# A Mixture Model for Clustering Ensembles

Alexander Topchy   Anil K. Jain   William Punch

## Abstract

Clustering ensembles have emerged as a powerful method for improving both the robustness and the stability of unsupervised classification solutions. However, finding a consensus clustering from multiple partitions is a difficult problem that can be approached from graph-based, combinatorial or statistical perspectives. We offer a probabilistic model of consensus using a finite mixture of multinomial distributions in a space of clusterings. A combined partition is found as a solution to the corresponding maximum likelihood problem using the EM algorithm. The excellent scalability of this algorithm and comprehensible underlying model are particularly important for clustering of large datasets. This study compares the performance of the EM consensus algorithm with other fusion approaches for clustering ensembles. We also analyze clustering ensembles with incomplete information and the effect of missing cluster labels on the quality of overall consensus. Experimental results demonstrate the effectiveness of the proposed method on large real-world datasets.

**keywords**: unsupervised learning, clustering ensemble, consensus function, mixture model, EM algorithm.

## 1 Introduction

Data clustering is a difficult inverse problem, and as such is ill-posed when prior information about the underlying data distributions is not well defined [15, 16, 21]. Numerous clustering algorithms are capable of producing different partitions of the same data that capture various distinct aspects of the data. The exploratory nature of clustering tasks demands efficient methods that would benefit from combining the strengths of many individual clustering algorithms. This is the focus of research on clustering ensembles, seeking a combination of multiple partitions that provides improved overall clustering of the given data. Clustering ensembles can go beyond what is typically achieved by a single clustering algorithm in several respects:

- *Robustness*. Better average performance across the domains and datasets.
- *Novelty*. Finding a combined solution unattainable by any single clustering algorithm.
- *Stability and confidence estimation*. Clustering solutions with lower sensitivity to noise, outliers or sampling variations. Clustering uncertainty can be assessed from ensemble distributions.
- *Parallelization and Scalability*. Parallel clustering of data subsets with subsequent combination of results. Ability to integrate solutions from multiple distributed sources of data or attributes (features).

Clustering ensembles can also be used in multiobjective clustering as a compromise between individual clusterings with conflicting objective functions and play an important role in distributed data mining [27].

The problem of combination of multiple clusterings brings its own new challenges. The major difficulty is in finding a consensus partition from the output partitions of various clustering algorithms. Unlike supervised classification, the patterns are unlabeled and, therefore, there is no explicit correspondence between the labels delivered by different clusterings. An extra complexity arises when different partitions contain different numbers of clusters, often resulting in an intractable label correspondence problem. The combination of multiple clustering can also be viewed as finding a median partition with respect to the given partitions which is proven to be NP-complete [2].

Another challenging issue is the choice of the clustering algorithms for the ensemble. Diversity of the individual clusterings can be achieved by a number of approaches, including: using different conventional algorithms [30], their relaxed versions [31], built-in randomness [11, 12], or by data sampling [8, 9, 25].

This work focuses on the primary problem of clustering ensembles, namely the consensus function, which creates the combined clustering. We propose a new fusion method for these kinds of unsupervised decisions that is based on a probability model of the consensus

Department of Computer Science and Engineering,
Michigan State University, East Lansing, MI, 48824, USA
{topchyal, jain, punch}@cse.msu.edu. This research was supported by ONR grant N00014-01-1-0266 (AKJ) and fellowship award from Center for Biological Modeling at Michigan State University (AT)

partition in the space of contributing clusters. The consensus partition is found as a solution to the maximum likelihood problem for a given clustering ensemble. The likelihood function of an ensemble is optimized with respect to the parameters of a finite mixture distribution. Each component in this distribution corresponds to a cluster in the target consensus partition, and is assumed to be a multivariate, multinomial distribution. The maximum likelihood problem is solved using the EM algorithm [6]. There are three main advantages to our approach:

1. It completely avoids solving the label correspondence problem.
2. The low computational complexity of the EM consensus function – $O(kNH)$ for $k$ clusters in the target partition, $N$ patterns, and $H$ clusterings in the ensemble. This results in fast convergence that is comparable to the $k$-means algorithm.
3. The ability to handle missing data, in this case missing cluster labels (or labels determined to be unknown) for certain patterns in the ensemble (for example, when bootstrap method is used to generate the ensemble).

This finite mixture approach is readily applicable to large datasets, as opposed to other consensus functions which are based on the co-association of patterns in clusters from an ensemble with quadratic complexity $O(kN^2H)$. Moreover, unlike algorithms that search for a consensus partition via re-labeling and subsequent voting, this approach is not constrained to a predetermined number of clusters in the ensemble partitions. Our approach can operate with arbitrary partitions with varying numbers of clusters, not necessarily equal to the target number of clusters in the consensus partition.

One way to understand the proposed probability model and possible mechanism behind the consensus formation is to view the multiple clusterings as new features "extracted" by the individual clustering algorithms. It is these new features that our consensus function uses to obtain a final partition. Whereas the finite mixture model may not be valid for the patterns in original space (the initial representation), this model more naturally explains the separation of groups of patterns in the space of "extracted" features (labels generated by the partitions). It is somewhat reminiscent of classification approaches based on kernel methods which rely on linear discriminant functions in the transformed space. For example, Support Vector Clustering [4] seeks spherical clusters after the kernel transformation that correspond to more complex cluster shapes in the original pattern space.

Section 2 describes relevant research on clustering combination. In Section 3 we formally introduce the problem of clustering combination and briefly analyze existing approaches to consensus. Section 4 presents a finite mixture probability model and derives the EM algorithm for the maximum likelihood parameter estimation. Finally, in Section 5, we experimentally compare the finite mixture approach with several known consensus functions. This empirical study demonstrates the accuracy of the consensus clustering that emerges from multiple partitions. We specifically analyze the dependence of combination accuracy as a function of the number of partitions in the ensemble and the number of clusters in the partitions. We also consider the effect of missing cluster labels on the overall quality of clustering, by varying the percentage of patterns with deleted labels.

## 2 Motivation and Related Work

Approaches to combination of clusterings differ in two main respects, namely the way in which the contributing component clusterings are obtained and the method by which they are combined. Fred [11] proposed to summarize various clustering results in a co-association matrix. Co-association values represent the strength of association between objects by analyzing how often each pair of objects appears in the same cluster. The final clustering in [11] is determined using a voting-type algorithm applied to the co-association matrix. Hence the co-association matrix serves as a similarity matrix for the data items. Clusters are formed from the co-association matrix by linking the objects whose co-association value exceeds a certain threshold. Further work by Fred and Jain [12] also used co-association values, but instead of a fixed threshold, they applied a hierarchical (single-link) clustering to the co-association matrix. These studies use the k-means algorithm with various values of $k$ and random initializations (for selecting the $k$ cluster centers) for generating the component clusterings.

Strehl and Ghosh [30] have considered three different consensus functions for ensemble clustering: one is similar to the evidence accumulation approach [11, 12], and the other two methods are based on a hypergraph representation. All of them use various hypergraph operations to search for a solution. The Cluster-based Similarity Partitioning Algorithm (CSPA) [30] induces a graph from a co-association matrix and clusters it using the METIS algorithm [19]. Hypergraph partitioning algorithm (HGPA) [30] represents each cluster by a hyperedge in a graph where the nodes correspond to a given set of objects. Good hypergraph partitions are found using minimal cut algorithms such as HMETIS [18] coupled with the proper objective functions, which also control partition size. Hyperedge collapsing operations are considered in another hypergraph-based Meta-Clustering algorithm (MCLA) in [30]. The meta-clustering algorithm uses these operations to determine soft cluster-membership values for each object.

A different consensus function was developed in [31] based on information-theoretic principles, namely using generalized mutual information. It was shown that the underlying objective function is equivalent to the total intra-cluster variance of the partition in the specially transformed space of labels. Therefore, the *k*-means algorithm in such a space can quickly find corresponding consensus solutions. The work [31] also employed ensembles of so-called weak clustering algorithms and demonstrated how accurate consensus can be obtained from not very reliable components.

A combination of partitions obtained from multiple bootstrap samples is implemented in [8, 9]. Both works pursued direct re-labeling approaches to the correspondence problem. A re-labeling can be done optimally between two clusterings using the Hungarian algorithm. This was used in [8] to re-label each bootstrap partition using a single reference partition. The reference partition is determined by a single clustering of the entire data set. After an overall consistent re-labeling, voting can be applied to determine cluster membership for each pattern.

Bagging of multiple *k*-means clustering results was done in [23] by clustering *k*-means centers and assigning the objects to the closest cluster center. In fact, their component clusterings do not keep information about the individual object labels but only information about cluster prototypes. The *k*-means centers are "bagged" and clustered by a hierarchical procedure. Such an approach [23] approach is unique in that the components of the combination and the final clustering are defined implicitly via prototypes rather than by explicit labelings.

Kellam et al. [20] also combined clusterings through a type of co-association matrix. However, this matrix is used only to find the clusters with the highest value of support based on object co-occurrences. As a result, only a set of so-called robust clusters is produced which may not contain all the initial objects

A genetic algorithm is employed in [13] to produce the most stable partitions from an evolving ensemble (population) of clustering algorithms along with a special objective function. The objective function evaluates multiple partitions according to changes caused by data perturbations and prefers those clusterings that are least susceptible to those perturbations.

Dimitriadou et al. [5] proposed a voting/merging procedure that combines clusterings pair-wise and iteratively. The cluster correspondence problem must be solved at each iteration and the solution is not unique. Fuzzy membership decisions are accumulated during the course of merging. The final clustering is obtained by assigning each object to a derived cluster with the highest membership value.

The distributed clustering algorithm [17] constructs a global dendrogram for a set of objects from multiple local models produced by single-link algorithms. Collective hierarchical clustering combines dendrograms built on different subsets of features. The global hierarchy uses a specific approximation of distance that is estimated based on merging thresholds of component dendrograms.

## 3 Consensus Functions

Combination of multiple partitions can be viewed as a partitioning task itself. Typically, each partition in the combination is represented as a set of labels assigned by a clustering algorithm. The combined partition is obtained as a result of yet another clustering algorithm whose inputs are the cluster labels of the contributing partitions. We will assume that the labels are nominal values. In general, the clusterings can be "soft", i.e. described by the real values indicating the degree of pattern membership in each cluster in a partition. We consider only "hard" partitions below, noting however, that combination of "soft" partitions can be solved by numerous clustering algorithms and does not appear to be more complex.

Suppose we are given a set of $N$ data points $\boldsymbol{X} = \{\mathbf{x}_1,\ldots, \mathbf{x}_N\}$ and a set of $H$ partitions $\Pi = \{\pi_1,\ldots, \pi_H\}$ of objects in $\boldsymbol{X}$. Different partitions of $\boldsymbol{X}$ return a set of labels for each point $\mathbf{x}_i$, $i = 1,\ldots, N$:

$$\mathbf{x}_i \rightarrow \{\pi_1(\mathbf{x}_i), \pi_2(\mathbf{x}_i), \ldots, \pi_H(\mathbf{x}_i)\}. \tag{1}$$

Here, $H$ different clusterings are indicated and $\pi_j(\mathbf{x}_i)$ denotes a label assigned to $\mathbf{x}_i$ by the $j$-th algorithm. No assumption is made about the correspondence between the labels produced by different clustering algorithms. For simplicity, we use the notation $y_{ij} = \pi_j(\mathbf{x}_i)$ or $\mathbf{y}_i = \boldsymbol{\pi}(\mathbf{x}_i)$. The problem of clustering combination is to find a new partition $\pi_C$ of data $\boldsymbol{X}$ that summarizes the information from the gathered partitions $\Pi$. Our main goal is to construct a consensus partition without the assistance of the original patterns in $\boldsymbol{X}$, but only from their labels $\boldsymbol{Y}$ delivered by the contributing clustering algorithms. Thus, such potentially important issues as the underlying structure of both the partitions and data are ignored for the sake of a solution to the pure unsupervised consensus problem.

We emphasize that a space of new features is induced by the set $\Pi$. One can view each component partition $\pi_i$ as a new feature with categorical values, i.e. cluster labels. The values assumed by the $i$-th new feature are simply the cluster labels from partition $\pi_i$. Therefore, membership of an object $\mathbf{x}$ in different partitions is treated as a new feature vector $\mathbf{y} = \boldsymbol{\pi}(\mathbf{x})$, an $H$-tuple. In this case, one can consider partition $\pi_j(x)$ as a feature extraction function. Combination of clusterings becomes equivalent to the problem of clustering of $H$-tuples if we use only the existing clusterings $\{\pi_1,\ldots, \pi_H\}$, without the original features of data $\boldsymbol{X}$. Hence the problem of combining partitions can be viewed as a categorical clustering problem. The consensus clustering is found as a partition $\pi_C$ of a set of vectors $\boldsymbol{Y} = \{\mathbf{y}_i\}$ that directly translates to the partition of the underlying data points $\{\mathbf{x}_\text{i}\}$. Unique properties of the space of cluster labels are exploited by several known consensus functions that we briefly review below.

**3.1 Co-association Methods.** Similarity between objects can be estimated by the number of clusters shared by two objects in all the partitions of an ensemble. This similarity definition expresses the strength of co-association of objects by a matrix containing the values:

$$S_{ij} = S(x_i, x_j) = \frac{1}{H}\sum_{k=1}^{H} \delta(\pi_k(x_i), \pi_k(x_j)),$$

$$\delta(a,b) \equiv \begin{cases} 1, \text{ if } a = b \\ 0, \text{ if } a \neq b \end{cases}.$$

Thus, one can use numerous similarity-based clustering algorithms by applying them to the matrix of co-association values. It is also possible to construct more sophisticated similarity definitions when the ensemble $\Pi$ contains “soft” partitions. There are three main concerns with this intuitively appealing approach. First, it has a quadratic complexity in the number of patterns $O(N^2)$. Second, there are no established guidelines concerning which clustering algorithm should be applied, e.g. single linkage or complete linkage. The shapes of the clusters embedded in the space of clustering labels may or may not reflect cluster shapes in the original space. Third, an ensemble with a small number of clusterings may not provide a reliable estimate of the co-association values

**3.2 Hypergraph Methods.** All the clusters in the ensemble partitions can be represented as hyperedges on a graph with $N$ vertices. Each hyperedge describes a set of objects belonging to the same cluster. A consensus function is formulated as a solution to $k$-way min-cut hypergraph partitioning problem. Each connected component after the cut corresponds to a cluster in the consensus partition. However, the cuts only remove hyperedges as a whole. Hypergraph algorithms seem to work the best for nearly balanced clusters. Though the hypergraph partitioning problem is NP-hard, efficient heuristics have been developed for its solution. For example, complexity of CSPA, HGPA and MCLA is estimated in [30] as $O(kN^2H)$, $O(kNH)$, and $O(k^2NH^2)$, respectively.

**3.3 Mutual Information Approach.** The objective function for a clustering ensemble can be formulated as the mutual information (MI) between the empirical probability distribution of labels in the consensus partition and the labels in the ensemble. Under the assumption of independence of partitions, mutual information can be written as the sum of pair-wise MIs between target and given partitions. Using the classical definition of MI, one can easily compute its value for a candidate partition solution and the ensemble. However, such a definition does not offer a search algorithm for maximizing the consensus. An elegant solution can be obtained from a generalized definition of MI. Quadratic MI information can be effectively maximized by the $k$-means algorithm in the space of specially transformed cluster labels of the given ensemble. Computational complexity of this algorithm is low, $O(kNH)$, but it may require a few re-starts in order to avoid convergence to low quality local minima.

**3.4 Re-labeling Approach.** If a label correspondence problem is solved for all the given partitions, then a simple voting procedure can be used to assign objects in clusters. However, label correspondence is exactly what makes unsupervised combination difficult. A heuristic approximation to consistent labeling is possible. All the partitions in the ensemble can be re-labeled according to their best agreement with some chosen reference partition. The reference partition can be taken as one from the ensemble, or from a new clustering of the dataset. Also, a meaningful voting procedure assumes that the number of clusters in every given partition is the same as in the target partition. This requires that the number of clusters in the target consensus partition is known.

To summarize, existing consensus functions suffer from a number of drawbacks that include complexity, heuristic character of objective function and uncertain statistical status of the consensus solution. The next section introduces a mixture model of the clustering combination that aims to overcome these drawbacks.

## 4 A Mixture Model of Consensus

Our approach to the consensus problem is based on a finite mixture model for the probability of the cluster labels **y**=**π**(**x**) of the pattern/object **x**. The main assumption is that the labels $\mathbf{y}_i$ are modeled as random variables drawn from a probability distribution described as a mixture of multivariate component densities:

$$P(\mathbf{y}_i \mid \Theta) = \sum_{m=1}^{M} \alpha_m P_m(\mathbf{y}_i | \boldsymbol{\theta}_m), \tag{2}$$

where each component is parametrized by $\boldsymbol{\theta}_m$. The $M$ components in the mixture are identified with the clusters of the consensus partition $\pi_C$. The mixing coefficients $\alpha_m$ correspond to the prior probabilities of the clusters. In this model, data points $\{\mathbf{y}_i\}$ are presumed to be generated in two steps: first, by drawing a component according to the probability mass function $\alpha_m$, and then sampling a point from the distribution $P_m(\mathbf{y}|\theta_m)$. All the data $\boldsymbol{Y} = \{\mathbf{y}_i\}_{i=1}^{N}$ are assumed to be independent and identically distributed. This allows one to represent the log likelihood function for the parameters $\Theta=\{\alpha_1,\ldots, \alpha_M, \boldsymbol{\theta}_1,\ldots, \boldsymbol{\theta}_M\}$ given the data set $\boldsymbol{Y}$ as:

$$\begin{aligned}\log L(\Theta \mid \mathbf{Y}) &= \log \prod_{i=1}^{N} P(\mathbf{y}_i \mid \Theta) \\ &= \sum_{i=1}^{N} \log \sum_{m=1}^{M} \alpha_m P_m(\mathbf{y}_i \mid \boldsymbol{\theta}_m).\end{aligned} \tag{3}$$

The objective of consensus clustering is now formulated as a maximum likelihood estimation problem. To find the best fitting mixture density for a given data $\boldsymbol{Y}$, we must maximize the likelihood function with respect to the unknown parameters Θ:

$$\Theta^* = \underset{\Theta}{\arg\max}\ \log L(\Theta \mid \mathbf{Y}) \cdot \tag{4}$$

The next important step is to specify the model of component-conditional densities $P_m(\mathbf{y}|\theta_m)$. Note, that the original problem of clustering in the space of data $\boldsymbol{X}$ has been transformed, with the help of multiple clustering algorithms, to a space of new multivariate features **y** = **π**(**x**). To make the problem more tractable, a conditional independence assumption is made for the components of vector $\mathbf{y}_i$, namely that the conditional probability of $\mathbf{y}_i$ can be represented as the following product:

$$P_m(\mathbf{y}_i | \boldsymbol{\theta}_m) = \prod_{j=1}^{H} P_m^{(j)}(y_{ij} | \boldsymbol{\theta}_m^{(j)}) \cdot \tag{5}$$

To motivate this, one can note that even if the different clustering algorithms (indexed by $j$) are not truly independent, the approximation by product in Eq. (5) can be justified by the excellent performance of naive Bayes classifiers in discrete domains [22]. Our ultimate goal is to make a discrete label assignment to the data in $\boldsymbol{X}$ through an indirect route of density estimation of $\boldsymbol{Y}$. The assignments of patterns to the clusters in $\pi_C$ are much less sensitive to the conditional independence approximation than the estimated values of probabilities $P(\mathbf{y}_i \mid \Theta)$, as supported by the analysis of naïve Bayes classifier in [7].

The last ingredient of the mixture model is the choice of a probability density $P_m^{(j)}(y_{ij} | \boldsymbol{\theta}_m^{(j)})$ for the components of the vectors $\mathbf{y}_i$. Since the variables $y_{ij}$ take on nominal values from a set of cluster labels in the partition $\pi_j$, it is natural to view them as the outcome of a multinomial trial:

$$P_m^{(j)}(y | \boldsymbol{\theta}_m^{(j)}) = \prod_{k=1}^{K(j)} \vartheta_{jm}(k)^{\delta(y,k)} \ . \tag{6}$$

Here, without the loss of generality, the labels of the clusters in $\pi_j$ are chosen to be integers in $\{1,\ldots,K(j)\}$. To clarify the notation, note that the probabilities of the outcomes are defined as $\vartheta_{jm}(k)$ and the product is over all the possible values of $y_{ij}$ labels of the partition $\pi_j$. Also, the probabilities sum up to one:

$$\sum_{k=1}^{K(j)} \vartheta_{jm}(k) = 1, \forall j \in \{1,\ldots,H\},\ \forall m \in \{1,\ldots,M\} \cdot \tag{7}$$

For example, if the $j$-th partition has only two clusters, and possible labels are 0 and 1, then Eq. (5) can be simplified as:

$$P_m^{(j)}(y | \boldsymbol{\theta}_m^{(j)}) = \vartheta_{jm}^{y} (1 - \vartheta_{jm})^{1-y} \tag{8}$$

The maximum likelihood problem in Eq. (3) generally cannot be solved in a closed form when all the parameters $\Theta=\{\alpha_1,\ldots, \alpha_M, \boldsymbol{\theta}_1,\ldots, \boldsymbol{\theta}_M\}$ are unknown. However, the likelihood function in Eq. (2) can be optimized using the EM algorithm. In order to adopt the EM algorithm, we hypothesize the existence of hidden data $\boldsymbol{Z}$ and the likelihood of complete data (***Y***, ***Z***). The distribution of **z** should be consistent with the observed values ***Y***:

$$\log P(\mathbf{Y} \mid \Theta) = \log \sum_{z \in \mathbf{Z}} P(\mathbf{Y}, \mathbf{z} \mid \Theta)\,. \qquad (9)$$

If the value of $\mathbf{z}_i$ is known then one could immediately tell which of the $M$ mixture components was used to generate the point $\mathbf{y}_i$. It means that with each observed data point $\mathbf{y}_i$, we associate a hidden vector variable $\mathbf{z}_i = \{z_{i1,\dots,}z_{iM}\}$ such that $z_{im}$ =1 if $\mathbf{y}_i$ belongs to the $m$-th component and $z_{im}$ = 0, otherwise. It is convenient to write the complete data likelihood as:

$$\begin{aligned}\log L(\Theta \mid \mathbf{Y}, \mathbf{Z}) &= \log \prod_{i=1}^{N} P(\mathbf{y}_i, \mathbf{z}_i \mid \Theta) \\ &= \log \prod_{i=1}^{N} \prod_{m=1}^{M} \left(\alpha_m P_m(\mathbf{y}_i \mid \boldsymbol{\theta}_m)\right)^{z_{im}} \qquad (10) \\ &= \sum_{i=1}^{N} \sum_{m=1}^{M} z_{im} \log \alpha_m P_m(\mathbf{y}_i \mid \boldsymbol{\theta}_m)\,.\end{aligned}$$

According to the general EM approach, we have to define an auxiliary function $Q(\Theta;\Theta')$ that serves as a lower bound on the observed data likelihood in Eq. (3):

$$Q(\Theta;\Theta') = \sum_{z} \log\left(P(\mathbf{Y}, \mathbf{z} \mid \Theta)\right) p(\mathbf{z} \mid \mathbf{Y}, \Theta'). \qquad (11)$$

Classical convergence analysis of EM algorithm [6, 24] establishes that the maximization of the function $Q(\Theta;\Theta')$ with respect to Θ is equivalent to increasing the observed likelihood function in Eq. (3). Evaluation of $Q(\Theta;\Theta')$ is the first step of the EM algorithm. Substitution of Eq. (10) in the definition of Q function gives:

$$\begin{aligned}Q(\Theta;\Theta') &= \sum_{z} \sum_{i=1}^{N} \sum_{m=1}^{M} z_{im} \log \alpha_m P_m(\mathbf{y}_i \mid \boldsymbol{\theta}_m) p(\mathbf{z} \mid \mathbf{Y}, \Theta') \\ &= \sum_{i=1}^{N} \sum_{m=1}^{M} E[z_{im}] \log \alpha_m P_m(\mathbf{y}_i \mid \boldsymbol{\theta}_m). \qquad (12)\end{aligned}$$

The last expression depends on the previous estimate of parameters $\Theta'$ only via the expected values of the hidden variables $E[z_{im}]$, which are defined as:

$$E[z_{im}] = \sum_{z} z_{im}\; p(\mathbf{z} \mid \mathbf{Y}, \Theta') = \frac{\alpha'_m P_m(\mathbf{y}_i \mid \boldsymbol{\theta}'_m)}{\sum_{n=1}^{M} \alpha'_n P_n(\mathbf{y}_i \mid \boldsymbol{\theta}'_n)}\,. \qquad (13)$$

Here, the current guess about the parameters $\Theta' = \{\alpha'_1, \dots, \alpha'_M, \boldsymbol{\theta}'_1, \dots, \boldsymbol{\theta}'_M\}$ is used to compute the expectations. Taking into account the concrete form of the component-conditional densities $P_m(\mathbf{y}_i \mid \boldsymbol{\theta}_m)$ from Eqs. (5) and (6), we obtain the $E$-step of the algorithm:

$$E[z_{im}] = \frac{\alpha'_m \prod_{j=1}^{H} \prod_{k=1}^{K(j)} \left(\vartheta'_{jm}(k)\right)^{\delta(y_{ij},k)}}{\sum_{n=1}^{M} \alpha'_n \prod_{j=1}^{H} \prod_{k=1}^{K(j)} \left(\vartheta'_{jn}(k)\right)^{\delta(y_{ij},k)}}\,. \qquad (14)$$

The $M$-step consists of maximizing the $Q$ function in Eq. (12) by the parameters Θ, given the expected values of the hidden variables $E[z_{im}]$ from $E$-step in Eq. (14):

$$\begin{aligned}\Theta^* &= \underset{\Theta}{\arg\max}\, Q(\Theta;\Theta') \\ &= \underset{\{\alpha_m, \vartheta_{jm}\}}{\arg\max} \sum_{i=1}^{N} \sum_{m=1}^{M} \left(E[z_{im}] \log \alpha_m + E[z_{im}] \log P_m(\mathbf{y}_i \mid \boldsymbol{\theta}_m)\right). \qquad (15)\end{aligned}$$

The two terms on the right hand side can be optimized independently. The expression for the coefficients $\alpha_m$ is easily found using a Lagrange multiplier along with the constraint $\Sigma_m\, \alpha_m = 1$:

$$\frac{\partial Q(\Theta;\Theta')}{\partial \alpha_m} = \frac{\partial}{\partial \alpha_m} \left( \sum_{i=1}^{N} \sum_{m=1}^{M} E[z_{im}] \log \alpha_m + \lambda \left( \sum_{m=1}^{M} \alpha_m - 1 \right) \right) = 0 \qquad (16)$$

$$\alpha_m = \frac{\sum_{i=1}^{N} E[z_{im}]}{\sum_{i=1}^{N} \sum_{m=1}^{M} E[z_{im}]}\,. \qquad (17)$$

Similarly, obtaining the optimizing values of $\vartheta_{jm}(k)$, is facilitated by the independence assumption for the component-conditional densities of variables $y_{ij}$ as described by Eq. (3). Again, natural constraints $\Sigma_k\, \vartheta_{jm}(k) = 1$ and Lagrange multipliers $\lambda_{jm}$ are utilized:

$$\begin{aligned}\frac{\partial Q(\Theta;\Theta')}{\partial \vartheta_{jm}(k)} = \frac{\partial}{\partial \vartheta_{jm}(k)} \Bigg( &\sum_{i=1}^{N} \sum_{m=1}^{M} E[z_{im}] \log P_m(\mathbf{y}_i \mid \boldsymbol{\theta}_m) + \\ &+ \lambda_{jm} \left( \sum_{k=1}^{K(j)} \vartheta_{jm}(k) - 1 \right) \Bigg) = 0 \qquad (18)\end{aligned}$$

$$\vartheta_{jm}(k) = \frac{\sum_{i=1}^{N} \delta(y_{ij}, k) E[z_{im}]}{\sum_{i=1}^{N} \sum_{k=1}^{K(j)} \delta(y_{ij}, k) E[z_{im}]} \qquad (19)$$

To summarize, the EM algorithm starts with an initial guess for the values of the parameters

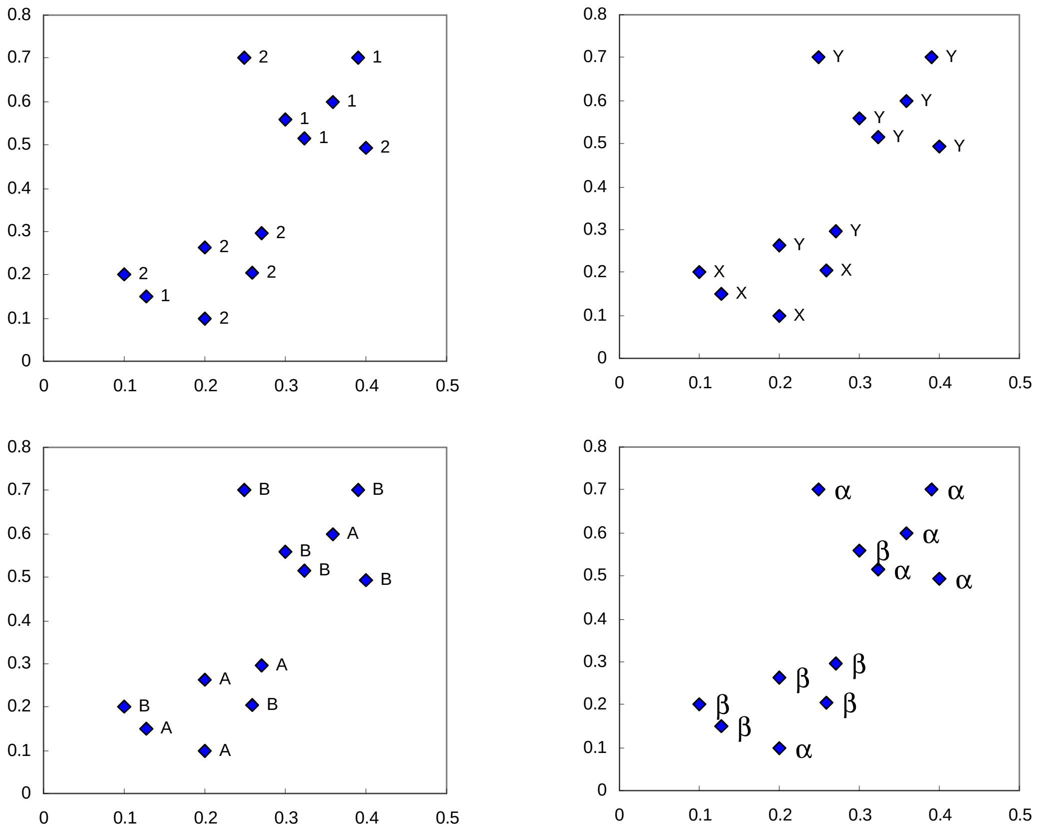

Figure 1: Four possible partitions of 12 data points into 2 clusters. Different partitions use different sets of labels

$\{\alpha'_1,\ldots,\alpha'_M,\boldsymbol{\theta}'_1,\ldots,\boldsymbol{\theta}'_M\}$ of the mixture density. After this, E and M steps are repeated until a chosen convergence criterion is satisfied. The *E*-step computes the expected values of the hidden variables $E[z_{im}]$ according to Eq. (13). The *M*-step maximizes the likelihood by computing new best parameters estimates according to Eqs. (17,19). The convergence criteria can be based on the increase in the amount of the likelihood function between two consequent M-steps or on the stability of the assignment of points from $\boldsymbol{Y}$, or equivalently from $\boldsymbol{X}$. In fact, it is the stability criterion that is more relevant to the clustering task. The solution to the consensus clustering problem is obtained by a simple inspection of the expected values of the variables $E[z_{im}]$, due to the fact that $E[z_{im}]$ represents the probability that the pattern $y_i$ was generated by the *m*-th mixture component. Once convergence is achieved, a pattern $\mathbf{y}_i$ is assigned to the component which has the largest value for the hidden label $\mathbf{z}_i$. Intuitively, each *E*-step is essentially equivalent to the naïve Bayes computation procedure. However, the M-step uses "soft" real-valued cluster memberships to determine pattern contributions to the component-conditional densities, in contrast to conventional supervised maximum likelihood estimation.

Table 1: Clustering ensemble and consensus solution

| | $\pi_1$ | $\pi_2$ | $\pi_3$ | $\pi_4$ | $E\,[z_{i1}]$ | $E\,[z_{i2}]$ | Consensus |
|---|---|---|---|---|---|---|---|
| $\mathbf{y}_1$ | 2 | B | X | β | 0.999 | 0.001 | **1** |
| $\mathbf{y}_2$ | 2 | A | X | α | 0.997 | 0.003 | **1** |
| $\mathbf{y}_3$ | 2 | A | Y | β | 0.943 | 0.057 | **1** |
| $\mathbf{y}_4$ | 2 | B | X | β | 0.999 | 0.001 | **1** |
| $\mathbf{y}_5$ | 1 | A | X | β | 0.999 | 0.001 | **1** |
| $\mathbf{y}_6$ | 2 | A | Y | β | 0.943 | 0.057 | **1** |
| $\mathbf{y}_7$ | 2 | B | Y | α | 0.124 | 0.876 | **2** |
| $\mathbf{y}_8$ | 1 | B | Y | α | 0.019 | 0.981 | **2** |
| $\mathbf{y}_9$ | 1 | B | Y | β | 0.260 | 0.740 | **2** |
| $\mathbf{y}_{10}$ | 1 | A | Y | α | 0.115 | 0.885 | **2** |
| $\mathbf{y}_{11}$ | 2 | B | Y | α | 0.124 | 0.876 | **2** |
| $\mathbf{y}_{12}$ | 1 | B | Y | α | 0.019 | 0.981 | **2** |

It is instructive to consider a simple example of an ensemble. Figure 1 shows four 2-cluster partitions of 12 two-dimensional data points. Correspondence problem is emphasized by different label systems used by the partitions. Table 1 shows the expected values of latent variables after 6 iterations of the EM algorithm and the resulting consensus clustering. In fact, stable combination appears even after the third iteration, and it corresponds to the true underlying structure of the data.

Our mixture model of consensus admits generalization for clustering ensembles with incomplete partitions. Such partitions can appear as a result of clustering of subsamples or resampling of a dataset. For example, a partition of a bootstrap sample only provides labels for the selected points. Therefore, the ensemble of such partitions is represented by a set of vectors of cluster labels with potentially missing components. Moreover, different vectors of cluster labels are likely to miss different components. Incomplete information can also arise when some clustering algorithms do not assign outliers to any of the clusters. Different clusterings in the diverse ensemble can consider the same point $\mathbf{x}_i$ as an outlier or otherwise, that results in missing components in the vector $\mathbf{y}_i$. Yet another scenario leading to missing information can occur in clustering combination of distributed data or ensemble of clusterings of non-identical replicas of a dataset.

It is possible to apply the EM algorithm in the case of missing data [14], namely missing cluster labels for some of the data points. In these situations, each vector $\mathbf{y}_i$ in $\boldsymbol{Y}$ can be split into observed and missing components $\mathbf{y}_i$ = ($\mathbf{y}_i^{\mathrm{obs}}$, $\mathbf{y}_i^{\mathrm{mis}}$). Incorporation of a missing data leads to a slight modification of the computation of E and M steps. First, the expected values $E[z_{im} |\mathbf{y}_i^{\mathrm{obs}}, \Theta' ]$ are now inferred from the observed components of vector $\mathbf{y}_i$, i.e. the products in Eq. (14) are taken over known labels:

$$\prod_{j=1}^{H} \rightarrow \prod_{j:y^{obs}} .$$

Additionally, it is required to compute the expected values $E[z_{im}\, \mathbf{y}_i^{\mathrm{mis}}| \mathbf{y}_i^{\mathrm{obs}},\Theta' ]$ and substitute them, as well as $E[z_{im} | \mathbf{y}_i^{\mathrm{obs}}, \Theta' ]$, in the M-step for re-estimation of parameters $\vartheta_{jm}(k)$. More details on handling missing data can be found in [14, 24].

Though data with missing cluster labels can be obtained in different ways, we analyze only the case when components of $\mathbf{y}_i$ are missing completely at random [29]. It means that the probability of a component to be missing does not depend on other observed or unobserved variables. Note, that the outcome of clustering of data subsamples (e.g., bootstrap) is different from clustering the entire data set and then deleting a random subset of labels. However, our goal is to present a consensus function for general settings. We expect that experimental results for ensembles with missing labels are applicable, at least qualitatively, even for a combination of bootstrap clusterings.

The proposed ensemble clustering based on mixture model consensus algorithm is summarized below:

```
begin
  for $i$=1 to $H$   // $H$ - number of clusterings
    cluster a dataset $\pi \leftarrow k$-means($\boldsymbol{X}$)
    add partition to the ensemble $\Pi = \{\Pi,\pi\}$
  end
  initialize model parameters $\Theta = \{\alpha_1,\ldots, \alpha_M, \boldsymbol{\theta}_1,\ldots, \boldsymbol{\theta}_M \}$
  do until convergence criterion is satisfied
    compute expected values $E[z_{im}]$, $i$=1..$N$, $m$=1..$M$
    compute $E[z_{im}\, \mathbf{y}_i^{\mathrm{mis}}]$ for missing data (if any)
    re-estimate parameters $\vartheta_{jm}(k)$, $j$=1..$H$, $m$=1..$M$, $\forall k$
  end
  $\pi_C(\mathbf{x}_i)$ = index of component of $\mathbf{z}_i$ with largest expected
   value,  $i$=1..$N$
  return $\pi_C$  // consensus partition
end
```

Note that any clustering algorithm can be used to generate ensemble instead of the $k$-means algorithm shown in the above pseudocode.

The value of $M$, number of components in the mixture, deserves a separate discussion that is beyond the scope of this paper. Here, we assume that the target number of clusters is predetermined. It should be noted, however, that mixture models in unsupervised classification greatly facilitate estimation of the true number of clusters [10]. Maximum likelihood formulation of the problem specifically allows us to estimate $M$ by using additional objective functions during the inference, such as the minimum description length of the model. In addition, the proposed consensus algorithm can be viewed as a version of Latent Class Analysis (e.g. see [3]), which has rigorous statistical means for quantifying plausibility of a candidate mixture model.

## 5 Empirical Study

The experiments were conducted with artificial and real-world datasets, where true natural clusters are known, to validate both accuracy and robustness of consensus via mixture model. We explored the datasets using five different consensus functions.

**5.1 Datasets.** Table 2 summarizes the details of the datasets. Five datasets have been used in the experiments. Two large real-world benchmarks: (i) The dataset of galaxies and stars, characterized by 14 features extracted from their images, with known classification provided by domain experts [26], (ii) Biochemical dataset of water molecules found in protein structures and categorized as

Table 2: Characteristics of the datasets.

| Dataset | No. of features | No. of classes | No. of points/class | Total no. of points | Av. *k*-means error (%) |
|---|---|---|---|---|---|
| Biochem. | 7 | 2 | 2138-3404 | 5542 | 47.4 |
| Galaxy | 14 | 2 | 2082-2110 | 4192 | 21.1 |
| 2-spirals | 2 | 2 | 100-100 | 200 | 43.5 |
| Half-rings | 2 | 2 | 100-300 | 400 | 25.6 |
| Iris | 4 | 3 | 50-50-50 | 150 | 15.1 |

either conserved or non-conserved type of molecules in the bound structure of proteins [1, 28]. Molecules are described by 8 physical and chemical features. The first feature, atomic density, was not used in the experiments because of its high correlation with atomic hydrophilicity. We also used two artificial datasets, “half-rings” and “2 spirals” shown on Figure 2, as well as classical Iris data from UCI benchmark repository.

We evaluated the performance of the evidence accumulation clustering algorithms by matching the detected and the known partitions of the datasets. The best possible matching of clusters provides a measure of performance expressed as the misassignment rate. To determine the clustering error one needs to solve the correspondence problem between the labels of known and derived clusters. The optimal correspondence can be obtained using the Hungarian method for minimal weight bipartite matching problem with $O(k^3)$ complexity for $k$ clusters.

**5.2 Selection of parameters and algorithms**. Accuracy of the EM algorithm has been compared to four other consensus functions:

1. CSPA for partitioning of hypergraphs induced from the co-association values. Its complexity is $O(N^2)$ that leads to severe computational limitations. We did not apply this algorithm to “Galaxy” and “Biochemical” data. For the same reason, we did not use other co-association methods, such as single-link. The performance of these methods was already analyzed in [11, 12, 31].
2. HGPA for hypergraph partitioning.
3. MCLA, that modifies HGPA via extended set of hyperedge operations and additional heuristics.
4. Quadratic mutual information (QMI) as described in [31].

First three methods (CSPA, HGPA and MCLA) were introduced in [30] and their code is available at http://www.strehl.com.

The $k$-means algorithm was used as a method of generating the partitions for the combination. Diversity of the partitions is ensured by the solutions obtained after random initialization of the algorithm. Two parameters of the clustering ensemble are especially important:

1. $H$ – the number of combined clusterings. We varied this value in the range [5..50].

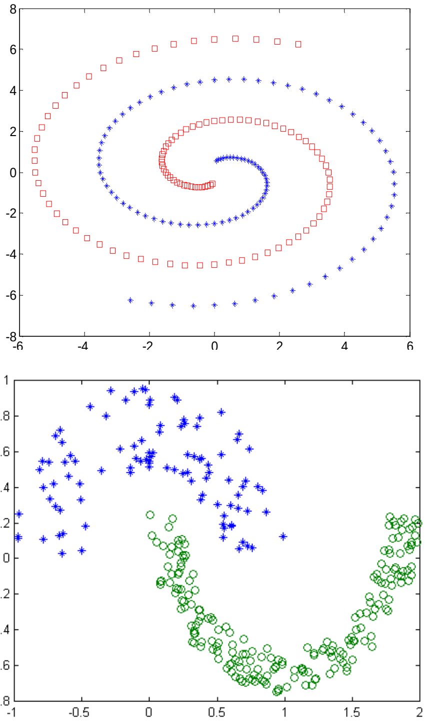

Figure 2: “2 spirals” and “Half-rings” datasets difficult for any centroid based clustering algorithms.

2. $k$ – the number of clusters in the component clusterings produced by $k$-means algorithm was taken in the range [2..10].

Both, the EM and QMI algorithm are susceptible to the presence of local minima of the objective functions. To reduce the risk of convergence to a lower quality solution, we used a simple heuristic afforded by low computational complexities of these algorithms. The final partition was picked from the results of three runs (with random initializations) according to the value of objective function. The highest value of the likelihood function served as a criterion for the EM algorithm.

**5.3 Experiments with complete partitions**. Only main results for each of the datasets are presented in Tables 3-7 due to space limitations. The tables report the mean error rate (%) of clustering combination from 10 independent runs for large biochemical and astronomical data and from 20 runs for other smaller datasets.

First observation is that none of the consensus functions is the absolute winner. Good performance was achieved by different combination algorithms across the values of parameters $k$ and $H$. The EM algorithm slightly outperforms other algorithms for ensembles of smaller size, while MCLA becomes stronger when number of clusterings $H > 20$. However, ensembles of very large size are less important in practice. All co-association methods are usually unreliable with number of clusterings $H < 50$ and this is where we position the proposed EM algorithm. Both, EM and QMI consensus functions need to estimate at least *kHM* parameters. Therefore, accuracy degradation will inevitably occur with increasing number of partitions when sample size is fixed. However, there was no noticeable decrease in the accuracy of the EM algorithm in current experiments. The EM algorithm also should benefit from the datasets of large size due to the improved reliability of model parameter estimation.

A valuable property of the EM consensus algorithm is its fast convergence rate. Mixture model parameter estimates nearly always converged in less than 10 iterations for all the datasets. Moreover, pattern assignments were typically settled in 4-6 iterations.

Clustering combination accuracy also depends on the number of clusters *M* in the ensemble partitions, or more precisely, on its ratio to the target number of clusters, i.e. *k*/M. For example, the EM algorithm worked best with *k*=3 for Iris dataset, *k*=3,4 for "Galaxy" dataset and *k*=2 for "Half-rings" data. These values of *k* are equal or slightly greater than the number of clusters in the combined partition. In contrast, accuracy of MCLA slightly improves with an increase in the number of clusters in the ensemble. Figure 3 shows the error as a function of *k* for different consensus functions for the galaxy data.

It is also interesting to note that, as expected, the average error of consensus clustering was lower than average error of the *k*-means clusterings in the ensemble (Table 1) when *k* is chosen to be equal to the true number of clusters. Moreover, the clustering error obtained by EM and MCLA algorithms with *k*=4 for "Biochemistry" data was the same as found by the advanced supervised classifiers applied to this dataset [28].

**5.4 Experiments with incomplete partitions**. This set of experiments focused on the dependence of clustering accuracy on the number of patterns with missing cluster labels. As before, an ensemble of partitions was generated using the *k*-means algorithm. Then, we randomly deleted cluster labels for a fixed number of patterns in each of the partitions. The EM consensus algorithm was used on such an ensemble. The number of missing labels in each partition was varied between [10%…50%] of the total number of patterns. The main results averaged over 10 independent runs are reported in Table 8 for "Galaxy" and "Biochemistry" datasets for various values of *H* and *k*. Also, a typical dependence of error on the number of patterns with missing data is shown for Iris data on Figure 4 (*H*=5, *k*=3).

One can note that combination accuracy decreases only insignificantly for biochemical data when up to 50% of labels are missing. This can be explained by the low inherent accuracy for this data, leaving little room for further degradation. For the "Galaxy" data, the accuracy drops by almost 10% when k=3,4. However, when just 10-20% of the cluster labels are missing, then there is just a small change in accuracy. Also, with different values of *k*, we see different sensitivity of the results to the missing labels. For example, with *k*=2, the accuracy drops by only slightly more than 1%. Ensembles of larger size *H*=10 suffered less from missing data than ensembles of size *H*=5.

## 6 Conclusion and Future Work

We have proposed a solution to the problem of clustering combination. A consensus clustering is derived from a solution of the maximum likelihood problem for a finite mixture model of the ensemble of partitions. Ensemble is modeled as a mixture of multivariate multinomial distributions in the space of cluster labels. Maximum likelihood problem is effectively solved using the EM algorithm. The EM-based consensus function is also capable of dealing with incomplete contributing partitions. Experimental results indicate good performance of the approach for several datasets and favorable comparison with other consensus functions. Among the advantages of the approach is its low computational complexity and well-grounded statistical model. This study can be extended in order to take into account non-independence of partitions in the ensemble. The presented consensus function is equivalent to a certain kind of Latent Class Analysis, which offers established statistical approaches to measure and use dependencies (at least pair-wise) between variables. It is also interesting to consider a combination of partitions of different quality. In this case one needs to develop a consensus function that weights the contributions of different partitions proportionally to their strength. We hope to address these issues in our future work.

Table 3: Mean error rate (%) for the "Galaxy" dataset.

| | | Type of Consensus Function | | | |
|---|---|---|---|---|---|
| *H* | *k* | **EM** | **QMI** | **HGPA** | **MCLA** |
| 5 | 2 | 18.9 | 19.0 | 50.0 | 18.9 |
| 5 | 3 | 11.6 | 13.0 | 50.0 | 13.5 |
| 5 | 4 | 11.3 | 13.0 | 50.0 | 11.7 |
| 5 | 5 | 13.9 | 18.0 | 50.0 | 14.3 |
| 5 | 7 | 14.5 | 21.9 | 50.0 | 15.6 |
| 5 | 10 | 13.4 | 31.1 | 50.0 | 15.4 |
| 10 | 2 | 18.8 | 18.8 | 50.0 | 18.8 |
| 10 | 3 | 14.9 | 15.0 | 50.0 | 14.8 |
| 10 | 4 | 11.6 | 11.1 | 50.0 | 12.0 |
| 10 | 5 | 14.5 | 13.0 | 50.0 | 13.6 |
| 15 | 2 | 18.8 | 18.8 | 50.0 | 18.8 |
| 15 | 3 | 14.0 | 13.3 | 50.0 | 14.8 |
| 15 | 4 | 11.7 | 11.5 | 50.0 | 11.6 |
| 15 | 5 | 12.9 | 11.5 | 50.0 | 12.9 |
| 20 | 2 | 18.8 | 18.9 | 50.0 | 18.8 |
| 20 | 3 | 12.8 | 11.7 | 50.0 | 14.3 |
| 20 | 4 | 11.0 | 10.8 | 50.0 | 11.5 |
| 20 | 5 | 16.2 | 12.1 | 50.0 | 12.3 |

Table 4: Mean error rate (%) for the “Biochemistry” dataset.

|  |  | Type of Consensus Function |  |  |
|---|---|---|---|---|
| *H* | *k* | **EM** | **QMI** | **MCLA** |
| 5 | 2 | 44.8 | 44.8 | 44.8 |
| 5 | 3 | 43.2 | 48.8 | 44.7 |
| 5 | 4 | 42.0 | 45.6 | 42.7 |
| 5 | 5 | 42.7 | 44.3 | 46.3 |
| 10 | 2 | 45.0 | 45.1 | 45.1 |
| 10 | 3 | 44.3 | 45.4 | 40.2 |
| 10 | 4 | 39.3 | 45.1 | 37.3 |
| 10 | 5 | 40.6 | 45.0 | 41.2 |
| 20 | 2 | 45.1 | 45.2 | 45.1 |
| 20 | 3 | 46.6 | 47.4 | 42.0 |
| 20 | 4 | 37.2 | 42.6 | 39.8 |
| 20 | 5 | 40.5 | 42.1 | 39.9 |
| 30 | 2 | 45.3 | 45.3 | 45.3 |
| 30 | 3 | 47.1 | 48.3 | 46.8 |
| 30 | 4 | 37.3 | 42.3 | 42.8 |
| 30 | 5 | 39.9 | 42.9 | 38.4 |
| 50 | 2 | 45.2 | 45.3 | 45.2 |
| 50 | 3 | 46.9 | 48.3 | 44.6 |
| 50 | 4 | 40.1 | 39.7 | 42.8 |
| 50 | 5 | 39.4 | 38.1 | 42.1 |

Table 5: Mean error rate (%) for the “Half-rings” dataset.

|  |  | Type of Consensus Function |  |  |  |  |
|---|---|---|---|---|---|---|
| *H* | *k* | **EM** | **QMI** | **CSPA** | **HGPA** | **MCLA** |
| 5 | 2 | 25.4 | 25.4 | 25.5 | 50.0 | 25.4 |
| 5 | 3 | 24.0 | 36.8 | 26.2 | 48.8 | 25.1 |
| 10 | 2 | 26.7 | 33.2 | 28.6 | 50.0 | 23.7 |
| 10 | 3 | 33.5 | 39.7 | 24.9 | 26.0 | 24.2 |
| 30 | 2 | 26.9 | 40.6 | 26.2 | 50.0 | 26.0 |
| 30 | 3 | 29.3 | 35.9 | 26.2 | 27.5 | 26.2 |
| 50 | 2 | 27.2 | 32.3 | 29.5 | 50.0 | 21.1 |
| 50 | 3 | 28.8 | 35.3 | 25.0 | 24.8 | 24.6 |

Table 6: Mean error rate (%) for the “2-spirals” dataset.

|  |  | Type of Consensus Function |  |  |  |  |
|---|---|---|---|---|---|---|
| *H* | *k* | **EM** | **QMI** | **CSPA** | **HGPA** | **MCLA** |
| 5 | 2 | 43.5 | 43.6 | 43.9 | 50.0 | 43.8 |
| 5 | 3 | 41.1 | 41.3 | 39.9 | 49.5 | 40.5 |
| 5 | 5 | 41.2 | 41.0 | 40.0 | 43.0 | 40.0 |
| 5 | 7 | 45.9 | 45.4 | 45.4 | 42.4 | 43.7 |
| 5 | 10 | 47.3 | 45.4 | 47.7 | 46.4 | 43.9 |
| 10 | 2 | 43.4 | 43.7 | 44.0 | 50.0 | 43.9 |
| 10 | 3 | 36.9 | 40.0 | 39.0 | 49.2 | 41.7 |
| 10 | 5 | 38.6 | 39.4 | 38.3 | 40.6 | 38.9 |
| 10 | 7 | 46.7 | 46.7 | 46.2 | 43.0 | 45.7 |
| 10 | 10 | 46.7 | 45.6 | 47.7 | 47.1 | 42.4 |
| 20 | 2 | 43.3 | 43.6 | 43.8 | 50.0 | 43.9 |
| 20 | 3 | 40.7 | 40.2 | 37.1 | 49.3 | 40.0 |
| 20 | 5 | 38.6 | 39.5 | 38.2 | 40.0 | 38.1 |
| 20 | 7 | 45.9 | 47.6 | 46.7 | 44.4 | 44.2 |
| 20 | 10 | 48.2 | 47.2 | 48.7 | 47.3 | 42.2 |

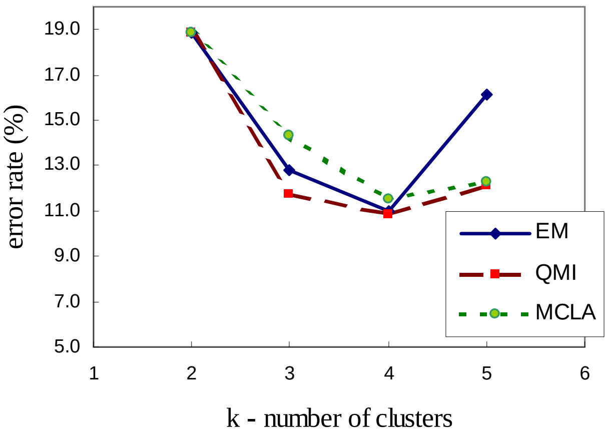

Figure 3: Consensus error as a function of the number of clusters in the contributing partitions for Galaxy data and ensemble size *H*=20.

Table 7: Mean error rate (%) for the Iris dataset.

|  |  | Type of Consensus Function |  |  |  |  |
|---|---|---|---|---|---|---|
| *H* | *k* | **EM** | **QMI** | **CSPA** | **HGPA** | **MCLA** |
| 5 | 3 | 11.0 | 14.7 | 11.2 | 41.4 | 10.9 |
| 10 | 3 | 10.8 | 10.8 | 11.3 | 38.2 | 10.9 |
| 15 | 3 | 10.9 | 11.9 | 9.8 | 42.8 | 11.1 |
| 20 | 3 | 10.9 | 14.5 | 9.8 | 39.1 | 10.9 |
| 30 | 3 | 10.9 | 12.8 | 7.9 | 43.4 | 11.3 |
| 40 | 3 | 11.0 | 12.4 | 7.7 | 41.9 | 11.1 |
| 50 | 3 | 10.9 | 13.8 | 7.9 | 42.7 | 11.2 |

Table 8: Clustering error rate as a function of the number of missing labels for the large datasets

|  |  | Missing | "Galaxy" | "Biochem." |
|---|---|---|---|---|
| *H* | *k* | labels (%) | error (%) | error (%) |
| 5 | 2 | 10 | 18.81 | 45.18 |
| 5 | 2 | 20 | 18.94 | 44.73 |
| 5 | 2 | 30 | 19.05 | 45.08 |
| 5 | 2 | 40 | 19.44 | 45.64 |
| 5 | 2 | 50 | 19.86 | 46.23 |
| 5 | 3 | 10 | 12.95 | 43.79 |
| 5 | 3 | 20 | 13.78 | 43.89 |
| 5 | 3 | 30 | 14.92 | 45.67 |
| 5 | 3 | 40 | 19.58 | 47.88 |
| 5 | 3 | 50 | 23.31 | 48.41 |
| 5 | 4 | 10 | 11.56 | 43.10 |
| 5 | 4 | 20 | 11.98 | 43.59 |
| 5 | 4 | 30 | 14.36 | 44.50 |
| 5 | 4 | 40 | 17.34 | 45.12 |
| 5 | 4 | 50 | 24.47 | 45.62 |
| 10 | 2 | 10 | 18.87 | 45.14 |
| 10 | 2 | 20 | 18.85 | 45.26 |
| 10 | 2 | 30 | 18.86 | 45.28 |
| 10 | 2 | 40 | 18.93 | 45.13 |
| 10 | 2 | 50 | 19.85 | 45.35 |
| 10 | 3 | 10 | 13.44 | 44.97 |
| 10 | 3 | 20 | 14.46 | 45.20 |
| 10 | 3 | 30 | 14.69 | 47.91 |
| 10 | 3 | 40 | 14.40 | 47.21 |
| 10 | 3 | 50 | 15.65 | 46.92 |
| 10 | 4 | 10 | 11.06 | 39.15 |
| 10 | 4 | 20 | 11.17 | 37.81 |
| 10 | 4 | 30 | 11.32 | 40.41 |
| 10 | 4 | 40 | 15.07 | 37.78 |
| 10 | 4 | 50 | 16.46 | 41.56 |

## References

[1] E. E. Abola, F. C. Bernstein, S. H. Bryant, T. F. Koetzle, and J. Weng, *Protein Data Bank*, In Crystallographic Databases–Information Content, Software Systems, Scientific Applications (F. Allen et al. eds), Data Commission of the International Union of Crystallography, Bonn/Cambridge/Chester, 107–132, 1987.

[2] J.-P. Barthelemy and B. Leclerc, *The median procedure for partition*, In Partitioning Data Sets, I.J. Cox et al eds., AMS DIMACS Series in Discrete Mathematics, 19: 3-34, 1995.

[3] D. J. Bartholomew and M. Knott, *Latent variable models and factor analysis*, 2nd ed, Kendall's Library of Statistics 7. London: Arnold, 1999.

[4] A. Ben-Hur, D. Horn, H.T. Siegelmann, and V.Vapnik, *Support vector clustering*, Journal of Machine Learning Research, 2:125-137, 2001.

[5] E. Dimitriadou, A. Weingessel and K. Hornik, *Voting-merging: An ensemble method for clustering*, In Proc. Int. Conf. on Artificial Neural Networks, Vienna, 217-224, 2001.

[6] 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 B, 39: 1-22, 1997.

[7] P. Domingos and M. Pazzani, *On the optimality of the simple Bayesian classifier under zero-one loss*, Machine Learning, 29: 103–130, 1997.

[8] S. Dudoit and J. Fridlyand, *Bagging to improve the accuracy of a clustering procedure*, Bioinformatics, 19 (9): 1090-1099, 2003

[9] B. Fischer, J.M. Buhmann, *Path-Based Clustering for Grouping of Smooth Curves and Texture Segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 25 (4): 513-518, 2003.

[10] Figueiredo M., Jain, A.K.: Unsupervised learning of finite mixture models. IEEE Transaction on Pattern Analysis and Machine Intelligence 24:381--396, 2002.

[11] A.L.N. Fred, *Finding Consistent Clusters in Data Partitions*, In Proc. 3d Int. Workshop on Multiple Classifier Systems. Eds. F. Roli, J. Kittler, LNCS 2364: 309-318, 2001.

[12] A.L.N. Fred and A.K. Jain, *Data Clustering using Evidence Accumulation*, In Proc. of the 16th International Conference on Pattern Recognition, ICPR 2002 ,Quebec City: 276 – 280, 2002.

[13] W. Gablentz, M. Köppen, and E. Dimitriadou, *Robust Clustering by Evolutionary Computation*, In Proc. 5th Online World Conf. on Soft Computing in Industrial Applications (WSC5), 2000.

[14] Z. Ghahramani, and M. Jordan, *Supervised learning from incomplete data via an EM approach*, In Proceedings of Advances in Neural Information Processing Systems (NIPS 6): 120-127, 1993.

[15] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall Inc., New Jersey, 1988.

[16] A.K. Jain, M. N. Murty, and P. Flynn, *Data clustering: A review*, ACM Computing Surveys, 31(3): 264–323, 1999.

[17] E. Johnson and H. Kargupta, *Collective, hierarchical clustering from distributed, heterogeneous data*, In Large-Scale Parallel KDD Systems. Eds. Zaki M. and Ho C., Volume 1759 of LNCS, Springer-Verlag, 221–244, 1999.

[18] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, *Multilevel Hypergraph Partitioning: Applications in VLSI Design*, In Proc. ACM/IEEE Design Automation Conference, 526-529, 1997

[19] G. Karypis and V. Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM Journal of Scientific Computing, 20(1): 359-392,1998.

[20] P. Kellam, X. Liu, N.J. Martin, C. Orengo, S. Swift, and A.Tucker, *Comparing, contrasting and combining clusters in viral gene expression data*, Proceedings of 6th Workshop on Intelligent Data Analysis in Medicine and Pharmocology, 56-62, 2001.

[21] J. Kleinberg, *An Impossibility Theorem for Clustering*, In Proceedings of Advances in Neural Information Processing Systems (NIPS 2002) 15, 2002.

[22] P. Langley, W. Iba, and K. Thompson, *An analysis of Bayesian classifiers*, In Proceedings of the Tenth National Conference on Artificial Intelligence, San Jose, CA, AAAI Press, 399–406, 1992.

[23] F. Leisch, *Bagged clustering*, Working Papers SFB "Adaptive Information Systems and Modeling in Economics and Management Science", no.51, Aug. 1999, Institut für Information, Abt. Produktionsmanagement, Wien, Wirtschaftsuniv, 1999.

[24] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, Wiley, New York, 1997.

[25] B. Minaei, A. Topchy, and W. Punch, *Ensembles of Partitions via Data Resampling*, In Proc. International Conference on Information Technology, ITCC 2004, Las Vegas, NV, April 2004, in press.

[26] S.C. Odewahn, E.B. Stockwell, R.L. Pennington, R.M. Humphreys, and W.A. Zumach, *Automated Star/Galaxy Discrimination with Neural Networks*, Astronomical Journal, 103: 308-331, 1992.

[27] B.H. Park and H. Kargupta, *Distributed Data Mining*, In The Handbook of Data Mining, Ed. Nong Ye, Lawrence Erlbaum Associates, 2003.

[28] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, *Dimensionality reduction using genetic algorithms*, IEEE Transactions on Evolutionary Computation, 4(2): 164 - 171, 2000.

[29] D.B. Rubin, *Inference with Missing Data*, Biometrika, 63: 581-592, 1976.

[30] A. Strehl and J. Ghosh, *Cluster ensembles - a knowledge reuse framework for combining multiple partitions*, Journal of Machine Learning Research, 3: 583-617, 2002.

[31] A. Topchy, A.K. Jain, and W. Punch, *Combining Multiple Weak Clusterings*, In Proc. IEEE Intl. Conf. on Data Mining, Melbourne, FL, 331-338, 2003.