Flexible Bivariate Beta Mixture Model

The Flexible Bivariate Beta Mixture Model (FBBMM) leverages the flexibility of the bivariate beta distribution to model clusters with a variety of shapes, addressing the limitations of traditional clustering methods, which often assume convex cluster shapes. In this section, we describe the FBBMM in detail, including the PDF of the flexible bivariate beta distribution, the FBBMM density function, and the parameter learning process.

### 3.1 Bivariate Beta Distribution

The definition of the beta distribution is unique. However, the beta distribution is only defined on a univariate variable within the interval  $[0, 1]$  or  $(0, 1)$ . When the number of variates is greater than one, the definition of the multivariate beta distribution is ambiguous [10, 5]. Eventually, we use the flexible bivariate beta distribution based on the definition

Table 2: Definition of Variables in FBBMM

Variable	Definition
$N$	Number of data points
$C$	Number of clusters
$\mathbf{x}_n$	A data point (indexed by $n$ ), $\mathbf{x}_n = [x_{n,1}, x_{n,2}]$
$z_n$	A latent variable indicating the cluster membership of $x_n$ , $z_n \in \{1, 2, \dots, C\}$
$\boldsymbol{\pi}$	The probabilities of a data point belongs to the cluster $1, 2, \dots, C$ , $\boldsymbol{\pi} = [\pi_1, \dots, \pi_C]$
$\alpha_j^c$	The $j$ th parameter of the bivariate beta distribution for the cluster $c$ , $j \in \{1, \dots, 4\}$ , $c \in \{1, \dots, C\}$

provided by [11] because this definition is one of the few that allows for a positive or negative correlation between covariates, making the cluster shapes more flexible.

Our bivariate beta distribution is defined based on Dirichlet distribution. Let  $(U_1, U_2, U_3, U_4)$  be a set of random variables sampled from Dirichlet distributions with parameters  $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ . The PDF is given by:

$$f(u_1, u_2, u_3, u_4) = \frac{u_1^{\alpha_1-1} u_2^{\alpha_2-1} u_3^{\alpha_3-1} u_4^{\alpha_4-1}}{B(\boldsymbol{\alpha})}, \quad (1)$$

where  $\alpha_{ij} \geq 0$  and  $B(\boldsymbol{\alpha})$  is the normalization term, as defined below.

$$B(\boldsymbol{\alpha}) = \frac{\prod_i \Gamma(\alpha_i)}{\Gamma(\sum_i \alpha_i)}. \quad (2)$$

The support of the Dirichlet distribution  $u_j$ s must follow the following two conditions:  $0 \leq u_j \leq 1$  and  $u_1 + u_2 + u_3 + u_4 = 1$ . By replacing  $u_4$  in the above formula with  $1 - u_1 - u_2 - u_3$ , we get a PDF involving three random variables:

$$f(u_1, u_2, u_3) = \frac{u_1^{\alpha_1-1} u_2^{\alpha_2-1} u_3^{\alpha_3-1} (1 - u_1 - u_2 - u_3)^{\alpha_4-1}}{B(\boldsymbol{\alpha})}. \quad (3)$$

Next, we define two random variables  $X$  and  $Y$ :

$$X = U_1 + U_2, \quad Y = U_1 + U_3. \quad (4)$$

The marginal distribution of the Dirichlet distribution is defined as a beta distribution. Thus, the PDF of the bivariate beta distribution of  $X$  and  $Y$  can be written as a function involving only  $u_1$  as follows.

$$\begin{aligned} BB\epsilon(x, y | \boldsymbol{\alpha}) &= \int_{\Omega} f(u_1, u_2, u_3) du_1 \\ &= \frac{1}{B(\boldsymbol{\alpha})} \int_{\Omega} u_1^{\alpha_1-1} (x - u_1)^{\alpha_2-1} (y - u_1)^{\alpha_3-1} (1 - x - y + u_1)^{\alpha_4-1} du_1, \end{aligned} \quad (5)$$

where  $\Omega = \{u_{11} : \max(0, x + y - 1) < u_{11} < \min(x, y)\}$ .

Different parameters  $\boldsymbol{\alpha}$  result in different bivariate beta distributions. The PDF mode changes according to the values of  $\boldsymbol{\alpha}$ . The mode is in the center if all the parameters  $\alpha_j$  are equal. If the value of  $\alpha_1$  becomes larger, the mode moves toward the upper right corner, i.e., the modes of the two varieties become larger. Figure 1 gives PDF examples using different  $\boldsymbol{\alpha}$ s.

### 3.2 Generative Process and Probability Density Function of FBBMM

We introduce the FBBMM from the perspective of a generative process. Figure 2 gives the plate notations of the observed and latent variables of the FBBMM, with the notations listed in Table 2. An observed random variable  $\mathbf{x}_n$  is assumed to be sampled by the following process. First, we sample a latent variable  $z_n$  from a multinomial distribution with parameters  $\boldsymbol{\pi} = [\pi_1, \dots, \pi_C]$ . The latent variable  $z_n$  represents the cluster ID of the data point  $\mathbf{x}_n$ . Next,  $\mathbf{x}_n$  is sampled from the flexible bivariate beta distribution with parameters  $\alpha_1^{z_n}, \alpha_2^{z_n}, \alpha_3^{z_n}, \alpha_4^{z_n}$ , the four parameters defining the bivariate beta distribution for the cluster  $z_n$ .

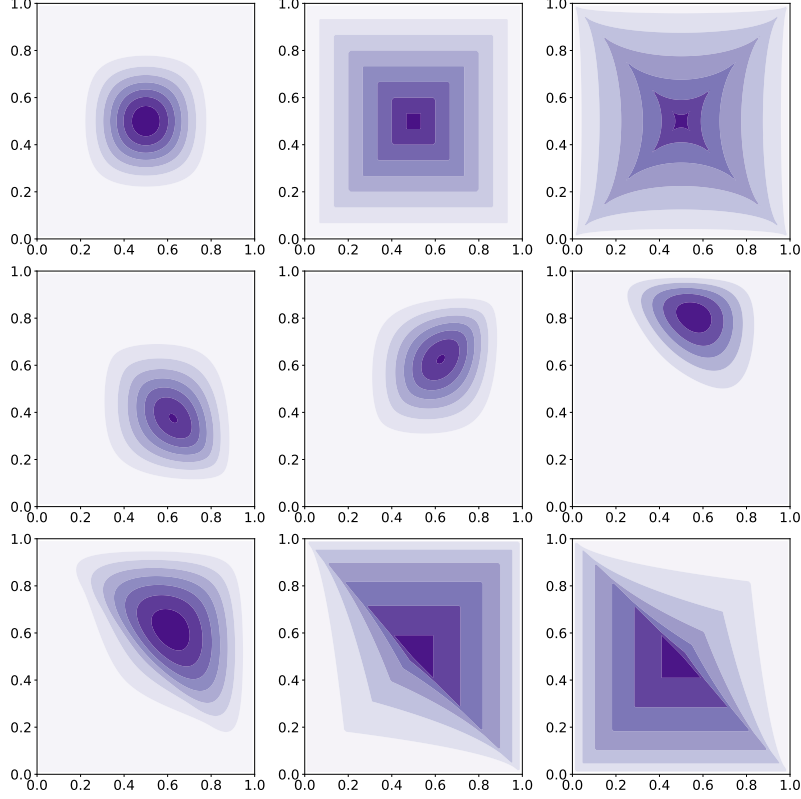


Figure 1: The PDF plots of the bivariate beta distribution with different parameters. The top row:  $\alpha = (3, 3, 3, 3)$ ;  $\alpha = (1, 1, 1, 1)$ ;  $\alpha = (0.8, 0.8, 0.8, 0.8)$ . The middle row:  $\alpha = (2, 4, 2, 2)$ ;  $\alpha = (4, 2, 2, 2)$ ;  $\alpha = (4, 2, 4, 0.5)$ . The bottom row:  $\alpha = (2, 2, 2, 0)$ ;  $\alpha = (1, 1, 1, 0.5)$ ;  $\alpha = (0.5, 1, 1, 1)$ . The shapes could be nonconvex (e.g., upper right subfigure). The covariates could be positively correlated (e.g., the middle center subfigure) or negatively correlated (e.g., the lower left subfigure), or non-correlated (e.g., the upper middle subfigure).

Assume that all data points  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$  are generated from an unknown parameterized FBBMM, the PDF of FBBMM is expressed as:

$$p(\mathbf{X}|\boldsymbol{\theta}) = \prod_{n=1}^N p(\mathbf{x}_n|\boldsymbol{\theta}) = \prod_{n=1}^N \sum_{c=1}^C \pi_c BB\epsilon(\mathbf{x}_n|\boldsymbol{\theta}_c), \quad (6)$$

where  $\mathbf{x}_n = [x_{n,1}, x_{n,2}]$  is a 2D data point,  $n \in \{1, \dots, N\}$ ,  $\boldsymbol{\theta}_c = \{\alpha_1^c, \alpha_2^c, \alpha_3^c, \alpha_4^c\}$  are the parameters of cluster  $c$ ,  $\boldsymbol{\theta}$  includes the parameters of all clusters, and  $\pi_c$  is the probability that a data point belong to the cluster  $c$ , thus  $\sum_{c=1}^C \pi_c = 1$ .

### 3.3 Parameter Learning for FBBMM

In practice, we only observe  $\mathbf{x}_1, \dots, \mathbf{x}_N$ , but the other variables  $\alpha_1^{1:C}, \alpha_2^{1:C}, \alpha_3^{1:C}, \alpha_4^{1:C}$ , and  $\pi_1, \dots, \pi_C$  are unknown. To learn the parameters of the FBBMM, we use the Expectation Maximization (EM) algorithm. Our objective is to find the parameters  $\boldsymbol{\theta}$  that maximize the likelihood function:

$$L(\boldsymbol{\theta}) = p(\mathbf{X}|\boldsymbol{\theta}) = \prod_{n=1}^N p(\mathbf{x}_n|\boldsymbol{\theta}) = \prod_{n=1}^N \sum_{c=1}^C \pi_c BB\epsilon(\mathbf{x}_n|\boldsymbol{\theta}_c). \quad (7)$$

Due to the numerical instability of multiplications when  $N$  is large, we take the logarithm of the likelihood function by convention to form the log-likelihood.

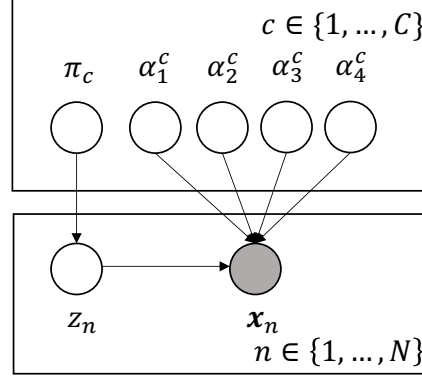


Figure 2: The plate notation of the flexible bivariate beta mixture model

$$\log(L(\boldsymbol{\theta})) = \sum_{n=1}^N \log \left( \sum_{c=1}^C \pi_c \text{BB}e(\mathbf{x}_n | \boldsymbol{\theta}_c) \right). \quad (8)$$

Assuming that we know the latent variable  $z_n$ , which indicates the membership of the cluster of each  $x_n$ , the complete log-likelihood is:

$$\log(L(\boldsymbol{\theta})) = \sum_{n=1}^N \sum_{c=1}^C I(z_n = c) (\log \pi_c + \log \text{BB}e(\mathbf{x}_n | \boldsymbol{\theta}_c)), \quad (9)$$

where  $I()$  is the indicator function, i.e., its output is 1 if  $z_n = c$  and 0 otherwise.

In practice, since  $z_n$  is unobservable, we compute  $\gamma_{n,c}$ , the expected probability that  $x_n$  belongs to cluster  $c$ .

$$\gamma_{n,c} = \frac{\pi_c \text{BB}e(\mathbf{x}_n | \boldsymbol{\theta}_c)}{\sum_{k=1}^C \pi_k \text{BB}e(\mathbf{x}_n | \boldsymbol{\theta}_k)}. \quad (10)$$

In the E-step of EM, we assume that all the parameters  $\boldsymbol{\theta}_c$ s and  $\pi_c$ s are correct and use them to compute  $\gamma_{n,c}$  (Equation 10). In the M-step, we update the parameters using maximum likelihood estimation. The update for  $\pi_c$  is:

$$\pi_c = \frac{1}{N} \sum_{n=1}^N \gamma_{n,c}. \quad (11)$$

For the parameters  $\boldsymbol{\theta}_c = \{\alpha_1^c, \alpha_2^c, \alpha_3^c, \alpha_4^c\}$  of each cluster  $c$ , we use the Sequential Least Squares Programming optimizer (SLSQP) to maximize the expected value of Equation 9 since there seems to be a lack of closed-form solutions.

$$E_{z_{1:N}}[\log(L(\boldsymbol{\theta}))] = \sum_{n=1}^N \sum_{c=1}^C \gamma_{n,c} (\log \pi_c + \log \text{BB}e(\mathbf{x}_n | \boldsymbol{\theta}_c)). \quad (12)$$

The algorithm 1 provides the pseudocode for parameter learning in FBBMM.

## 4 Experiments

This section presents the results of experiments that compare the performance of FBBMM with baseline clustering algorithms on different datasets. The compared methods include  $k$ -means, MeanShift, DBSCAN, Agglomerative Clustering, GMM, and MBMM. The experiments were carried out on synthetic and real-world datasets, including a structural dataset and an image dataset.

**Algorithm 1** FBBMM Parameter Learning

---

```

1: Input:  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ :  $N$  input data points, each data point  $\mathbf{x}_n$  is 2-dimensional;  $C$ : the number of clusters
2: Output: Final parameters  $\theta_{1:C} = \{\alpha_1^{1:C}, \alpha_2^{1:C}, \alpha_3^{1:C}, \alpha_4^{1:C}\}$ ;  $\pi = \{\pi_1, \dots, \pi_C\}$ 
3: Initialize parameters  $\theta_{1:C}$  and  $\pi$ 
4: Old_prob  $\leftarrow -\infty$ 
5: for  $i = 1$  to Epochs do
6:   // E-step
7:   Compute each  $\gamma_{n,c}$  by Equation 10
8:   Compute New_prob by Equation 9
9:   if  $|\text{New\_prob} - \text{Old\_prob}| < \epsilon$  then
10:     break
11:   end if
12:   Old_prob  $\leftarrow$  New_prob
13:   // M-step
14:   Compute  $\alpha_1^{1:C}, \alpha_2^{1:C}, \alpha_3^{1:C}, \alpha_4^{1:C}$  by maximizing Equation 12 using SLSQP
15:   Compute each  $\pi_c$  using Equation 11
16: end for

```

---

## 4.1 Experimental Setup

We preprocess the data such that the value of each feature is normalized: let  $\mathbf{x}_n = [x_{n,1}, \dots, x_{n,m}]$ , each  $x_{n,j}$  is normalized below.

$$x_{n,j} = 0.01 + \frac{(x_{n,j} - \min(x_{*,j})) (0.99 - 0.01)}{\max(x_{*,j}) - \min(x_{*,j})}, \quad (13)$$

where  $x_{*,j} = [x_{1,j}, x_{2,j}, \dots, x_{N,j}]$ , i.e., the  $j$ th feature of all instances.

## 4.2 Experiments on the Synthetic Datasets

The synthetic datasets were generated using scikit-learn to test the characteristics of different clustering algorithms. These datasets consist of five different shapes. Each dataset includes 500 two-dimensional data points.

Figure 3 compares the clustering results of  $k$ -means, MeanShift, DBSCAN, Agglomerative Clustering, GMM, MBMM, and FBBMM on five synthetic datasets.

The first dataset includes concentric circles. If a point from the outer circle is selected, the most distant data point is positioned on the opposite side of the same circle. This characteristic makes the synthetic dataset highly challenging for centroid-based and distribution-based methods to group the entire outer circle into a single cluster. As shown in the first row of Figure 3, density-based algorithms (DBSCAN) and Hierarchical clustering method (Agglomerative Clustering) and two beta distribution-based models (MBMM and FBBMM) successfully separate the two circles.

The second dataset contains two distant 2D Gaussian distributions with small variances in each dimension, and the third distribution has a large variance, located in the middle. Thus, several data points sampled from the third distribution are mixed with the first two distributions. Since the middle cluster has a wider spread, the centroid is far from some points within the same cluster, making certain clustering algorithms,  $k$ -means, MeanShift, and DBSCAN, misidentify some data points in the middle cluster as other clusters; details are in the second row of Figure 3.

The third and fourth datasets each comprise three 2D Gaussian distributions with isolated means. However, the two covariates are highly correlated: the covariates are negatively correlated for the third dataset and positively correlated for the fourth. As a result, data points are sometimes closer to those generated from other distributions. Thus,  $k$ -means, MeanShift, DBSCAN, and Agglomerative Clustering make errors on some data points. MBMM only handles data points whose covariates are positively correlated [5]. FBBMM and GMM are the only models that handle the two datasets well, as presented in the third and fourth rows of Figure 3.

Finally, the last dataset includes data points from three 2D Gaussian distributions with distant means and small variances in each dimension. Therefore, a data point is close to other data points within the same Gaussian distribution but far from others. All clustering algorithms perform well in this ideal case.

Overall, our proposed FBBMM performs well on all synthetic datasets.

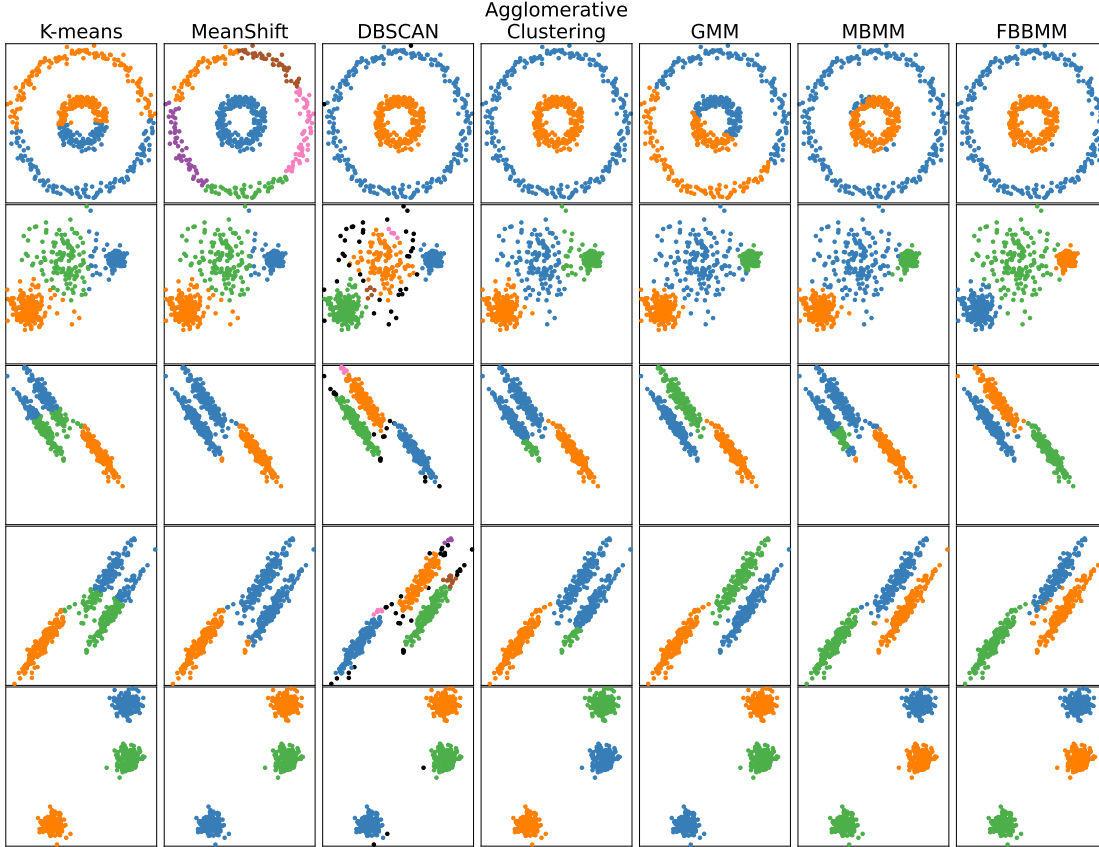


Figure 3: Clustering results on synthetic datasets

### 4.3 Experimented the Open Datasets

The open datasets include the wine dataset [12] and the MNIST dataset [13]. The wine dataset contains chemical analysis results of wines grown in the same region of Italy but derived from three different cultivars. There are 178 instances with 13 features. The second dataset, the MNIST dataset, comprises 70,000 grayscale images of handwritten digits (0-9), with each image having 28x28 pixels. The two datasets represent structural data and image data, respectively.

### 4.4 Evaluation Metrics for Open Datasets

We evaluate clustering results using three metrics: Clustering Accuracy (CA), Adjusted Rand Index (ARI), and Adjusted Mutual Information (AMI).

Clustering Accuracy is calculated as the number of correctly clustered data points divided by the total number of data points. Since clustering results and actual labels may not directly correspond, a mapping is performed before computing accuracy. For example, assume that we have a dataset with four data points whose labels are  $[a, a, b, b]$ , and a clustering algorithm produces the output with cluster IDs  $[b, b, a, a]$ . Despite the mismatch between the cluster IDs and the actual labels, the clustering is perfect because all points with the actual label  $a$  are grouped in cluster  $b$  and vice versa. We call a set of lists the *identical lists* if one list can be transformed into another list by permuting the labels. Thus, clustering accuracy is defined as the maximum accuracy among all identical lists of predicted cluster IDs [14].

$$CA(\mathbf{y}, \hat{\mathbf{y}}) := \max_{\forall \tilde{\mathbf{y}} \in P(\hat{\mathbf{y}})} \left\{ \frac{1}{N} \sum_{i=1}^N I(\tilde{y}_i = y_i) \right\}, \quad (14)$$

Table 3: Clustering on Original Features (13 Dimensions) and FBBMM (2 Dimensions)

Method	CA	ARI	AMI
k-means	0.702	0.371	0.423
MeanShift	0.697	0.469	0.483
DBSCAN	0.506	0.297	0.380
Agglomerative Clustering	0.674	0.371	0.436
GMM	0.725	0.435	0.436
MBMM	0.719	0.391	0.389
FBBMM (2D)	<b>0.983</b>	<b>0.947</b>	<b>0.927</b>

Table 4: Clustering on Reduced Features (2 Dimensions)

Method	CA	ARI	AMI
k-means (2D)	0.961	0.882	0.860
MeanShift (2D)	0.961	0.882	0.860
DBSCAN (2D)	0.910	0.846	0.833
Agglomerative Clustering (2D)	0.978	0.930	0.900
GMM (2D)	0.949	0.847	0.833
MBMM (2D)	0.809	0.526	0.588
FBBMM (2D)	<b>0.983</b>	<b>0.947</b>	<b>0.927</b>

where  $\mathbf{y} = [y_1, \dots, y_N]$  is the list of ground-truth labels,  $\hat{\mathbf{y}} = [\hat{y}_1, \dots, \hat{y}_N]$  is a list of predicted cluster IDs,  $P(\hat{\mathbf{y}})$  returns a set of all identical lists for  $\hat{\mathbf{y}}$ ,  $I(\cdot)$  is an indicator function, and  $\tilde{\mathbf{y}} = [\tilde{y}_1, \dots, \tilde{y}_N]$  is an identical list of  $\hat{\mathbf{y}}$ .

We also use the Adjusted Rand Index (ARI) and Adjusted Mutual Information (AMI) for evaluation. ARI and AMI are biased toward different types of clustering results: ARI prefers balanced partitions (clusters with similar sizes), and AMI prefers unbalanced partitions [15, 14].

#### 4.5 Results on the Open Datasets

Table 3 compares the clustering performance of FBBMM with six baseline clustering methods on the wine dataset. Since each of the six baseline methods can handle datasets with any number of features, we use the entire 13 features provided in the wine dataset. However, because FBBMM only handles datasets with bivariate variables, we use an autoencoder to reduce the original feature to two dimensions. As shown, FBBMM outperforms all baseline clustering methods.

We also project the dataset from the original 13-dimensional to 2-dimensional dataset using an autoencoder and apply baseline clustering algorithms on the 2-dimensional dataset. In doing so, we ensure a consistent evaluation environment that isolates the effects of the dimensionality reduction, enabling us to accurately assess the strengths and weaknesses of each method in this specific context.

Table 4 compares clustering performance on the wine dataset after dimension reduction. Probably because of autoencoder’s ability in extracting key feature combinations, all baseline methods improved. FBBMM still performs the best in all three evaluation metrics, demonstrating its superiority in clustering.

The MNIST dataset consists of 70,000 grayscale image. Due to the high dimensionality and a convolutional neural network (CNN)’s ability to handle images, we use a CNN as the feature extractor, followed by applying an autoencoder for dimension reduction. Since we are dealing with a clustering algorithm, we need to prevent CNN from learning information from the labels in the MNIST dataset. Thus, we use fashion-MNIST [13] to train a CNN. After training, we remove the last fully connected layer. Then, we pass MNIST to this trained CNN to convert an image into a  $1 \times 512$  dimensional vector. Subsequently, we feed this vector into an autoencoder to reduce the features to 2 dimensional.

Table 5 shows the clustering performance in clustering digits 1 and 7 after the feature reduction. FBBMM, again, achieves the best performance in all metrics, demonstrating its effectiveness in handling the digit recognition task.



Table 5: Clustering on MNIST Digits 1 and 7

Method	CA	ARI	AMI
k-means (2D)	0.971	0.889	0.813
MeanShift (2D)	0.975	0.903	0.832
DBSCAN (2D)	0.944	0.857	0.761
Agglomerative Clustering (2D)	0.973	0.897	0.823
GMM (2D)	0.970	0.883	0.806
MBMM (2D)	0.930	0.738	0.633
FBBMM (2D)	<b>0.976</b>	<b>0.907</b>	<b>0.841</b>

## 5 Discussion and Future Work

This paper introduces the Flexible Bivariate Beta Mixture Model (FBBMM), a novel probabilistic clustering model leveraging the flexibility of the bivariate beta distribution. Experimental results show that FBBMM outperforms popular clustering algorithms such as  $k$ -means, MeanShift, DBSCAN, Gaussian Mixture Models, and MBMM, particularly on nonconvex clusters. Its ability to handle a wide range of cluster shapes and correlations makes it highly effective.

FBBMM offers several advantages. Its use of the beta distribution allows for flexible cluster shapes, capturing complex structures more accurately than traditional models. It supports soft clustering, assigning probabilities to data points for belonging to clusters, which is versatile for overlapping clusters. Additionally, FBBMM is generative, capable of producing new data resembling the original dataset, useful for tasks like data augmentation and simulation.

However, FBBMM has limitations, including higher computational complexity due to iterative parameter estimation. Future work could focus on improving efficiency through parallelization or better optimization strategies, extending FBBMM to multivariate data, and enhancing robustness to noise and outliers. Applying FBBMM in diverse domains such as bioinformatics and image analysis could further validate its versatility and impact.

## Acknowledgement

We acknowledge support from National Science and Technology Council of Taiwan under grant number 113-2221-E-008-100-MY3. We thank to National Center for High-performance Computing (NCHC) of National Applied Research Laboratories (NARLabs) in Taiwan for providing computational and storage resources.

## References

- [1] Ruei-Yuan Wang and Hung-Hsuan Chen. Detecting inactive cyberwarriors from online forums. In *2023 IEEE International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pages 9–15. IEEE, 2023.
- [2] Cheng-You Lien, Guo-Jhen Bai, and Hung-Hsuan Chen. Visited websites may reveal users’ demographic information and personality. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 248–252, 2019.
- [3] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.
- [4] Yi Yang, Dong Xu, Feiping Nie, Shuicheng Yan, and Yueting Zhuang. Image clustering using local discriminant models and global integration. *IEEE Transactions on Image Processing*, 19(10):2761–2773, 2010.
- [5] Yung-Peng Hsu and Hung-Hsuan Chen. Multivariate beta mixture model: Probabilistic clustering with flexible cluster shapes. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 233–245. Springer, 2024.
- [6] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.
- [7] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.
- [8] William HE Day and Herbert Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification*, 1(1):7–24, 1984.

- [9] Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.
- [10] Samuel Kotz, Narayanaswamy Balakrishnan, and Norman L Johnson. *Continuous multivariate distributions, Volume 1: Models and applications*, volume 334. John wiley & sons, 2019.
- [11] Ingram Olkin and Thomas A Trikalinos. Constructions for a bivariate beta distribution. *Statistics & Probability Letters*, 96:54–60, 2015.
- [12] Stefan Aeberhard, Danny Coomans, and Olivier De Vel. Comparative analysis of statistical pattern recognition methods in high dimensional settings. *Pattern Recognition*, 27(8):1065–1077, 1994.
- [13] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [14] Jo-Chun Chen and Hung-Hsuan Chen. Toward efficient and incremental spectral clustering via parametric spectral clustering. In *2023 IEEE International Conference on Big Data (BigData)*, pages 1070–1075. IEEE, 2023.
- [15] Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. Adjusting for chance clustering comparison measures. *Journal of Machine Learning Research*, 17(134):1–32, 2016.