Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 187053, 9 pages
http://dx.doi.org/10.1155/2015/187053

Research Article

Bayesian-OverDBC: A Bayesian Density-Based Approach for Modeling Overlapping Clusters

Mansooreh Mirzaie,[1,2] Ahmad Barani,[2] Naser Nematbakkhsh,[2] and Majid Mohammad-Beigi[3]

[1]*Department of Computer Engineering, Golpayegan University of Technology, Isfahan 87717-65651, Iran*
[2]*Department of Software Engineering, Faculty of Computer Engineering, University of Isfahan, Isfahan 81746-73441, Iran*
[3]*Department of Bio-Medical Engineering, University of Isfahan, Isfahan 81746-73441, Iran*

Correspondence should be addressed to Mansooreh Mirzaie; mirzayi@comp.ui.ac.ir

Received 18 March 2015; Revised 14 June 2015; Accepted 21 October 2015

Academic Editor: Huaguang Zhang

Although most research in density-based clustering algorithms focused on finding distinct clusters, many real-world applications (such as gene functions in a gene regulatory network) have inherently overlapping clusters. Even with overlapping features, density-based clustering methods do not define a probabilistic model of data. Therefore, it is hard to determine how "good" clustering, predicting, and clustering new data into existing clusters are. Therefore, a probability model for overlap density-based clustering is a critical need for large data analysis. In this paper, a new Bayesian density-based method (Bayesian-OverDBC) for modeling the overlapping clusters is presented. Bayesian-OverDBC can predict the formation of a new cluster. It can also predict the overlapping of cluster with existing clusters. Bayesian-OverDBC has been compared with other algorithms (nonoverlapping and overlapping models). The results show that Bayesian-OverDBC can be significantly better than other methods in analyzing microarray data.

1. Introduction

Clustering, that is, finding similar groups of objects in a dataset, is an interesting technique especially for large data. Usually clustering algorithms assume that every object must belong to one and only one cluster (single-membership), but there are several real situations in which objects belong to more than one group (overlapping or multiple-membership). One of the applications of overlapping clustering is in bioinformatics. In biology, genes have more than one function carried out by coding proteins that participate in multiple metabolic pathways. Therefore, overlapping clustering could be useful in microarray data, which assigns gene expression data to multiple clusters simultaneously [1].

Density-based clustering can find clusters of different shapes so that they are useful in finding overlapped clusters. Furthermore, it is rather robust concerning outliers [2] and is very effective in clustering microarray data. These methods, even with the ability of finding overlapping clusters, do not use a probabilistic model. So, it is difficult to determine the probability of events and to compare an overlapping method with other methods. Therefore, a probability density-based clustering model, which provides overlapping, is required.

In this paper, the Bayesian-OverDBC algorithm is presented. This algorithm is a novel density-based clustering algorithm that has several advantages over traditional algorithms. It defines a probabilistic model of data which can be used to predict distribution of overlapping clusters. Bayesian hypothesis could be tested to determine which of the clusters is an overlapping cluster and which ones are merged or even discarded. Therefore, the algorithm may be interpreted as a Dirichlet Process Mixture (DPM) model.

Bayesian-OverDBC is based on OverDBC [3]. In OverDBC, initial cores (points with high density values) are formed based on density functions. Clusters are formed around the core objects and can be improve through local search. These steps are also taken in Bayesian-OverDBC. But in this algorithm, the decision to create, merge, or delete

overlapping clusters is made by using probabilistic models and Bayesian hypotheses. Similar work has been done by Heller and Ghahramani [4] for modeling overlapping clusters (IOMM). This method uses an exponential distribution to model each cluster and creates overlap clusters using the product of distributions.

Evaluation results show that the Bayesian-OverDBC algorithm could find overlapped clusters and works more effectively than DBSCAN (a nonoverlapping density-based clustering) and IBP (an overlap clustering model) in microarray data. Obviously, this method can be generalized to other datasets in different applications.

The main contributions of the paper can be summarized as follows:

(1) It introduces a density function to find probable core objects.

(2) It introduces a probabilistic Bayesian model for overlapping density-based algorithm. The traditional density-based algorithms do not define a probabilistic model of data, so comparison with other models is hard.

(3) It introduces new parameters which affect overlapping and the possibility of their occurrence.

The rest of the paper is organized as follows. At first, in Section 2, we give a brief overview of some of the clustering methods (overlapping and nonoverlapping methods). In Section 3, the concepts of density-based clustering methods are reviewed. Section 4 includes concepts of the new Bayesian model. Bayesian-OverDBC is described in Section 5. In Section 6, the results of the evaluation synthetic microarray-like datasets and real datasets and also a comparison with other methods are described.

2. Related Work

Different clustering methods are introduced in statistics, machine learning, and data mining. The idea of multiple-membership clustering has recently emerged as an important topic in some research areas. Multiple-membership clustering methods were divided into three categories [5]: Soft Models, Multiple-Membership Extensions to Hierarchical Agglomerative Clustering, and Similarity-Space Additive Clustering. In the following, these multiple-membership clustering techniques and their features are generally reviewed.

(1) Soft Models: soft model algorithms allow a point to be a partial member of some or all clusters. There are two primary methods for soft clustering: soft k-means [6] and SVD-like matrix decompositions [7].

(2) Multiple-Membership Extensions to Hierarchical Agglomerative Clustering (HAC): HAC is a simple clustering algorithm and has served as the starting point for several multi-membership clustering algorithms. "Jardine-Sibson B-clustering and Articulation Point Cuts" [8] and "Pyramid Hierarchical Clustering" [9] are a straightforward extension of single-link agglomerative clustering.

(3) Similarity-Space Additive Clustering: ADCLUS [10] is an additive method for modeling similarity matrices. ADCLUS provides a weight for each cluster which is convenient for interpretation and discards unimportant clusters.

In [11], a probabilistic model of a microarray dataset is proposed. This method (SBK) models each observed expression value as a sample drawn from a Gaussian sample. The mean is a sum of real-valued activations of the processes that a gene participates in. The problem then is to find M (binary membership matrix) and A (real-valued activity matrix) so as to maximize the joint probability $p(X; M; A)$, where X is the input data. This paper demonstrates the application of the algorithm on the yeast stress response dataset finding that the discovered overlapping clusters have much better performance (as determined by p value) than clusters discovered by other overlapping methods.

SBK uses the expectation-maximization method [12], so it has the existing problems in this area such as the local maximum. In addition, the algorithm needs to define convergence threshold. Determining the threshold value is highly sensitive to data and may directly affect the convergence or nonconvergence of the algorithm. Also, the algorithm requires an automatic startup process, so it requires an initial value for the cluster membership matrix. The initial value usually is the output of the k-means or hierarchical clustering algorithms. All of these algorithms, in initialization phase, increase time complexity and space requirements.

Cheng and Church in [13] give a biclustering (coclustering) algorithm for finding biclusters in microarray data. A bicluster is a submatrix (rows i and columns j) that minimizes some objectives such as MSR (mean square residue). In [14] a Bayesian biclustering method is introduced which is named BCC. It allows a mixed membership to row and column clusters. BCC uses separate Dirichlet priors over the mixed membership and assumes each observation to be generated by an exponential family distribution corresponding to its row and column clusters. Some advantages of BCC are the following: the ability to handle sparse collections, being usable to diverse data types for all exponential family distributions, and flexible Bayesian priors using Dirichlet distributions; none of [13] or [14] provides overlapping functionality for clusters.

In [15] a probabilistic nonparametric Bayesian model for finding multiple clusters is introduced. This model can discover several possible clustering solutions and the feature subset views that generated each cluster partitioning simultaneously. This model allows for not only learning the multiple clustering but also automatically learning the number of views and the number of clusters in each view.

This model and a similar model in [16] both assume that the features in each view are not overlapping. However, in many applications, some features may be shared among views. In other words, although the concept of multifeature clustering has been considered, the models are not able to find overlapping clusters.

A new nonparametric Bayesian method, the Infinite Overlapping Mixture Model (IOMM), for modeling overlapping clusters, is presented in [4]. The IOMM uses exponential family distributions to model each cluster and forms an overlapping mixture by taking products of such distributions. The IOMM allows an unbounded number of clusters, and assignments of points to (multiple) clusters are modeled using an Indian Buffet Process (IBP) [17].

IOMM is implemented using a sampling method with a high repetition rate which needs a large time. Moreover, IOMM sampling method accepts all samples; the convergence of the algorithm is not provable in some datasets. In the next section some details of traditional density-based clustering algorithms, like DBSCAN, are reviewed. It also describes some of OverDBC features, that is, a density-based algorithm able to find overlapping clusters.

3. Traditional Density-Based Clustering

The key idea of density-based clustering is that each object in a cluster defines the neighborhood of a given radius with at least a minimum number of objects. Density-based clustering discovers clusters of arbitrary shapes in spatial databases with noise. Here density can be defined as the number of points within a specified radius. Density-based clustering techniques include mainly three techniques: DBSCAN (Density-Based Spatial Clustering of Application with Noise) [18], OPTICS (Ordering Points to Identify the Clustering Structure) [19], and DENCLUE (Density Clustering) [20].

The method presented in this paper (and also in OverDBC) uses the concepts of DBSCAN for clustering. So, some of the features of this algorithm are described. To find a cluster, DBSCAN starts with an arbitrary point p and retrieves all points density-reachable from p. An object q is directly density-reachable from object p if q is within the ε-neighborhood of p and p is a core point. This procedure yields a cluster around the p. If p is a border point (points on the border of the cluster), no points are density-reachable from p and DBSCAN visits the next point of the database.

There are several limitations to the traditional DBSCAN algorithm. The algorithm provides no guide to choosing the "correct" number of clusters. The quality of DBSCAN depends on the distance measure used in the algorithm. It is often difficult to know which distance metric to choose, especially for special data such as images or sequences and also for high-dimensional data. DBSCAN is not entirely deterministic; border points that are reachable from more than one cluster can be part of either cluster. This situation does not arise often but it is not inevitable.

OverDBC was introduced in Figure 1. It is a density-based algorithm for finding overlapping clusters which is based on DBSCAN. OverDBC allows objects to have multimembership in a restricted number of clusters where the total number of clusters is unbounded. In [3] it is proved that OverDBC is significantly better than nonoverlapping clustering algorithm such as DBSCAN in microarray data.

Traditional density-based algorithms do not define a probabilistic model of the data, so it is hard to ask how "good" a clustering is. Also, it is hard to compare this traditional method to other models, make predictions, or even cluster new data into existing clusters. In the following sections, statistical inference is used to overcome these limitations in OverDBC.

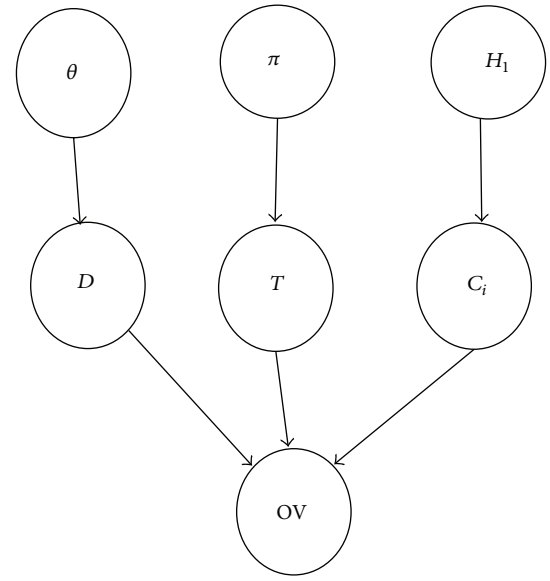

Figure 1: OverDBC algorithm.

4. Bayesian-OverDBC Model

In this section, Bayesian-OverDBC is presented. It defines a probabilistic model of data which can predict the distribution of overlapping clusters. This model obtains the probability of overlap between a new cluster with previous clusters. If the overlapping probability with previous clusters is low, local search is carried out and a new cluster is formed. But, if the overlapping probability is high, func_bound_over() will be invoked.

This function determines a lower bound on the number of shared objects of two clusters drown from a given dataset. It is defined based on double counting theory [21] and provides great improvement in overlap clustering. func_bound_over() compares the new cluster with all previous overlapping clusters. If there is a large number of overlap data points, these clusters are merged.

To get the overlap probability, effective parameters and variables must be specified. A Bayesian graphical model can clearly show these relationships. Definition of variables, parameters, and hyperparameters in a graphical model is discussed in inferential statistics in which the value of a latent variable can be inferred based on the values of other variables.

In this paper, the overlap among clusters is shown with a binary matrix ov including M rows and M columns. If cluster ith and cluster jth overlap, then $\text{ov}_{ij} = 1$. One of the effective factors on overlapping of the jth cluster is the dataset D which is under investigation to be used for the jth cluster formation. D has some parameters based on which data are distributed

(as shown with θ). Most of the data, especially microarray data, follow a normal distribution; so the parameters θ of D are the mean and the variance of data (in a vector form).

In addition to the data distribution in D, clusters that were created before the jth cluster can influence overlapping or nonoverlapping of cluster jth. If C_i is considered a symbol for each of the previous clusters, i can have a value between 1 and $j-1$. Hypothesis H_1^i is defined by C_i and it shows that the data in C_i are independent. Alternative hypothesis, $H'i$ indicates that the data in C_i are not independent and can be associated with two or more cores of clusters. The above idea is inspired by the assumption introduced in Heller and Ghahramani [22] for Bayesian hierarchical clustering.

Transaction matrix (T) is another effective variable on overlapping. T is a $N * M$ binary matrix (a pattern of 0's and 1's) showing the membership of points in clusters. The parameter π indicates the attraction probability for each of the core objects. π will influence the value of transaction matrix and is considered as $\pi = \{\pi_1, \pi_2, \ldots, \pi_m\}$, where m is the number of clusters. Each of the π_i shows the attraction of the ith core and consequently the presence probability of data objects in the ith cluster.

Based on the above parameters, a Bayesian graphical model for overlapping clusters is presented in Algorithm 1. This graphical model is a head-to-head Bayesian model [23], in which a node has multiple parents which are independent of each other.

This model shows the interaction among variables and parameters that affect overlap density-based clustering algorithms. As the graphic model Figure 1 shows that affected variables on overlapping D, C_i, and T are independent. By considering an occurrence of transaction matrix T, creation of clusters C_i, and dataset D, the probability of the overlap of jth cluster with the ith cluster ($\text{ov}_{ij} = 1$) is computed by

$$P\left(\text{ov}_{ij} = 1, T, D, C_i\right) = P\left(\text{ov}_{ij} = 1 \mid T, D, C_i\right) * P(T) * P(D) * P\left(C_i\right). \quad (1)$$

In Section 4.1, we will show how the finite mixture model concepts can be used to compute $P(T)$. The computation of $P(C_i)$, data distribution $P(D)$, and conditional probability $P(\text{ov}_{ij} = 1 \mid T, D, C_i)$ will be described in Section 4.2.

4.1. Probability of Transaction Matrix. The computation of $P(T)$ has been done based on the finite mixture model [24]. In finite mixture models we assume that there are M cores, each associated with a parameter π_k, the attraction value of core kth for all data points in D. According to the graphical model in Figure 1, $P(T)$ is computed by

$$P(T) = \sum_{k=1}^{M} P\left(T \mid \pi_k\right) * P\left(\pi_k\right). \quad (2)$$

In finite mixture model, N objects and M cores are defined. The fact that object i belongs to cluster k is indicated by a binary variable T_{ik}. Each object may belong to multiple clusters, so ith row of $T(T_i)$ does not have any restrictions. The T_{ij} ($i = 1, \ldots, N$, $j = 1, \ldots, M$), thus, forms a binary $N * M$ transaction matrix (T). We will assume that each object belongs to cluster k with probability of π_k; therefore, the clusters are generated independently. Under this model, given $\pi = \{\pi_1, \pi_2, \ldots, \pi_m\}$, the conditional probability of the matrix T with having π_k is computed by

$$P\left(T \mid \pi_k\right) = \prod_{i=1}^{N} P\left(T_{ik} \mid \pi_k\right) = \pi_k^{m_k}\left(1 - \pi_k\right)^{N-mk}, \quad (3)$$

where m_k is the number of data points that are in the neighborhood of core kth and have a radius of less than ε. In the following, computation methods for π_k and $P(\pi_k)$ will be described.

In order to find π_k, first we should compute distance between two points p and k ($d(p,k)$). $d(p,k)$ can be computed by Euclidean distance between p and kth core, or it may be a distance measure which is obtained from Pearson correlation coefficient. In this paper, the Pearson correlation coefficient is used.

Lower $d(p,k)$ indicates more correlation with the kth core and, hence, the greater density of the core object. The parameter σ is the computed standard deviation for all data points in the neighborhood region of kth core. A core density function of an object k is the impact of all the data points on its neighborhood. For each of the m_k points in the neighborhood of k, the density function is defined by the following [25]:

$$\text{Density}\left(p,k\right) = e^{-d(p,k)^2/2\sigma^2}. \quad (4)$$

So A_k (the attraction value of core kth for all data points in D) is the sum of the probability density functions for all points, which is specified in the following [25]:

$$A_k = \sum_{p=1}^{N} \text{Density}\left(p,k\right) = \sum_{p=1}^{N} e^{-d(p,k)^2/2\sigma^2}. \quad (5)$$

We define π_k as the event of the attraction by the kth core. So π_k is specified in the following:

$$\pi_k = \frac{A_k}{\sum_{j=1}^{M} A_j}. \quad (6)$$

If we assume that a prior on π_k follows a beta distribution with the parameters r and s and is conjugate to the binomial. The probability of any π_k under the Beta(r,s) distribution and the concept of Bayesian inference [24] is given by

$$P\left(\pi_k\right) = \frac{\pi_k^{r-1}\left(1 - \pi_k\right)^{s-1}}{B\left(r,s\right)}, \quad (7)$$

where $B(r,s)$ is the beta function and is computed by

$$B\left(r,s\right) = \int_0^1 \pi_k^{r-1}\left(1 - \pi_k\right)^{s-1} d\pi_k = \frac{\Gamma\left(r\right)\Gamma\left(s\right)}{\Gamma\left(r+s\right)}. \quad (8)$$

If in (8) $r = \alpha/M$ and $s = 1$ (α is the concentration parameter for each core density and M is the number of cores) are assumed then (7) is rewritten as

$$B\left(\frac{\alpha}{M}, 1\right) = \frac{\Gamma\left(\alpha/M\right)}{\Gamma\left(1 + \alpha/M\right)} = \frac{M}{\alpha}. \quad (9)$$

```
Overlapping Density Based Clustering Algorithm (OverDBC)
Input: Expression Matrix (X)
output: Overlap clusters set (C_new)
//phase 1: For each two point g_i and g_j in X
    Find the value of Similarity matrix (S_ij)
//phase 2: Find_core_list(X, S);
//phase 3: For all O_i of core_list do
            C_i = next_cluster
            expandcluster(O_i, C_i, Neighbors);
            add C_i to C
//phase 4: Func_bound_over(C, C_new);

Expandcluster(O_i, C_i, Neighbors)
  C_i = Link list.new ( );
  For each point g in neighbors
    If g is in Volume(O_i)
       Neighbors = neighbors U neighbors' g
Return C_i
Find_core_list(X, S)
  Core_list = undefined
   For each g_i in X
     If Density(g_i) > avg_Density
       Core_list.insert(g_i);
   Core_list. Sort( ) base on Closeness Centrality value
   Return Sorted Core_list;
Func_bound_over(C, C_new)
 Compute λ (the maximum number of overlap point)
For all C_i, C_j in C
If C_i ∩ C_j >= λ then
  C_i = merge(C_i, C_j);
  Delete C_j from C
Return C
```

Algorithm 1: Graphic Bayesian model for overlap clustering.

Equation (8) is achieved by exploiting the recursive definition of the gamma function, where we have used the fact that $\Gamma(x) = (x-1) * \Gamma(x-1)$ for $x > 1$ [24]. So (7) can be rewritten as

$$P(\pi_k) = \frac{\alpha}{M}\pi_k^{r-1}(1-\pi_k)^{s-1}. \tag{10}$$

4.2. Probability Model of Clusters and Data Distribution. According to the Bayesian model presented in the previous section, computing methods of $P(C_i)$ and $P(D)$ will be described in this section. As defined in Section 4, the hypothesis H_1^i shows that all the data in cluster i are in fact generated independently and only belong to cluster i. The alternative hypothesis $H_1^{\prime i}$ states that data in cluster i may belong to two or more clusters. Obviously, the relation (11) exists between H_1^i and $H_1^{\prime i}$:

$$P\left(H_1^{\prime i}\right) = 1 - P\left(H_1^i\right). \tag{11}$$

Thus, considering the graphical model (Figure 1), $P(C_i)$ is computed by

$$P(C_i) = P\left(C_i \mid H_1^i\right) * P\left(H_1^i\right) + P\left(C_i \mid H_1^{\prime i}\right) * P\left(H_1^{\prime i}\right). \tag{12}$$

$P(H_1^i)$ is the prior probability of H_1^i. To compute $P(C_i)$ from (12), first, $P(H_1^{\prime i})$ is computed. If x_j represents a point and T_{jk} is the value of transaction matrix for jth point and kth core (whose value is zero or one) then the expected number of presence x_j in different clusters ($E[x_j]$) will be computed by

$$E\left[x_j\right] = \sum_{k=1}^{M} T_{jk}. \tag{13}$$

The greater value for $E[x_j]$ shows the more probability of the presence of x_j in some clusters. If $E[x_j]$ is computed for all expected x_j in C_i, then $P(H_1^{\prime i})$ is obtained by

$$P\left(H_1^{\prime i}\right) = \prod_{j=1}^{n}\left(1 - \frac{E\left[x_j\right]}{k}\right) \quad \forall x_j \text{ where } T_{ji} = 1. \tag{14}$$

$P(H_1^i)$ can be obtained based on (11) and (14).

To compute $P(C_i \mid H_1^i)$, the IBP model [24] will be used. IBP is a simple generative process obtained from the case of customers eating from Indian buffets. N customers (i.e., data points in our clustering model) line up on one side of an Indian buffet with infinite number of dishes (i.e., clusters).

The first customer serves himself from Poisson (α) dishes (α is the concentration parameter of clusters). The next customers serve themselves dishes in proportion to the dish popularity, such that customer i serves herself the dish k with probability m_k/i, where m_k is the number of previous customers which had served themselves with dish k. $P(C_i \mid H_1^i)$ is obtained in

$$P\left(C_i \mid H_1^i\right) = \frac{P\left(H_1^i\right) * P\left(H_1^i \mid C_i\right)}{P\left(H_1^i\right) * P\left(H_1^i \mid C_i\right) + P\left(H_1^{\prime i}\right) * P\left(H_1^{\prime i} \mid C_i\right)}. \tag{15}$$

By using the IBP model, $P(H_1^i \mid C_i)$ is computed by

$$P\left(H_1^i \mid C_i\right) = \frac{n_{ci} - n_{ci(\mathrm{ov})} + \alpha/M}{N + \alpha/M}, \tag{16}$$

where n_{ci} is the number of objects in C_i. $n_{ci(\mathrm{ov})}$ is defined as a new symbol for the expected number of objects in C_i presented in multiple clusters; the value of $n_{ci(\mathrm{ov})}$ is equal to the number of x_i in which for them $E[x_i] > 1$ is satisfied. Therefore, $P(C_i \mid H_1^{\prime i})$ is computed by

$$P\left(H_1^{\prime i} \mid C_i\right) = \frac{n_{ci(\mathrm{ov})} + \alpha/k}{N + \alpha/k}. \tag{17}$$

Based on (17), It is clear that the greater value for $n_{ci(\mathrm{ov})}$ reduces the probability of C_i formation. $P(C_i \mid H_1^{\prime i})$ can be obtained in a similar way in (15). By placing (14) and (15) in (12), the value of $P(C_i)$ will be computed.

In the following, the computation of $P(D)$ will be described. Graphical model in Figure 1 represents a dataset $D = \{x_1, \ldots, x_N\}$, which is generated independently and uniquely from a probability model with vector parameters θ. Each of the x_i is a one-dimensional vector. Generally, microarray data (which are used for the evaluation algorithm) have a normal distribution, so θ could be normal distribution parameters (μ and σ vectors) which are the median and variance, respectively. By using dataset D, the conditional probability of D can be computed by using the following [25]:

$$\begin{aligned} P\left(D \mid \mu, \sigma^2\right) &= \prod_{i=1}^{n} P\left(x_i \mid \mu, \sigma^2\right) \\ &= \left(2\pi\sigma^2\right)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(x_i - \mu\right)^2\right\}. \end{aligned} \tag{18}$$

In the graphic model, some of the variables may be latent or unobserved. For example we might not know the mean and variance of the Gaussian distribution which generated our data, and we may also be interested in inferring these values. If there is information about D, values of hidden variables could be inferred using Variational Bayes method [26].

To complete the last part of (1), that is, the overlap probability for ith and jth clusters, $P(\mathrm{ov}_{ij=1} \mid T, D, C_i)$ is computed based on the number of data points expected to be present in two clusters simultaneously by the following:

$$P\left(\mathrm{ov}_{ij=1} \mid T, D, C_i\right) = \frac{\left|n_{cij(\mathrm{ov})}\right|}{n_{ci} + n_{cj}}. \tag{19}$$

In (16), n_{ci} is the expected number of points in C_i, n_{cj} is the expected number of points in C_j, and $n_{cij(\mathrm{ov})}$ is the expected number of points in both C_i and C_j. These parameters are computed using transaction matrix (T).

By computing $P(\mathrm{ov}_{ij} = 1, T, X, C_i)$ in (1), prediction of overlap degree between the new cluster and all other previous clusters is possible. In Section 5, this prediction will be used to provide an overlap Bayesian clustering algorithm.

5. Bayesian-OverDBC Algorithm

In this section, Bayesian-OverDBC algorithm will be introduced. This algorithm defines a probability model for the data which can be used to predict the distribution of overlap clusters. This algorithm completes OverDBC algorithm. More details of OverDBC are in [3]. OverDBC consists of four phases which are as follows:

(1) Selection of the original core points.

(2) Density estimation and determining whether a selected point is really a core or not.

(3) Improving clustering by using local search around core points.

(4) Merging clusters if it is possible (in case clusters have excessive same genes).

The first phases ((1)–(3)) in Bayesian-OverDBC are the same as in OverDBC. The primary difference of these two algorithms is in phase (4). In this phase, based on Bayesian model shown in Figure 1, the overlap probability of a new cluster is computed in comparison to all other previous clusters. If the overlap probability is smaller than θ, the local search is continued around the core and a new cluster is formed. Value of θ will be determined by trial and error method on the dataset. If the overlap probability is more than θ, func_bound_over() is invoked. This function determines a lower bound on the number of shared objects of two clusters drown from a given dataset. The func_bound_over() is defined based on double counting theory [21] and provides great improvement in overlap clustering. Output of the function is represented as λ. If the number of overlap points of two clusters is greater than λ, the two clusters should be merged to form a larger cluster. Obviously, with these changes the membership matrix Z is also changed.

Bayesian-OverDBC (Algorithm 2) has many advantages over traditional density-based clustering methods. It defines a probabilistic model of data which can be used to predicate distribution of overlap clusters. Bayesian hypothesis testing could be used to decide which of the clusters exists as overlap clusters and which one merges or even is discarded. In the next section the results of comparison of Bayesian-OverDBC with other algorithms will be described.

6. Evaluation

Our evaluation experiments were performed on two different types of data: synthetic microarray-like data and real dataset of microarray. By using microarray techniques; it is possible

Baysian Overlapping Density Based Clustering Algorithm (OverDBC)
Input: Expression Matrix (D) $n * p$, Data model $P(D \mid \theta)$
Output: Bayesian overlap clusters, Z (membership-matrix).
New cluster may be merged based on $P(\text{ov}_{ij})$ probability.
//phase 1
(1) compute transaction matrix (T) $N * M$
//phase 2
(2) Find Core genes based on Density and Closeness Centrality
(3) Add gene g_i to Core genes (O) based on density and Cc relations.
//phase 3
(4) For All g_j in O (Set of Core Object) Repeat:
(5) If $j = 1$ Start Local Search to find nearest neighbors g_1, Save cluster C_1.
Else For $i = 1$ to $j - 1$
$P\left(\text{ov}_{ij} = 1, T, D, C_i\right) = P\left(\text{ov}_{ij=1} \mid T, X, C_i\right) * P(T) * P(D) * P\left(C_i\right)$
Based above probability select one of these paths:
if $P < \theta$ then Start local search to construct new cluster C_j.
Else *invoke func_bound_over( )* and return results
End of For
End of If

Algorithm 2: Bayesian-OverDBC.

Table 1: Comparison of precision, recall, and $F1$ measures.

Dataset	Precision			Recall			$F1$		
	Algorithm								
	DBSCAN	IOMM	Bayesian-OverDBC	DBSCAN	IOMM	Bayesian-OverDBC	DBSCAN	IOMM	Bayesian-OverDBC
Small-synthetic	.81	.83	**.76**	.53	.57	.63	0.6886	0.6758	0.6889
Medium-synthetic	.66	.73	.73	.72	.67	**.69**	0.8335	0.6987	0.7094
Large-synthetic	.87	.81	.87	.80	.83	.86	0.5192	0.8198	0.8649
DS1	.42	.67	.79	.68	.73	.82	0.6182	0.6987	0.804
DS2	.56	.74	.74	.64	.70	.76	0.5966	0.7194	0.7498

to measure the expression levels of thousands of genes under several experimental conditions. Microarray data provide a lot of information about the molecule transaction in genome level, which is important for gene regulatory network detection. In a formal representation, microarray data were represented as a matrix. Rows represent genes and columns represent conditions. i'th and j'th matrix member shows the expression level of gene i in condition j. In [11], apart from demonstrating their approach on gene microarray data and evaluating standard biology databases, they also showed results on microarray-like synthetic data. We employed three synthetic datasets of different sizes:

(1) Small-synthetic dataset: a dataset with $n = 75$.

(2) Medium-synthetic dataset: a dataset with $n = 200$.

(3) Large-synthetic dataset: a dataset with $n = 1000$.

Bayesian OverDBC has been evaluated on two real datasets of microarray gene expression data. The algorithm has been implemented on the Arabidopsis thaliana abiotic stress dataset (DS1) [27] and on the yeast cell cycle dataset (DS2) [28].

DS1 is a 3D dataset from multiple sclerosis patients which has been published in 2003. The condition dimension consisted of 13 multiple-sclerosis patients, monitored over 7 time points after IFN-β injection. The Arabidopsis thaliana datasets were composed of different abiotic stress stimulus experiments conducted in the root and shoot tissue.

DS2 was extracted from a dataset that shows the fluctuation of expression levels of approximately 6000 genes over two cell cycles (17 time points).

To evaluate the clustering results, precision, recall, and F-measure were calculated over pairs of points. These measures try to determine whether the prediction of the pair existing in the same cluster was correct with respect to the underlying true categories in the data. Precision is calculated as the fraction of pairs correctly put in the same cluster. Recall is the fraction of actual pairs that were identified. F-measure is the harmonic mean of precision and recall.

We compared Bayesian-OverDBC results with DBSCAN, which can only assign each object to a single cluster. We compared these algorithms using an $F1$ score, which takes into accounts both precision and recall and which can be computed from the true gene assignments to clusters. Also, we compared Bayesian-OverDBC with IOMM that allows genes to belong to multiple overlap clusters (Table 1).

The first column is the name of the datasets, the second column is precision value, and the third and fourth columns are recall and $F1$ measure.

Table 2: Notations for omega index.

Method 1\method 2	$c1$	$c2$	$\cdots$	$c\|C\|$	Sums
$t1$	$n11$	$n12$	$\cdots$	$n1\|C\|$	$n1.$
$t2$	$n21$	$n22$	$\cdots$	$n2\|C\|$	$n2.$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
$t\|T\|$	$n\|T\|1$	$n\|T\|2$	$\cdots$	$n\|T\|\|C\|$	$n\|T\|.$
Sums	$n.1$	$n.2$	$\cdots$	$n.\|C\|$	n

Although in a few positions the values of precision or recall for Bayesian-OverDBC is lower than other algorithms, the $F1$ measure has higher values in comparison with other methods indicating the good performance of Bayesian-OverDBC algorithm.

We compared our method with IOMM using omega index. The omega index extends the Adjusted Rand Index (ARI) [29] to overlapping clustering [30]. In addition to counting the number of common pairs occurring together in 0 cluster or 1 cluster, Omega index also counts the number of pairs occurring together in $2, \ldots, M$ clusters. Using the terms from Table 2 omega index (Ω) and expected omega ($E[\Omega]$) are computed by the following, respectively:

$$\Omega = \frac{\sum_{j=0}^{\min(|\tau|,|c|)} n_j}{N}, \tag{20}$$

$$E[\Omega] = \frac{\sum_{j=0}^{\min(|\tau|,|c|)} n_{j.} n_{.j}}{n^2}. \tag{21}$$

Table 2 shows parameters used in omega index. It contains symbols which are required to compare two methods of clustering.

In Table 2, clusters in the first algorithm (Bayesian-OverDBC) form rows and clusters in the second algorithm (IOMM) form columns. So, $|\tau|$ is the number of clusters in the first algorithm and $|c|$ is the number of clusters in the second algorithm. In this table, n_{ij} is the number of objects which are in ith cluster by method 1 and in jth cluster be method 2. n_{ii} (n_i as a brief form) is the number of objects which are the same clusters by method 1 and method 2. More details about omega index are in [30]. The omega index requires an adjustment to remove clusters sharing the same number of labels by chance which is computed by (22)

$$\begin{aligned}\Omega_{\text{adj}} &= \frac{\left(\sum_{j=0}^{\min(|\tau|,|c|)} n_j\right)/N - E[\Omega]}{1 - E[\Omega]} \\ &= \frac{\text{observed index} - \text{expected index}}{\text{maximun index} - \text{expected index}}.\end{aligned} \tag{22}$$

Of the other metrics such as NMI, PNMI, and aligned NMI [30], the omega index gives the most optimistic measure of multiple-membership similarity. We compared Bayesian-OverDBC and IOMM using omega index and we found $\Omega_{\text{adj}} = .83$ for DS1 and $\Omega_{\text{adj}} = .86$ for DS2. It indicates that Bayesian-OverDBC assigns data points to overlap clusters in a similar way with IOMM.

These results also show that Bayesian-OverDBC is an effective density-based method for overlap clustering and its performance in finding relevant pairs is very similar to or even better than IOMM. Furthermore, IOMM sampler should be run for 2000–3000 iterations. Time complexity of IOMM is $O(n^2)$ and the time complexity of Bayesian-OverDBC is $O(N \cdot M)$. As a result, Bayesian-OverDBC has better performance in time complexity compared to IOMM.

7. Discussion

This paper explained Bayesian-OverDBC which is a new density-based clustering method modelling overlapped clusters. The Bayesian-OverDBC extends traditional density-based model using probabilistic method to find and predict overlap clusters. While most of the research in this area has focused on disjoint clustering, many real microarray datasets, and as a result many gene regulatory networks, have inherent overlapping partitions. Density-based clustering methods, even with the ability of producing overlapping clusters, do not use a probabilistic model. So, it is difficult to determine probability of events and to compare an overlapping method with other methods. Therefore, a probability density-based clustering model, which provides overlapping, is required. It is proved that Bayesian overlapping clustering may be significantly better than other similar methods of clustering. As overlapping clustering is a still-developing field, there are several subjects for future development such as techniques for visualization and interpretation, new algorithms and new means of comparison, and techniques for model selection.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] R. Harpaz and R. Haralick, "Exploiting the geometry of gene expression patterns for unsupervised learning," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, pp. 670–674, IEEE, Hong Kong, August 2006.

[2] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Addison Wesley, 2011.

[3] M. Mirzaie, A. Barani, N. NematBakhsh, and M. Beigi, "OverDBC: a new density-based clustering method for detecting overlapped clusters from microarray data," *IDA Journal*, vol. 19, no. 6, 2015.

[4] K. A. Heller and Z. Ghahramani, "A nonparametric Bayesian approach to modelling overlapping clusters," in *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS '07)*, vol. 2 of *JMLR Workshop and Conference Proceedings*, pp. 187–194, San Juan, Puerto Rico, March 2007.

[5] A. H. Katherine and Z. Ghahramani, "A nonparametric Bayesian approach to modelling overlapping clusters," in *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS '07)*, San Juan, Puerto Rico, March 2007.

[6] X. Bai, S. Luo, and Y. Zhao, "Entropy based soft K-means clustering," in *Proceedings of the IEEE International Conference on Granular Computing (GrC '08)*, pp. 107–110, August 2008.

[7] O. Alter, P. O. Brown, and D. Botstein, "Singular value decomposition for genome-wide expression data processing and modeling," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 18, pp. 10101–10106, 2000.

[8] N. Jardine and R. Sibson, "The construction of hierarchic and non-hierarchic classifications," *Computer Journal*, vol. 11, no. 2, pp. 177–184, 1968.

[9] P. Bertrand and E. Diday, "A visual representation of the compatibility between an order and a dissimilarity index: the pyramids," *Computational Statistics Quarterly*, vol. 2, no. 1, pp. 31–41, 1985.

[10] R. N. Shepard and P. Arabie, "Additive clustering: representation of similarities as combinations of discrete overlapping properties," *Psychological Review*, vol. 86, no. 2, pp. 87–123, 1979.

[11] E. Segal, A. Battle, and D. Koller, "Decomposing gene expression into cellular processes," in *Proceedings of the 8th Pacific Symposium on Biocomputing (PSB '03)*, pp. 89–100, Lihue, Hawaii, USA, January 2003.

[12] 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*, vol. 39, no. 1, pp. 1–38, 1977.

[13] Y. Cheng and G. M. Church, "Biclustering of expression data," in *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ICMB '00)*, pp. 93–103, La Jolla, Calif, USA, August 2000.

[14] H. Shan and A. Banerjee, "Bayesian Co-clustering," in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM '08)*, pp. 530–539, Pisa, Italy, December 2008.

[15] Y. Guan, J. G. Dy, D. Niu, and Z. Ghahramani, "Variational inference for nonparametric multiple clustering," in *Proceedings of the Workshop on Discovering, Summarizing and Using Multiple Clustering at the ACM SIGKDD International Conference on Knowledge Discovering and Data Mining (MultiClust '10)*, 2010.

[16] V. Mansinghka, E. Jonas, C. Petschulat, B. Cronin, P. Shafto, and J. Tenenbaum, "Cross-categorization: a method for discovering multiple overlapping clusterings," in *Proceedings of the Nonparametric Bayes Workshop at NIPS*, Whistler, Canada, December 2009.

[17] T. T. Grifths and Z. Z. Ghahramani, "Infinite latent feature models and the Indian buffet process," in *Proceedings of the 20th Neural Information Processing Systems*, Vancouver, Canada, December 2006.

[18] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD '96)*, pp. 291–316, Portland, Ore, USA, August 1996.

[19] M. Ankerst, M. Breunig, H. P. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 49–60, Philadelphia, Pa, USA, June 1999.

[20] A. Hinneburg and H.-H. Gabriel, "Denclue 2.0: fast clustering based on kernel density estimation," in *Advances in Intelligent Data Analysis VII*, vol. 4723 of *Lecture Notes in Computer Science*, pp. 70–80, Springer, Berlin, Germany, 2007.

[21] S. Jukna, *Extremal Combinatorics with Applications in Computer Science*, Springer, New York, NY, USA, 2nd edition, 2011.

[22] K. A. Heller and Z. Ghahramani, "Bayesian hierarchical clustering," in *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, pp. 297–304, August 2005.

[23] E. Alpaydın, *Introduction to Machine Learning*, The MIT Press, Cambridge, Mass, USA, 2010.

[24] T. L. Griffiths and Z. Ghahramani, "The Indian buffet process: an introduction and review," *Journal of Machine Learning Research*, vol. 12, pp. 1185–1224, 2011.

[25] E. Biçici and D. Yuret, "Locally scaled density based clustering," in *Adaptive and Natural Computing Algorithms*, vol. 4431 of *Lecture Notes in Computer Science*, pp. 739–748, Springer, Berlin, Germany, 2007.

[26] C. Fox and S. Roberts, "A tutorial on variational Bayesian inference," *Artificial Intelligence Review*, vol. 38, no. 2, pp. 85–95, 2012.

[27] M. Garcia-Hernandez, T. Z. Berardini, G. Chen et al., "TAIR: a resource for integrated *Arabidopsis* data," *Functional & Integrative Genomics*, vol. 2, no. 6, pp. 239–253, 2002.

[28] P. T. Spellman, G. Sherlock, M. Q. Zhang et al., "Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization," *Molecular Biology of the Cell*, vol. 9, no. 12, pp. 3273–3297, 1998.

[29] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[30] L. M. Collins and C. W. Dent, "Omega: a general formulation of the rand index of cluster recovery suitable for non-disjoint solutions," *Multivariate Behavioral Research*, vol. 23, no. 2, pp. 231–242, 1988.