A Generative Block-Diagonal Model for Clustering

Junxiang Chen
Dept. of Electrical & Computer Engineering
Northeastern University
Boston, MA 02115
jchen@ece.neu.edu

Jennifer Dy
Dept. of Electrical & Computer Engineering
Northeastern University
Boston, MA 02115
jdy@ece.neu.edu

Abstract

Probabilistic mixture models are among the most important clustering methods. These models assume that the feature vectors of the samples can be described by a mixture of several components. Each of these components follows a distribution of a certain form. In recent years, there has been an increasing amount of interest and work in similarity-matrix-based methods. Rather than considering the feature vectors, these methods learn patterns by observing the similarity matrix that describes the pairwise relative similarity between each pair of samples. However, there are limited works in probabilistic mixture model for clustering with input data in the form of a similarity matrix. Observing this, we propose a generative model for clustering that finds the block-diagonal structure of the similarity matrix to ensure that the samples within the same cluster (diagonal block) are similar while the samples from different clusters (off-diagonal block) are less similar. In this model, we assume the elements in the similarity matrix follow one of beta distributions, depending on whether the element belongs to one of the diagonal blocks or to off-diagonal blocks. The assignment of the element to a block is determined by the cluster indicators that follow categorical distributions. Experiments on both synthetic and real data show that the performance of the proposed method is comparable to the state-of-the-art methods.

1 INTRODUCTION

In many applications, we want to divide the data into a few groups. Clustering is a task of finding the structure and interesting patterns in data, by grouping objects in such a way that objects in the same group are similar to each other, but objects in different groups are different. Recently, much research has been focused on developing clustering techniques and on applying these techniques to different fields such as image segmentation and text mining [26].

Probabilistic mixture models [3, 8] are among the most important clustering methods. These models assume that the feature vectors of data can be described by a mixture of several components. Each of these components follows a distribution of a certain form. Although different probabilistic mixture models differ in the detailed assumptions, most of them try to fit the feature vectors with a mixture of distributions.

In recent years, there has been an increasing amount of interest and work in similarity-matrix-based methods. Rather than observing the feature vectors of the data, as existing probabilistic mixture models do, similarity-matrix-based methods learn patterns by observing the similarity matrix that describes the pairwise relative similarity between each pair of data samples. One example of these methods is spectral clustering [14, 20]. Similarity-matrix-based methods have been very successful in different applications, because it could be applied to data of any form, as long as we can compare the similarity between samples. However, there are limited works in generative models for clustering where the input is a similarity matrix.

In clustering problems, we want to ensure that the samples within the same cluster are similar while the samples from different clusters are less similar. Therefore, given a similarity matrix, if we sort the indices of the similarity matrix according to the cluster indicators, the elements in diagonal blocks usually have larger values, because these elements measure the similarity between samples in the same cluster; while the elements in the off-diagonal blocks usually have smaller values, because these elements measure the similarity between samples from different clusters. The block structure in a similarity matrix is illustrated in Figure 1. Observing this, we propose to cluster by finding the block-diagonal structure in the

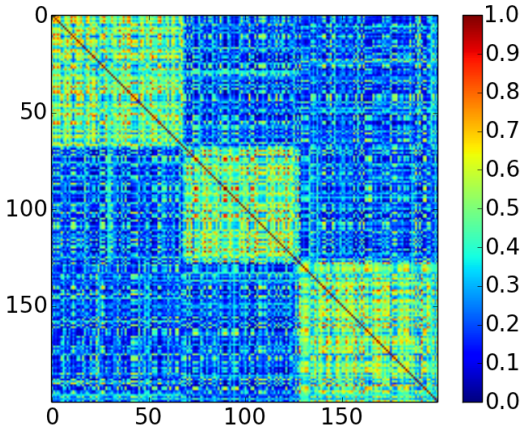


Figure 1: The block-diagonal structure in the similarity matrix. After we sort the indices of the similarity matrix according to the cluster indicators, elements in diagonal blocks have larger values, while elements in the off-diagonal blocks have smaller values.

similarity matrix.

Observing this, we propose a similarity-matrix-based probabilistic model for clustering, called *Block Mixture Model* (BMM). It is a generative model that discovers clusters by finding the block-diagonal structure in the similarity matrix. In BMM, we assume that the elements in the similarity matrix are drawn from a mixture of beta distributions, where the elements in the diagonal blocks are drawn from a beta distribution that is skewed towards one, and the elements in the off-diagonal blocks are drawn from a background beta distribution that is skewed towards zero. The assignment of each element to the blocks is determined by the cluster indicators for samples that follow categorical distributions. As a Bayesian method, BMM takes model uncertainty into consideration, allows us to provide an informative prior to the model, and safeguards against over-fitting [6].

Related Work Gaussian Mixture Model (GMM) [3] is the most well known generative model for clustering. GMM assumes data can be divided into several components and each component follows a Gaussian distribution. GMM usually has difficulty to cluster data with non-elliptical shape. To overcome this limitation, [8] proposes to warp a latent mixture of Gaussian distributions using Gaussian processes. This model can be applied when the clusters have more complex shapes. Although these generatives model methods differ in detailed assumptions, they all fit the real-valued feature vectors with a mixture of distributions. On the other hand, as a similarity-matrix-based method, BMM takes the similarity matrix as input. Therefore, it can be used to analyse any data types as long as the similarity between samples can be measured.

Spectral clustering [14, 20, 25, 21, 18] is a similarity-matrix-based clustering method. With this method, we first compute the eigenvectors of the Laplacian matrix that is derived from the similarity matrix. Then the clustering results are obtained by applying k-means [14, 20], or a probabilistic model [25, 21, 18] to analyse these eigenvectors. Unlike these methods, BMM is a generative model for the similarity matrix, which does not make use of the eigenvectors.

Stochastic blockmodels [24, 1, 10] are also generative models that find clusters. In these models, each sample belongs to a cluster and the connections between samples are determined by the corresponding pair of clusters. These methods are usually applied to an observed network, rather than similarity measures.

A generative clustering model for similarity matrices is proposed in [19]. It is assumed that the observed network is a noisy version of a latent network, where the latent network can be divided into several connected sub-networks. In contrast, BMM adopts a different strategy, where we try to find a block-diagonal structure in the similarity matrix. BMM tends to lead to better clustering results as demonstrated in the experiments (see Section 4).

The model proposed in [23] finds a block structure in Gaussian Graphical Models. Another related model is the orthogonal nonnegative matrix tri-factorization [4] that factorizes an observed matrix into 3 factors, which is equivalent to completing a non-negative matrix using a block structure. Both methods differ from BMM in that they are not probabilistic models and they are applied to the feature vectors directly but not on the similarity matrices.

Contributions of this work The contributions of this work can be summarized as follows:

1. We propose a new generative model for similarity-matrix-based clustering, we call *Block Mixture Model* (BMM), that searches for the diagonal-block structure in a similarity matrix.
2. We derive variational inference for BMM.
3. We test BMM on both synthetic and real data, and observe that the performance of BMM is comparable to the state-of-the-art methods.

2 MODEL FORMULATION

We propose a generative clustering model for the similarity matrix with a block-diagonal structure, we call *Block Mixture Model (BMM)*, to solve a clustering problem. To begin with, we construct a similarity matrix. Given a set of N data samples with C dimension $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, where $\boldsymbol{x}_n \in \mathbb{R}^C$ for $n = 1, \ldots, N$. One possible way to construct a similarity matrix is to use the Gaussian kernel

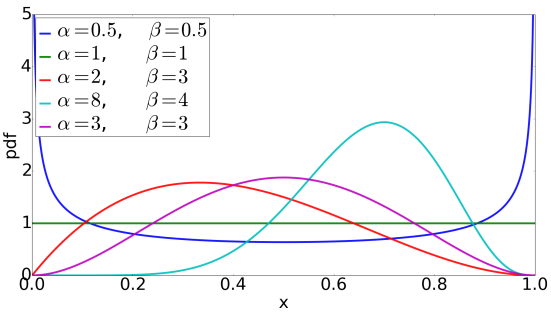


Figure 2: Beta distributions with different parameters

$\boldsymbol{W} \in \mathbb{R}^{N\times N}$ is defined as

$$\boldsymbol{W}_{ij} = \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}),\ for\ i,j = 1,\ldots,N, \quad (1)$$

where σ is a positive real bandwidth parameter. In this paper, we focus our discussion on Gaussian kernel; but BMM can be applied to other similarity measures whose values range from 0 to 1 (e.g., cosine similarity).

In BMM, we assume that the data can be divided into K clusters, where K is a pre-defined integer. BMM can be easily extended to a nonparametric version, with Dirichlet process mixtures [7], such that the model can automatically find K. Since this is not the major focus of this paper, we introduce a simpler, more accessible version where K is pre-defined. We assign each sample $\boldsymbol{x}_n$ a K-element cluster indicator $\boldsymbol{z}_n = \{z_{nk}\}_{k=1}^K$ such that $z_{nk} = 1$ if and only if $\boldsymbol{x}_n$ belongs to the k-th cluster, and otherwise $z_{nk} = 0$. We let $\boldsymbol{z}_n$ follow a categorical distribution such that

$$\boldsymbol{z}_n \sim Categorical(\boldsymbol{\pi}), \quad (2)$$

where $\boldsymbol{\pi}$ is a K-element vector, representing the probability that each cluster is assigned. We let $\boldsymbol{\pi}$ be a sample from a symmetric Dirichlet distribution, with concentration parameter λ, i.e.

$$\boldsymbol{\pi} \sim Dirichlet(\lambda). \quad (3)$$

Note that in Gaussian kernel $\boldsymbol{W}$, all elements satisfying $0 < \boldsymbol{W}_{ij} \leq 1$, where a large $\boldsymbol{W}_{ij}$ indicates that the i-th and j-th samples are similar. Because of the range of $\boldsymbol{W}_{ij}$, we model it using beta distributions, which are distributions defined on the interval $(0, 1)$, parameterized by two positive shape parameters α and β. We choose beta distribution because it is a simple and flexible distribution that describes random variables between 0 and 1. We plot some probability density function (PDF) of beta distributions with different parameters in Figure 2.

If a random variable t follows a beta distribution such that $t \sim Beta(\alpha, \beta)$, then its expected value and variance are given by [9]

$$\mathbb{E}[x] = \frac{\alpha}{\alpha+\beta}, \quad (4)$$

$$\mathbf{Var}[x] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} = \frac{\mathbb{E}[x](1-\mathbb{E}[x])}{\alpha+\beta+1}. \quad (5)$$

Now we assume that the similarity matrix $\mathbf{W}$ can be separated into K clusters. Then if we sort the indices of the similarity matrix according to the cluster indicators, we can observe a block-diagonal structure as shown in Figure 1. If $\boldsymbol{W}_{ij}$ is in one of the diagonal blocks, then it tends to have a large value. In this case, we let $\boldsymbol{W}_{ij}$ be a sample from a beta distribution that is parameterized by $\boldsymbol{\Theta}_k = (\alpha_k, \beta_k)$, such that it is skewed towards one. We assign a different parameter $\boldsymbol{\Theta}_k$ for each diagonal block, because in some clusters, the within cluster similarity might be larger than others. If $\boldsymbol{W}_{ij}$ is in off-diagonal blocks, then we let $\boldsymbol{\Theta}_0 = (\alpha_0, \beta_0)$ be the parameters for the beta distribution, such that it is close to zero. Since whether the i-th and j-th elements are in the diagonal or off-diagonal blocks can be derived by observing the cluster indicators $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ respectively, the probability density function of $\boldsymbol{W}_{ij}$ can be expressed as

$$p(\boldsymbol{W}_{ij}|\{\boldsymbol{\Theta_k}\}_{k=1}^K, \boldsymbol{\Theta}_0, \boldsymbol{Z}) = Beta(\boldsymbol{W}_{ij}|\alpha_0, \beta_0)^{1-\sum_k^K z_{ik}z_{jk}} \prod_{k=1}^K Beta(\boldsymbol{W}_{ij}|\alpha_k, \beta_k)^{z_{ik}z_{jk}}. \quad (6)$$

Note that if both samples $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same cluster k, then $z_{ik}z_{jk} = 1$ and W_{ij} belongs to the k-th diagonal block. Therefore, in the equation, the term $z_{ik}z_{jk}$ is an indicator that $\boldsymbol{W}_{ij}$ is in the k-th diagonal block and the term $1 - \sum_k^K z_{ik}z_{jk}$ is an indicator that the $\boldsymbol{W}_{ij}$ is in the off-diagonal blocks. With the equation, we let elements in the diagonal blocks and off-diagonal blocks follow the corresponding beta distributions, respectively. Note that, we do not care about the diagonal elements in the similarity matrix $\{\boldsymbol{W}_{ii}\}_{i=1}^N$, because these elements do not contain clustering information. Because $\boldsymbol{W}$ is a symmetric matrix; in the generative process, we only need to generate the upper triangle of this matrix.

If the data contain clustering structure, the elements in the diagonal blocks should have larger values than the off-diagonal blocks. Therefore, we assign different prior distributions to the beta distribution parameters $\{\boldsymbol{\Theta_k}\}_{k=1}^K$ and $\boldsymbol{\Theta}_0$. We let

$$p(\boldsymbol{\Theta}_k|\boldsymbol{\zeta}) \propto Beta(\frac{\alpha_k}{\alpha_k+\beta_k}|\alpha_\zeta, \beta_\zeta) Lognormal(\alpha_k+\beta_k|\mu_\zeta, \sigma_\zeta^2), \quad (7)$$

$$p(\boldsymbol{\Theta}_0|\boldsymbol{\eta}) \propto Beta(\frac{\alpha_0}{\alpha_0+\beta_0}|\alpha_\eta, \beta_\eta) Lognormal(\alpha_0+\beta_0|\mu_\eta, \sigma_\eta^2), \quad (8)$$

where $\boldsymbol{\zeta} = \{\mu_\zeta, \sigma_\zeta^2, \alpha_\zeta, \beta_\zeta\}$ and $\boldsymbol{\eta} = \{\mu_\eta, \sigma_\eta^2, \alpha_\eta, \beta_\eta\}$ are the hyper-parameters for $\{\boldsymbol{\Theta_k}\}_{k=1}^K$ and $\boldsymbol{\Theta}_0$ respectively. The expected value of a beta distribution, as described in Equation (4), has a value between 0 and 1. Therefore, we use another beta distribution to model this expected value, which is represented by the first factor in Equations (7) and (8). As shown in Equation (5) that given its expected value, the variance of the beta distribution is inversely proportional to the value of $\alpha + \beta + 1$; therefore, we let $\alpha + \beta$ follow a log-normal distribution, which is

Algorithm 1 Generative Process

for $k \leftarrow 1$ to K **do**
  Generate $\boldsymbol{\Theta}_k$ according to Equation (7)
end for
Generate $\boldsymbol{\Theta}_0$ according to Equation (8)
Generate $\boldsymbol{\pi}$ according to Equation (3)
for $n \leftarrow 1$ to N **do**
  Generate $\boldsymbol{z}_n$ according to Equation (2)
end for
for $i \leftarrow 1$ to N **do**
  for $j \leftarrow 1$ to $i-1$ **do**
    Generate $\boldsymbol{W}_{ij}$ according to Equation (6)
    $\boldsymbol{W}_{ji} \leftarrow \boldsymbol{W}_{ij}$
  end for
end for

represented by the second factor in Equations (7) and (8). Since we multiply two distributions to form the prior distributions in Equations (7) and (8), we need to introduce a normalization constant to make sure the integral of the new pdf over the entire space is equal to one. With the prior distributions in Equations (7) and (8), we can control the expected value and the variance of a beta distribution by adjusting the hyper-parameters $\boldsymbol{\zeta}$ and $\boldsymbol{\eta}$.

We assign the hyper-parameters $\boldsymbol{\zeta}$ and $\boldsymbol{\eta}$ to make sure that the similarity matrix demonstrates a diagonal-block structure as shown in Figure 1. We want the diagonal blocks to be relatively dense and distributed with smaller variance. Therefore, we let the value μ_ζ to be relatively large, and α_ζ larger than β_ζ. In practice, we let $\mu_\zeta = 15$, $\sigma_\zeta^2 = 1$, $\alpha_\zeta = 80,000$ and $\beta_\zeta = 20,000$. We also want to make sure the off-diagonal blocks are relatively sparse and distributed with larger variance. Therefore, we let the value μ_η to be relatively small and α_η smaller than β_η. In practice, we let $\mu_\eta = 0$, $\sigma_\eta^2 = 1$, $\alpha_\eta = 1,000$ and $\beta_\eta = 9,000$. The prior seems relatively strong. However, note that the number of observations for each beta distribution is proportional to N^2. Therefore, the posterior distributions may still be very different from the prior distributions because of the large number of observations. We analyse the sensitivity of these hyper-parameters in Section 4.3

The model is described using a directed graphical model in Figure 3. The generative process of BMM is summarized in Algorithm 1. In the generative process, due to symmetry of the similarity matrix, we only generate the upper triangular elements, $\boldsymbol{W}_{ij}$, $i \in 1, \ldots, N$ and $j \in 1, \ldots, i-1$.

3 INFERENCE

In this section, we introduce how the latent variables in BMM model can be learned via variational inference. The joint probability of the model is given by

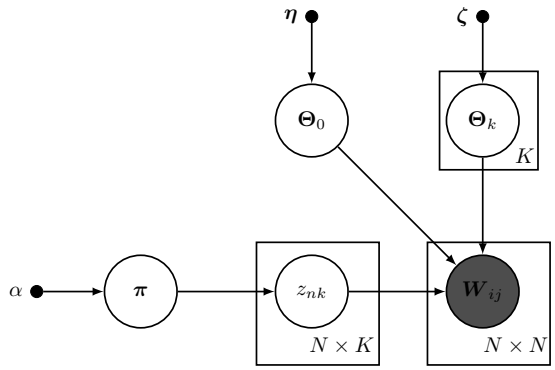


Figure 3: The graphical model. The dots represent the hyper-parameters. The regular circle represent latent random variables. The shaded circles represent observed random variables. The arrows represent the dependency between hyper-parameters and random variables. Each plate denotes that the structure inside the plate is repeated.

$$\begin{aligned} &p(\boldsymbol{W}, \boldsymbol{\pi}, \boldsymbol{Z}, \{\boldsymbol{\Theta}_k\}_{k=1}^K, \boldsymbol{\Theta}_0 | \boldsymbol{\zeta}, \boldsymbol{\eta}, \lambda) \\ =&p(\boldsymbol{\Theta}_0|\boldsymbol{\eta}) \prod_{k=1}^K p(\boldsymbol{\Theta}_k|\boldsymbol{\zeta}) p(\boldsymbol{\pi}|\lambda) \prod_{n=1}^N p(\boldsymbol{z}_n|\boldsymbol{\pi}) \\ &\prod_{i=1}^N \prod_{j=1}^{i-1} p(\boldsymbol{W}_{ij} | \{\boldsymbol{\Theta}_{\boldsymbol{k}}\}_{k=1}^K, \boldsymbol{\Theta}_0, \boldsymbol{Z}) \end{aligned} \tag{9}$$

We want to calculate the posterior distribution for the latent variables given the observed similarity matrix and the hyper-parameters, i.e. $p(\boldsymbol{\pi}, \boldsymbol{Z}, \{\boldsymbol{\Theta}_k\}_{k=1}^K, \boldsymbol{\Theta}_0 | \boldsymbol{W}, \boldsymbol{\zeta}, \boldsymbol{\eta}, \lambda)$. It is computationally intractable to directly calculate this posterior distribution. Therefore, we use a variational distribution $q(\boldsymbol{\pi}, \boldsymbol{Z}, \{\boldsymbol{\Theta}_k\}_{k=1}^K, \boldsymbol{\Theta}_0)$ to approximate the posterior distribution by minimizing the KL divergence $KL(q||p)$ [2]. As proven in [2], this is equivalent to maximizing a lower-bound $\mathcal{L}(q)$ that is defined as

$$\mathcal{L}(q) = \mathbb{E}_q[\log p(\boldsymbol{W}, \boldsymbol{\pi}, \boldsymbol{Z}, \{\boldsymbol{\Theta}_k\}_{k=1}^K, \boldsymbol{\Theta}_0) | \boldsymbol{\zeta}, \boldsymbol{\eta}, \lambda] + \mathcal{H}(q) \tag{10}$$

where $\mathbb{E}_q$ denotes that the expected value is taken with respect to the variational distribution q, and $\mathcal{H}(q)$ denotes the entropy of this variational distribution.

It is still impossible to directly calculate the variational distribution q. Therefore, we further assume that this distribution q can be factorized such that

$$q(\boldsymbol{\pi}, \boldsymbol{Z}, \{\boldsymbol{\Theta}_k\}_{k=1}^K, \boldsymbol{\Theta}_0) = q_{\boldsymbol{\pi}}(\boldsymbol{\pi}) \prod_{n=1}^N q_{\boldsymbol{z}_n}(\boldsymbol{z}_n) \prod_{k=1}^K q_{\boldsymbol{\Theta}_{\boldsymbol{k}}}(\boldsymbol{\Theta}_{\boldsymbol{k}}) q_{\boldsymbol{\Theta}_0}(\boldsymbol{\Theta}_0) \tag{11}$$

Because we do not use the conjugate prior distributions as the prior for the latent variables $\{\boldsymbol{\Theta}_{\boldsymbol{k}}\}_{k=1}^K$ and $\boldsymbol{\Theta}_0$, we cannot estimate the distributions $q_{\boldsymbol{\Theta}_{\boldsymbol{k}}}(\boldsymbol{\Theta}_{\boldsymbol{k}})$ and $q_{\boldsymbol{\Theta}_0}(\boldsymbol{\Theta}_0)$ in closed form. However, note that given the expected values $\mathbb{E}_{\boldsymbol{Z}}(\boldsymbol{Z})$, the distributions $q_{\boldsymbol{\Theta}_{\boldsymbol{k}}}(\boldsymbol{\Theta}_{\boldsymbol{k}})$ and $q_{\boldsymbol{\Theta}_0}(\boldsymbol{\Theta}_0)$ are independent in the lower-bound $\mathcal{L}(q)$ described in

Equation (10). Therefore, we can find point estimators for $\{\widehat{\boldsymbol{\Theta}}_k\}_{k=1}^K$ and $\widehat{\boldsymbol{\Theta}}_0$ that maximizes $\mathcal{L}(q)$, such that

$$\widehat{\boldsymbol{\Theta}}_k = \underset{(\alpha_k,\beta_k)}{\operatorname{argmax}}\, \mathcal{L}(q) \tag{12}$$

$$\widehat{\boldsymbol{\Theta}}_0 = \underset{(\alpha_0,\beta_0)}{\operatorname{argmax}}\, \mathcal{L}(q) \tag{13}$$

We find these point estimators using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [17].

Given these point estimators, we calculate the optimal variational distributions $q^*_{\boldsymbol{\pi}}$ and $\{q^*_{\boldsymbol{z}_n}\}_{n=1}^N$. According to [2], with the factorization assumption introduced in Equation (11), the optimal factorized variational distribution $q^*_{\boldsymbol{Y}_j}(\boldsymbol{Y}_j)$ is given by

$$\log q^*_{\boldsymbol{Y}_j}(\boldsymbol{Y}_j) = \mathbb{E}_{i\neq j}[\log p(\boldsymbol{X},\boldsymbol{Y})] + const \tag{14}$$

where $\boldsymbol{X}$ represents the observed data, $\boldsymbol{Y} = \{\boldsymbol{Y}_i\}_{i=1}^M$ represents all M factorized latent variables and $\mathbb{E}_{i\neq j}$ represents that the expected value is taken with respect to $\{q_{\boldsymbol{Y}_i}\}_{i\neq j}$.

By applying Equation (14), the optimal variational distribution $\{q^*_{\boldsymbol{z}_n}\}_{n=1}^N$ is given by

$$\begin{aligned}\log q^*_{\boldsymbol{z}_n}(\boldsymbol{z}_n) = \sum_{k=1}^K z_{nk}\{&\mathbb{E}_{\boldsymbol{\pi}}[\log \boldsymbol{\pi}_k] \\ &+ \sum_{i\neq n}\mathbb{E}_{\boldsymbol{z}_i}[z_{ik}][\log \boldsymbol{B}(\hat{\alpha}_0,\hat{\beta}_0) - \log \boldsymbol{B}(\hat{\alpha}_k,\hat{\beta}_k) \\ &+(\hat{\alpha}_k-\hat{\alpha}_0)\log W_{in} + (\hat{\beta}_k-\hat{\beta}_0)\log(1-W_{in})]\} + const\end{aligned} \tag{15}$$

where $\boldsymbol{B}$ represents the beta function. By observing this equation, we conclude that

$$q^*_{\boldsymbol{z}_n}(\boldsymbol{z}_n) = Categorical(\boldsymbol{\omega}_n) \tag{16}$$

where $\boldsymbol{\omega}_n$ is a K-element vector such that

$$\begin{aligned}\boldsymbol{\omega}_{nk} \propto exp(&\mathbb{E}_{\boldsymbol{\pi}}[\log \boldsymbol{\pi}_k] + \sum_{i\neq n}\mathbb{E}_{\boldsymbol{z}_i}[z_{ik}]\{\log \boldsymbol{B}(\hat{\alpha}_0,\hat{\beta}_0) - \log \boldsymbol{B}(\hat{\alpha}_k,\hat{\beta}_k) \\ &+ (\hat{\alpha}_k-\hat{\alpha}_0)\log W_{in} + (\hat{\beta}_k-\hat{\beta}_0)\log(1-W_{in})\})\end{aligned} \tag{17}$$

and $\boldsymbol{\omega}_n$ is normalized such that $\sum_{k=1}^K \boldsymbol{\omega}_{nk} = 1$.

By applying Equation (14), the optimal variational distribution $q^*_{\boldsymbol{\pi}}$ is given by

$$\log q^*_{\boldsymbol{\pi}}(\boldsymbol{\pi}) = \sum_{k=1}^K \left(\lambda + \sum_{n=1}^N \mathbb{E}_{\boldsymbol{z}_n}[z_{nk}] - 1\right)\log \boldsymbol{\pi}_k + const \tag{18}$$

By observing this equation, we note that

$$q^*_{\boldsymbol{\pi}}(\boldsymbol{\pi}) = Dirichlet(\boldsymbol{\phi}) \tag{19}$$

where $\boldsymbol{\phi}$ is a K-element vector, whose k-th element is given by

$$\boldsymbol{\phi}_k = \lambda + \sum_{n=1}^N \mathbb{E}_{\boldsymbol{z}_n}[z_{nk}] \tag{20}$$

Algorithm 2 Variational Inference

```
Initialize {q*_{z_n}}_{n=1}^N
Initialize q*_π
repeat
  for k ← 1 to K do
    Calculate Θ̂_k according to Equation (12)
  end for
  Calculate Θ̂_0 according to Equation (13)
  for n ← 1 to N do
    Update q*_{z_n} according to Equation (16)
  end for
  Update q_{π*} according to Equation (19)
until Convergence
```

We iteratively update $\{\widehat{\boldsymbol{\Theta}}_k\}_{k=1}^K$, $\widehat{\boldsymbol{\Theta}}_0$, $\{q_{\boldsymbol{z}_n}(\boldsymbol{z}_n)^*\}_{n=1}^N$ and $q^*_{\boldsymbol{\pi}}(\boldsymbol{\pi})$ until convergence. The expected values involved in the updates are obtained by

$$\mathbb{E}_{\boldsymbol{z}_n}[z_{nk}] = \boldsymbol{\omega}_{nk} \tag{21}$$

$$\mathbb{E}_{\boldsymbol{\pi}}[\log \boldsymbol{\pi}_k] = \psi(\phi_k) - \psi\left(\sum_{i=1}^K \phi_i\right) \tag{22}$$

where ψ is the digamma function that is defined as the logarithmic derivative of the gamma function. The algorithm is summarized in Algorithm 2.

4 EXPERIMENTS

In this section, we test BMM using both synthetic and real data. We choose the bandwidth parameter of the Gaussian kernel σ in Equation (1) to be the median of the pairwise Euclidean distances. If not specified otherwise, the parameters for BMM are given as $\mu_\zeta = 15$, $\sigma^2_\zeta = 1$, $\alpha_\zeta = 80,000$, $\beta_\zeta = 20,000$, $\mu_\eta = 0$, $\sigma^2_\eta = 1$, $\alpha_\eta = 1,000$, $\beta_\eta = 9,000$ and $\lambda = 1$. We discuss the choice of the parameters in more details in Section 4.3. Because variational inference can only guarantee finding local minima, we run the algorithm with 10 different initial values, and select the solution with the maximum lower-bound value. We generate 5 of the initial values using random initialization, and generate the other 5 of the initial values using spectral clustering. Note that spectral clustering might give different results, because k-means is applied after embedding, and k-means only guarantees local optima.

4.1 SYNTHETIC 2D DATA

To demonstrate that BMM works when the clusters have complex shape , we test BMM using some 2-dimensional synthetic data. The clustering results of BMM , with each cluster shown in different color, are illustrated in Figure 4. We can observe that BMM is able to separate all of these

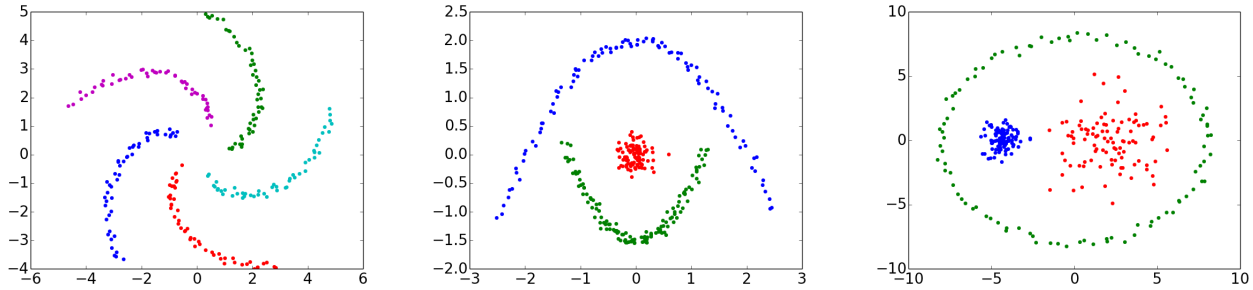


Figure 4: Clustering results for 2d synthetic data.

data perfectly. GMM fails to give similar results, because the cluster structure is complex shaped.

4.2 SYNTHETIC SIMILARITY MATRICES

In some applications, we are not directly given the feature vectors, but a similarity matrix. Similar to spectral clustering, BMM can also directly take a similarity matrix as an input. In this section, we test BMM using synthetic similarity matrices.

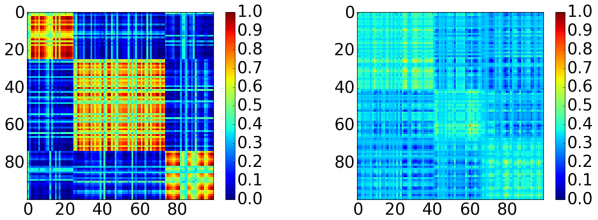
4.2.1 SIMILARITY MATRICES WITH A BLOCK-DIAGONAL STRUCTURE

To begin with, we test BMM using similarity matrices with different strength of block-diagonal structure. To generate a similarity matrix $\mathbf{W}$ with a block-diagonal structure, we let

$$\mathbf{W} = \mathbf{X}^T\mathbf{X}, \tag{23}$$

where $\boldsymbol{X}$ is a 100×3 matrix. Each row of $\boldsymbol{X}$ is a sample from a 3-element symmetric Dirichlet distribution with a positive concentration parameter α. We control the strength of the block-diagonal structure in the similarity matrix $\mathbf{W}$ by adjusting α. When α has a small value, the mass of the Dirichlet distribution tends to be concentrated in one of the three elements, and the similarly matrix has a strong block-diagonal structure, and vise versa. The ground-truth clustering label for each sample X_n is given by $L_n = \text{argmax}_{i=1,2,3} \boldsymbol{X}_{ni}$. To illustrate how α affects the block-diagonal structure, we plot $\boldsymbol{W}$ with different α in Figure 5, where indices of samples are sorted according to the ground-truth label $\mathbf{L} = \{L_n\}_{n=1}^{100}$. In the figure, we observe that when α is small (e.g., $\alpha = 0.25$), the block-diagonal structure is clear such that we can easily distinguish diagonal blocks from the off-diagonal blocks. However, when α is larger (e.g., $\alpha = 2$), the block structure is less clear.

We test BMM on $\mathbf{W}$ generated with different α between 0.06 to 4. For each α, we test BMM on 10 different random generated $\mathbf{W}$. We compare the clustering results of BMM against the results given by state-of-the-art methods that takes similarity matrices as input, including spectral



(a) $\alpha = 0.25$ (b) $\alpha = 2$

Figure 5: $\boldsymbol{W}$ with different α. The indices of samples are sorted according to the ground-truth labels L_n.

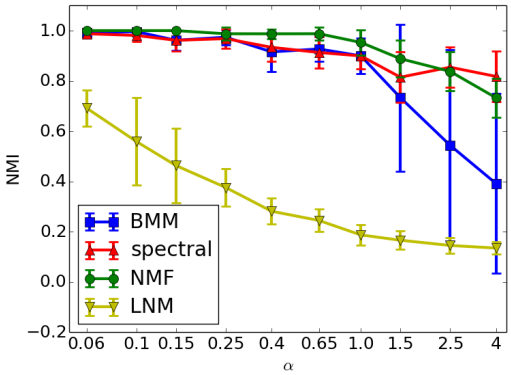


Figure 6: NMI on synthetic data $\mathbf{W}$ generated with different α. The line represents the mean value of the NMI, and the error bar demonstrates the standard deviation. The ticks on the horizontal axis are plotted with log scales.

clustering [14], Non-negative Matrix Factorization (NMF) [11] and Latent Network Model (LNM) [19]. We estimate the performance of the algorithms using Normalized Mutual Information (NMI) between the clustering results with respect to the labels $\mathbf{L}$. The NMI between two random variables X and Y is defined as [22]

$$\frac{\sum_{x\in X}\sum_{y\in Y} p(x,y)[\log p(x,y) - \log p(x)p(y)]}{\sqrt{\mathcal{H}(X)\mathcal{H}(Y)}} \tag{24}$$

where $\mathcal{H}(X)$ and $\mathcal{H}(Y)$ are the entropy for random variables X and Y respectively. The NMI ranges from 0 to 1, where a higher value indicates X and Y agree stronger with each other. We plot the mean values and standard

deviations of NMI for the clustering results with respect to the ground-truth label $\mathbf{L}$ in Figure 6. In this figure the lines represent the mean values of NMI, and the error bars denote the standard deviations.

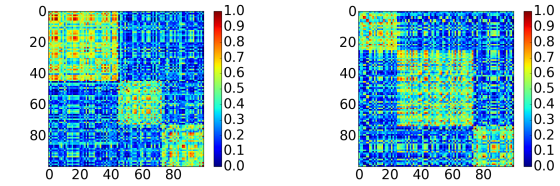
We can observe from the figure that NMF and spectral clustering outperform all other methods in this test. This is expected because $\mathbf{W}$ is generated based on non-negative matrix multiplication, which is consistent with the NMF assumption; and it is proved in [5] that there spectral clustering can be regarded as a relaxed version of NMF. We also observe from the figure that when $\alpha \leq 1$, the BMM results are comparable to the spectral clustering results. However, if $\alpha > 1$, the performance of BMM is worse. This is also expected since BMM gives clustering results based on the block-diagonal structure. When the similarity matrix contains a stronger block-diagonal structure, the performance of BMM is better. Note that in practice, we generate similarity matrices using Gaussian kernels that is described in Equation (1), with a bandwidth parameter σ set as the median value of the pairwise Euclidean distances. Therefore, the similarity matrix will be more similar to Figure 5(a) rather than Figure 5(b), because the pairwise similarity measures computed in this way usually differ significantly. LNM usually performs worse, because it only ensures samples in each cluster are well connected to their nearest neighbors respectively. LNM is more sensitive to the non-zero elements in the off-diagonal blocks.

4.2.2 SIMILARITY MATRICES WITH STRUCTURED NOISE

Now we consider the case when the similarity matrices contain structured noise. We generate two similarity matrices $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ in the same way as described in Equation (23), with $\alpha = 0.25$ such that they contains clear block-diagonal structure. We denote the ground-truth labels samples represented by $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ using $\mathbf{L}^{(1)}$ and $\mathbf{L}^{(2)}$, respectively.

Then we generate a similarity matrix $\boldsymbol{T} = \rho\, \boldsymbol{W}^{(1)} + (1 - \rho)\, \boldsymbol{W}^{(2)}$, where ρ is a real-value parameter between 0 and 1. By taking the weighted sum of matrices $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$, we introduce structured noise to the similarity matrix. After summation, $\boldsymbol{T}$ simultaneously have two block-diagonal structures, this is illustrated in Figures 7(a) and 7(b). Note that both figures show the same matrix $\boldsymbol{T}$, but we sort it according to the block-diagonal structure of $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ respectively. Multiple possible block-diagonal structures indicate that there are more than one meaningful way to separate the data into clusters, which are common in real applications, because objects might be divided into groups by different criteria, or they can be interpreted in different ways [16, 15].

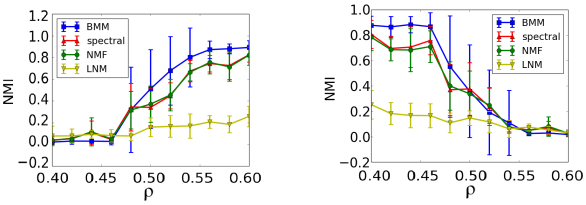
We vary the values of ρ between 0.4 to 0.6. For each ρ we generate 10 different matrices $\boldsymbol{W}$ and test BMM, spectral clustering, NMF and LNM using these matrices. The results are summarized in Figure 8.



(a) Plot of $\boldsymbol{T}$, with indices sorted according to the labels $\mathbf{L}^{(1)}$.

(b) Plot of $\boldsymbol{T}$, with indices sorted according to the labels $\mathbf{L}^{(2)}$.

Figure 7: $\boldsymbol{T}$ is constructed such that $\boldsymbol{T} = \rho \boldsymbol{W}^{(1)} + (1 - \rho)\boldsymbol{W}^{(2)}$, with $\rho = 0.45$. Both $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ contain a block-diagonal structure.



(a) NMI between the clustering results and the labels $\mathbf{L}^{(1)}$

(b) NMI between the clustering results and the labels $\mathbf{L}^{(2)}$

Figure 8: NMI on synthetic data. The line represents the mean value of the NMI, and the error bar demonstrates the standard deviation.

In Figure 8, we observe that when $0.4 < \rho < 0.45$, the block-diagonal structure of $\boldsymbol{W}^{(2)}$ dominates the matrix $\boldsymbol{T}$, and we can consider $\mathbf{L}^{(2)}$ as the ground truth. As shown in Figure 8(b), BMM outperforms all other methods, since its results have a higher NMI with respect to $\mathbf{L}^{(2)}$. When $0.45 < \rho < 0.55$, the contribution of $\boldsymbol{W}^{(1)}$ or $\boldsymbol{W}^{(2)}$ becomes similar. We observed that BMM usually has a higher mean NMI value with respect to both $\mathbf{L}^{(1)}$ and $\mathbf{L}^{(2)}$ compared to other methods. In addition, BMM has a higher standard deviation. This indicates that BMM tends to reveal the block-diagonal structure of either $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$, but other methods usually find neither of them. When $\rho > 0.55$, the block-diagonal structure of $\boldsymbol{W}^{(1)}$ dominates the matrix $\boldsymbol{W}$, and we can consider $\mathbf{L}^{(1)}$ as the ground truth. As shown in 8(a), BMM also outperforms spectral clustering, since its results have a higher mean NMI with respect to $\mathbf{L}^{(1)}$.

From the observation above, we conclude that BMM outperforms other methods if such structured noise is present. This is because spectral clustering finds the clusters by observing the eigenvectors of the Laplacian matrix that is derived from the similarity matrix. Spectral

clustering can find the correct clusters for $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ individually according to these eigenvectors respectively. However, when we take the weighted sum of $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$, the eigenvectors will change in general, and therefore, the clustering results given by spectral clustering are different from either $\mathbf{L}^{(1)}$ or $\mathbf{L}^{(2)}$. Due to the equivalence between NMF and spectral clustering [5], NMF performs similarly compared to spectral clustering. BMM avoids making use of the eigenvectors and looks for the strongest block-diagonal structure. It is more robust against the structured noise and is able to find the clusters similar to either $\mathbf{L}^{(1)}$ or $\mathbf{L}^{(2)}$.

In summary, in this section, we test BMM on synthetic similarity matrices. We observe that when the similarity matrix contains clear diagonal structure, BMM is comparable to spectral clustering. BMM is more robust to structured noise compared to spectral clustering and NMF.

4.3 HYPER-PARAMETER SENSITIVITY ANALYSIS

In this section, we discuss how the hyper-parameters $\mu_\zeta, \sigma_\zeta^2, \alpha_\zeta, \beta_\zeta, \mu_\eta, \sigma_\eta^2, \alpha_\eta$, and β_η affect the performance of BMM. We generate 10 synthetic random similarity matrices according to Equation (23) with $\alpha = 1$. We test BMM on each of the similarity matrices. In each test, we change one pair of the hyper-parameters at a time and keep all other hyper-parameters using the default values. We summarize the means of the NMI between the clustering results and the ground-truth label across the 10 similarity matrices, with each of the hyper-parameter settings, in Figure 9.

In Figures 9(a) and 9(b), we observe that the choices of $\mu_\zeta, \sigma_\zeta^2, \mu_\eta$ and σ_η^2 influence the clustering results less significantly. The mean NMI values are above 0.85, no matter what values are chosen. We set $\mu_\zeta = 15, \sigma_\zeta^2 = 1, \mu_\eta = 0$ and $\sigma_\eta^2 = 1$, since these values are consistent with the heuristic that the variance of the similarity measures in the diagonal blocks is smaller than that in the off-diagonal blocks. Note that they also provide high mean NMI.

From Figures 9(c) and 9(d), we can conclude that the values of $\alpha_\zeta, \beta_\zeta, \alpha_\eta$ and β_ζ have more effect on the clustering results. As mentioned in Section 2, we need to make sure the diagonal blocks are more dense than the off-diagonal blocks, i.e., $\alpha_\zeta/(\alpha_\zeta + \beta_\zeta) > \alpha_\eta/(\alpha_\eta + \beta_\eta)$. Therefore, we choose $\alpha_\zeta = 80,000$, $\beta_\zeta = 20,000$, $\alpha_\eta = 1,000$ and $\beta_\eta = 9,000$. We set hyper-parameters to these values in all other experiments in Section 4.

4.4 REAL DATA

In this section, we test BMM on several real dataset, and compare it with the state-of-the-art methods. Similar to spectral clustering, instead of just starting from the Gaussian kernel $\mathbf{W}$ defined in Equation (1), we also utilize the normalized similarly matrix that is defined as

$$\widetilde{\boldsymbol{W}} = \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}, \tag{25}$$

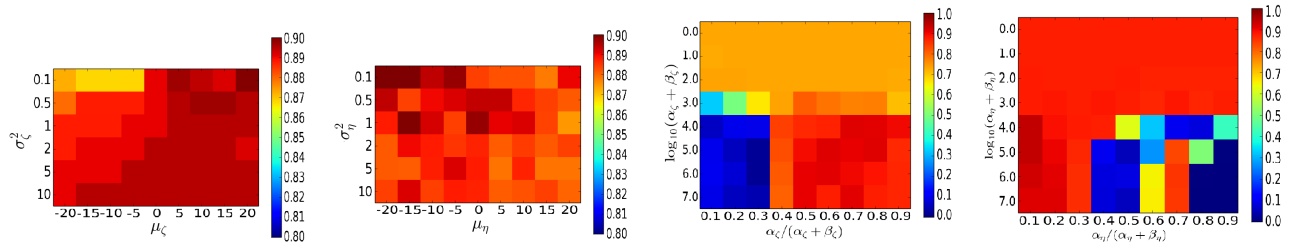
where $\boldsymbol{D}$ is a diagonal matrix such that $\boldsymbol{D}_{ii} = \sum_{j=1}^{N} \boldsymbol{W}_{ij}$. In this , we present the results using both un-normalized $\boldsymbol{W}$ and normalized $\widetilde{\boldsymbol{W}}$. In addition to the three similarity-matrix-based methods that are introduced in 4.2.1, we also compare BMM against k-means [13] and GMM [3].

First we introduce the experimental results on the Semeion handwritten digit dataset [12]. This dataset contains 1593 handwritten digits from 0 to 9 from 80 persons. The digits are stretched in a rectangular box 16x16 with $0/1$ values. We test the methods using different subsets of the dataset as different clustering tasks. In each task, we divide the dataset into 5 sets. We repeat the test 5 times, each time with one set taken out. The results are summarized in Table 1. The values in the table represent the means of the NMI. The values in the brackets represent the standard deviations. The values in bold is the largest mean NMI for each task.

We observe from Table 1 that BMM with the normalized similarity matrix $\widetilde{\boldsymbol{W}}$ is one of the best methods. In some tasks, this method outperforms all other methods by a relatively large margin. For example in the task of distinguishing 6 from 8, this outperforms the second best method by more than 0.1 in terms of mean NMI.

We also observe that making use of the normalized similarity matrix $\widetilde{\boldsymbol{W}}$ usually leads to better results, but in some tasks, such as the $\{0, 8\}$ and $\{4, 9\}$ tasks, utilizing the un-normalized similarity matrix $\boldsymbol{W}$ gives better results. However, in the $\{2, 3\}$ task, BMM with un-normalized similarity matrix $\boldsymbol{W}$ obtains a worse result compared to other methods. Although making the un-normalized similarity matrix might get better results in some of the tasks, we still recommend to use the normalized similarity matrix because its performance is better in general. In the $\{1, 7\}$ task, we observe that the performance of GMM is much better than all other methods. This might be due to that in the methods we compared, GMM is the only method that can scale the features. LNM usually performs worse, because it only ensures samples in each cluster are well connected to their nearest neighbors respectively, but do not guarantee that they are pairwise well connected. Note that BMM with normalized similarity matrix either performs comparably or outperforms spectral clustering and NMF in almost all tasks.

In addition to the Semeion data, we also compare BMM with the state-of-the-art methods using following UCI datasets [12]: iris dataset contains 150 samples from 3 classes of iris plants that are described using 4 features;



(a) μ_ζ vs. σ_ζ^2 (b) μ_η vs. σ_η^2 (c) α_ζ vs. β_ζ (d) α_η vs. β_η

Figure 9: Means NMI between the clustering results and the ground-truth label, with each hyper-parameter settings. We change a pair of hyper-parameters at a time.

Table 1: NMI on Semeion handwritten digit data

	BMM ($\widetilde{\boldsymbol{W}}$)	BMM ($\boldsymbol{W}$)	Spectral	K-means	GMM	NMF	LNM
$\{0, 8\}$	0.901(0.016)	**0.916(0.020)**	0.816 (0.017)	0.899(0.014)	0.820(0.028)	0.835(0.015)	0.191(0.025)
$\{1, 7\}$	0.177(0.015)	0.258(0.038)	0.176(0.012)	0.210(0.049)	**0.588(0.073)**	0.183(0.041)	0.118(0.013)
$\{2, 3\}$	**0.823(0.057)**	0.159(0.027)	0.531(0.033)	0.765(0.039)	0.708(0.042)	0.516(0.059)	0.095(0.031)
$\{4, 9\}$	0.734(0.053)	**0.792(0.058)**	0.728(0.054)	0.774(0.054)	0.719(0.050)	0.740(0.058)	0.104(0.020)
$\{6, 8\}$	**0.879(0.040)**	0.543(0.342)	0.617(0.019)	0.755(0.100)	0.688(0.033)	0.588(0.060)	0.155(0.018)
{0,1,2,3,4}	**0.740(0.005)**	0.610(0.009)	0.693(0.031)	0.690(0.018)	0.693(0.016)	0.606(0.047)	0.263(0.022)
{5,6,7,8,9}	**0.553(0.015)**	0.415(0.021)	0.542(0.018)	0.451(0.036)	0.419(0.022)	0.470(0.021)	0.195(0.047)
{0,1,2,3,4,5,6,7,8,9}	**0.522(0.037)**	0.501(0.032)	0.502(0.019)	0.497(0.051)	0.507(0.009)	0.512(0.046)	0.259(0.028)

Table 2: NMI on UCI data

	BMM ($\widetilde{\boldsymbol{W}}$)	BMM($\boldsymbol{W}$)	Spectral	K-means	GMM	NMF	LNM
Iris	0.631(0.035)	0.650(0.035)	0.624(0.020)	0.664(0.051)	**0.810(0.046)**	0.648(0.024)	0.350(0.025)
Synthetic Control	**0.781(0.012)**	0.739(0.013)	0.747(0.029)	0.696(0.012)	0.773(0.003)	0.686(0.030)	0.547(0.013)
Faults	**0.566(0.021)**	0.494(0.068)	0.493(0.070)	0.461(0.040)	0.494(0.081)	0.490(0.034)	0.336(0.035)
Wine	0.723(0.021)	**0.776(0.017)**	0.589(0.063)	0.707(0.032)	0.720(0.037)	0.690(0.043)	0.359(0.025)
CMU faces	0.834(0.083)	0.674(0.036)	**0.867(0.027)**	0.743(0.027)	0.852(0.019)	0.684(0.045)	0.462(0.077)

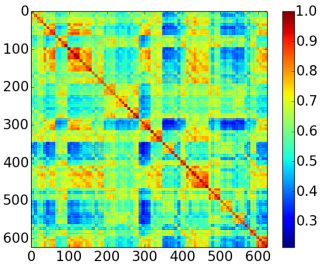


Figure 10: Similarity matrix for the CMU faces dataset.

synthetic control dataset contains 600 control charts that are synthetically generated from 6 classes; faults dataset contains $1,941$ samples from 7 types of steel plates faults; wine dataset contains chemical analysis results of 178 samples of wines that are derived from 3 different cultivars; CMU faces dataset consists of 640 face images of 20 people taken at varying poses.

The results are summarized in Table 2. In this table, we observe GMM performs well in the iris dataset because the iris clusters are elliptically shaped. BMM with normalized similarity $\widetilde{\boldsymbol{W}}$ is one of the best methods in most of the tasks. It outperforms all other methods in the synthetic control and faults datasets, while it has a comparable performance with most of the methods in iris and wine datasets. However, we observe that in the CMU faces dataset, BMM performs slightly worse than spectral clustering. To illustrate why BMM perform worse, we plot the similarity matrix for this dataset, with indices sorted using the ground-truth identity labels in Figure 10. We observe the elements in the off-diagonal blocks differ significantly in values. Note that, BMM uses only one background beta distribution to model all elements in off-diagonal blocks. The CMU faces dataset violates this assumptions of BMM, making BMM perform worse.

5 CONCLUSION

In this paper, we propose Block Mixture Model (BMM), a generative model for the similarity matrix with block-diagonal structure, to solve the clustering problem. In this model, we assume the elements in the similarity matrix follow one of beta distributions, depending on whether the element belongs to either one of the diagonal blocks or to the off-diagonal blocks. We derive variational inference to learn the latent variables in BMM. Experiments on synthetic data demonstrate that BMM performs at least comparably to spectral clustering if the similarity matrix contains a clear block-diagonal structure, and it is more robust to structured noise. We test BMM on real data and observe that the performance of BMM is comparable to the state-of-the-art methods.

ACKNOWLEDGEMENTS

This work was partially supported by NIH/NHLBI grants R01HL089856 & R01HL089857 and by NSF IIS-1546428.

References

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. In *Advances in Neural Information Processing Systems*, pages 33–40, 2009.

[2] C. M. Bishop et al. *Pattern recognition and machine learning*, volume 1, chapter 10 Approximate Inference, pages 461 – 474. Springer, New York, 2006.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[4] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM, 2006.

[5] C. H. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SDM*, volume 5, pages 606–610. SIAM, 2005.

[6] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*, chapter 3 Maximum Likelihood and Bayesian Parameter Estimation, pages 84 – 159. John Wiley & Sons, 2001.

[7] T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

[8] T. Iwata, D. Duvenaud, and Z. Ghahramani. Warped mixtures for nonparametric cluster shapes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI2013)*, 2013.

[9] N. L. Johnson, N. Balakrishnan, and S. I. Kotz. *Continuous Univariate Distributions*, volume 2. Wiley, 2nd edition, 1995.

[10] B. Karrer and M. E. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

[11] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562. MIT Press, 2001.

[12] M. Lichman. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, 2013.

[13] J. MacQueen. 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.

[14] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 2:849–856, 2002.

[15] D. Niu, J. G. Dy, M. Jordan, et al. Iterative discovery of multiple alternativeclustering views. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(7):1340–1353, 2014.

[16] D. Niu, J. G. Dy, and M. I. Jordan. Multiple non-redundant spectral clustering views. In *Proceedings of the 27th International Conference on Machine Learning (ICML2010)*, pages 831–838, 2010.

[17] J. Nocedal and S. J. Wright. *Numerical Optimization*, chapter 6 Quasi-Newton Methods, pages 135 – 162. Springer, New York, 2nd edition, 2006.

[18] L. K. Poon, A. H. Liu, T. Liu, and N. L. Zhang. A model-based approach to rounding in spectral clustering. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence, 2012*, pages 685–694.

[19] R. Rosales and B. Frey. Learning generative models of similarity matrices. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 485–492. Morgan Kaufmann Publishers Inc., 2002.

[20] J. Shi and J. Malik. Normalized cuts and image segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 731–737. IEEE, 1997.

[21] R. Socher, A. L. Maas, and C. D. Manning. Spectral chinese restaurant processes: Nonparametric clustering based on similarities. In *AISTATS*, pages 698–706, 2011.

[22] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[23] S. Sun, H. Wang, and J. Xu. Inferring block structure of graphical models in exponential families. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 939–947, 2015.

[24] Y. J. Wang and G. Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19, 1987.

[25] T. Xiang and S. Gong. Spectral clustering with eigenvector selection. *Pattern Recognition*, 41(3):1012–1029, 2008.

[26] R. Xu and D. Wunsch. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.