

# Clustering Ensembles: Models of Consensus and Weak Partitions\*

Alexander Topchy, Anil K. Jain, and William Punch

Department of Computer Science and Engineering, Michigan State University

East Lansing, Michigan, 48824, USA

{topchyal, jain, punch}@cse.msu.edu

**Abstract.** Clustering ensembles have emerged as a powerful method for improving both the robustness as well as 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. This study extends previous research on clustering ensembles in several respects. First, we introduce a unified representation for multiple clusterings and formulate the corresponding categorical clustering problem. Second, we propose 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. Third, we define a new consensus function that is related to the classical intra-class variance criterion using the generalized mutual information definition. Finally, we demonstrate the efficacy of combining partitions generated by weak clustering algorithms that use data projections and random data splits. A simple explanatory model is offered for the behavior of combinations of such weak clustering components. Combination accuracy is analyzed as a function of several parameters that control the power and resolution of component partitions as well as the number of partitions. 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 methods on several real-world datasets.

**KEYWORDS:** clustering, ensembles, multiple classifier systems, consensus function, mutual information

\* This research was supported by ONR grant N00014-01-1-0266. Parts of this work have been presented at the IEEE International Conference on Data Mining, ICDM 03, Melbourne, Florida, November 2003 and SIAM International Conference on Data Mining, SDM 04, Florida, April 2004.

# 1 Introduction

In contrast to supervised classification, clustering is inherently an ill-posed problem, whose solution violates at least one of the common assumptions about scale-invariance, richness, and cluster consistency [33]. Different clustering solutions may seem equally plausible without a priori knowledge about the underlying data distributions. Every clustering algorithm implicitly or explicitly assumes a certain data model, and it may produce erroneous or meaningless results when these assumptions are not satisfied by the sample data. Thus the availability of prior information about the data domain is crucial for successful clustering, though such information can be hard to obtain, even from experts. Identification of relevant subspaces [2] or visualization [24] may help to establish the sample data's conformity to the underlying distributions or, at least, to the proper number of clusters.

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. Fusion of clusterings using multiple

sources of data or features becomes increasingly important in distributed data mining, e.g., see review in [41]. Several recent independent studies [10, 12, 14, 15, 43, 47] have pioneered clustering ensembles as a new branch in the conventional taxonomy of clustering algorithms [26, 27]. Please see the Appendix for detailed review of the related work, including [7, 11, 16, 19, 28, 31, 35].

The problem of clustering combination can be defined generally as follows: given multiple clusterings of the data set, find a combined clustering with better quality. While the problem of clustering combination bears some traits of a classical clustering problem, it also has three major issues which are specific to combination design:

1. Consensus function: How to combine different clusterings? How to resolve the label correspondence problem? How to ensure symmetrical and unbiased consensus with respect to all the component partitions?
2. Diversity of clustering: How to generate different partitions? What is the source of diversity in the components?
3. Strength of constituents/components: How “weak” could each input partition be? What is the minimal complexity of component clusterings to ensure a successful combination?

Similar questions have already been addressed in the framework of multiple classifier systems. Combining results from many supervised classifiers is an active research area (Quinlan 96, Breiman 98) and it provides the main motivation for clusterings combination. However, it is not possible to mechanically apply the combination algorithms from classification (supervised) domain to clustering (unsupervised) domain. Indeed, no labeled training data is available in clustering; therefore the ground truth feedback necessary for boosting the overall accuracy cannot be used. In addition, different clusterings may produce incompatible data labelings, resulting in intractable correspondence problems, especially when the numbers of clusters are different. Still, the supervised classifier combination demonstrates, in principle, how multiple solutions reduce the variance component of the expected error rate and increase the robustness of the solution.

From the supervised case we also learn that the proper combination of weak classifiers [32, 25, 18, 6] may achieve arbitrarily low error rates on training data, as well as reduce the predictive error. One can expect that using many simple, but computationally inexpensive components will be preferred to combining clusterings obtained by sophisticated, but computationally involved algorithms.

This paper further advances ensemble methods in several aspects, namely, design of new effective consensus functions, development of new partition generation mechanisms and study of the resulting clustering accuracy.

## 1.1 Our Contribution

We offer a representation of multiple clusterings as a set of new attributes characterizing the data items. Such a view directly leads to a formulation of the combination problem as a categorical clustering problem in the space of these attributes, or, in other terms, a median partition problem. Median partition can be viewed as the best summary of the given input partitions. As an optimization problem, median partition is NP-complete [3], with a continuum of heuristics for an approximate solution.

This work focuses on the primary problem of clustering ensembles, namely the consensus function, which creates the combined clustering. We show how median partition is related to the classical intra-class variance criterion when generalized mutual information is used as the evaluation function. Consensus function based on quadratic mutual information (QMI) is proposed and reduced to the  $k$ -means clustering in the space of specially transformed cluster labels.

We also propose a new fusion method for unsupervised decisions that is based on a probability model of the consensus 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 [8].

There are several advantages to QMI and EM consensus functions. These include: (i) complete avoidance of solving the label correspondence problem, (ii) low computational complexity, and (iii) ability to handle missing data, i.e. missing cluster labels for certain patterns in the ensemble (for example, when bootstrap method is used to generate the ensemble).

Another goal of our work is to adopt weak clustering algorithms and combine their outputs. Vaguely defined, a weak clustering algorithm produces a partition, which is only slightly better than a random partition of the data. We propose two different weak clustering algorithms as the component generation mechanisms:

1. Clustering of random 1-dimensional projections of multidimensional data. This can be generalized to clustering in any random subspace of the original data space.
2. Clustering by splitting the data using a number of random hyperplanes. For example, if only one hyperplane is used then data is split into two groups.

Finally, this paper compares the performance of different consensus functions. We have investigated the performance of a family of consensus functions based on categorical clustering including the co-association based hierarchical methods [15, 16, 17], hypergraph algorithms [47, 29, 30] and our new consensus functions. Combination accuracy is analyzed as a function of the number and the resolution of the clustering components. In addition, we study clustering performance when some cluster labels are missing, which is often encountered in the distributed data or re-sampling scenarios.

## **2 Representation of Multiple Partitions**

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  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  and a set of  $H$  partitions  $\Pi = \{\pi_1, \dots, \pi_H\}$  of objects in  $X$ . Different partitions of  $X$  return a set of labels for each point  $\mathbf{x}_i$ ,  $i=1, \dots, N$ :

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

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. Also no assumptions are needed at the moment about the data input: it could be represented in a non-metric space or as an  $N \times N$  dissimilarity matrix. 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  $X$  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  $X$ , but only from their labels  $Y$  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 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  $\{\pi_1, \dots, \pi_H\}$ , without the original features of data  $X$ .

Hence the problem of combining partitions can be transformed to a categorical clustering problem. Such a view gives insight into the properties of the expected combination, which can be inferred through various statistical and information-theoretic techniques. In particular, one can estimate the sensitivity of the combination to the correlation of components (features) as well as analyze various sample size issues. Perhaps the main advantage of this representation is that it facilitates the use of known algorithms for categorical clustering [37, 48] and allows one to design new consensus heuristics in a transparent way. The extended representation of data  $X$  can be illustrated by a table with  $N$  rows and  $(d+H)$  columns:

|                | $\pi_1$  | ... | $\pi_H$  |              |     |              |
|----------------|----------|-----|----------|--------------|-----|--------------|
| $\mathbf{x}_1$ | $x_{11}$ | ... | $x_{1d}$ | $\pi_1(x_1)$ | ... | $\pi_H(x_1)$ |
| $\mathbf{x}_2$ | $x_{21}$ | ... | $x_{2d}$ | $\pi_1(x_2)$ | ... | $\pi_H(x_2)$ |
| ...            | ...      | ... | ...      | ...          | ... | ...          |
| $\mathbf{x}_N$ | $x_{N1}$ | ... | $x_{Nd}$ | $\pi_1(x_N)$ | ... | $\pi_H(x_N)$ |

$\underbrace{\hspace{10em}}$   
Original  $d$  features

$\underbrace{\hspace{10em}}$   
"New"  $H$  features

The consensus clustering is found as a partition  $\pi_C$  of a set of vectors  $\mathbf{Y} = \{\mathbf{y}_i\}$  that directly translates to the partition of the underlying data points  $\{\mathbf{x}_i\}$ .

### 3 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  $\mathbf{y}=\boldsymbol{\pi}(\mathbf{x})$  of the pattern/object  $\mathbf{x}$ . 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:

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

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  $\mathbf{Y}=\{\mathbf{y}_i\}_{i=1}^N$  are assumed to be independent and identically distributed. This allows one to represent the log likelihood function for the parameters  $\Theta=\{\alpha_1,\dots,\alpha_M,\theta_1,\dots,\theta_M\}$  given the data set  $\mathbf{Y}$  as:

$$\log L(\Theta|\mathbf{Y})=\log\prod_{i=1}^N P(\mathbf{y}_i|\Theta)=\sum_{i=1}^N\log\sum_{m=1}^M\alpha_m P_m(\mathbf{y}_i|\theta_m). \quad (3)$$

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  $\mathbf{Y}$ , we must maximize the likelihood function with respect to the unknown parameters  $\Theta$ :

$$\Theta^*=\arg\max_{\Theta}\log L(\Theta|\mathbf{Y}). \quad (4)$$

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  $\mathbf{X}$  has been transformed, with the help of multiple clustering algorithms, to a space of new multivariate features  $\mathbf{y}=\boldsymbol{\pi}(\mathbf{x})$ . 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:

$$P_m(\mathbf{y}_i|\theta_m)=\prod_{j=1}^H P_m^{(j)}(y_{ij}|\theta_m^{(j)}). \quad (5)$$

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 [34]. Our ultimate goal is to make a discrete label assignment to the data in  $\mathbf{X}$  through an indirect route of density estimation of  $\mathbf{Y}$ . 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  $P(\mathbf{y}_i|\Theta)$ , as supported by the analysis of naïve Bayes classifier in [9].



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:

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

Here, without the loss of generality, the labels of the clusters in  $\pi_j$  are chosen to be integers in  $\{1, \dots, K(j)\}$ . 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:

$$\sum_{k=1}^{K(j)} \vartheta_{jm}^{(k)} = 1, \forall j \in \{1, \dots, H\}, \forall m \in \{1, \dots, M\}. \quad (7)$$

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:

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

The maximum likelihood problem in Eq. (3) generally cannot be solved in a closed form when all the parameters  $\Theta = \{\alpha_1, \dots, \alpha_M, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_M\}$  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  $\mathbf{Z}$  and the likelihood of complete data  $(\mathbf{Y}, \mathbf{Z})$ . 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  $\mathbf{y}_i$ . The detailed derivation of the EM solution to the mixture model with multivariate, multinomial components is given in the Appendix. Here we give only the equations for the E- and M-steps which are repeated at each iteration of the algorithm:

$$E[z_{im}] = \frac{\alpha'_m \prod_{j=1}^H \prod_{k=1}^{K(j)} (\vartheta'_{jm}(k))^{\delta^{(y_{ij}, k)}}}{\sum_{n=1}^M \alpha'_n \prod_{j=1}^H \prod_{k=1}^{K(j)} (\vartheta'_{jn}(k))^{\delta^{(y_{ij}, k)}}}. \quad (9)$$

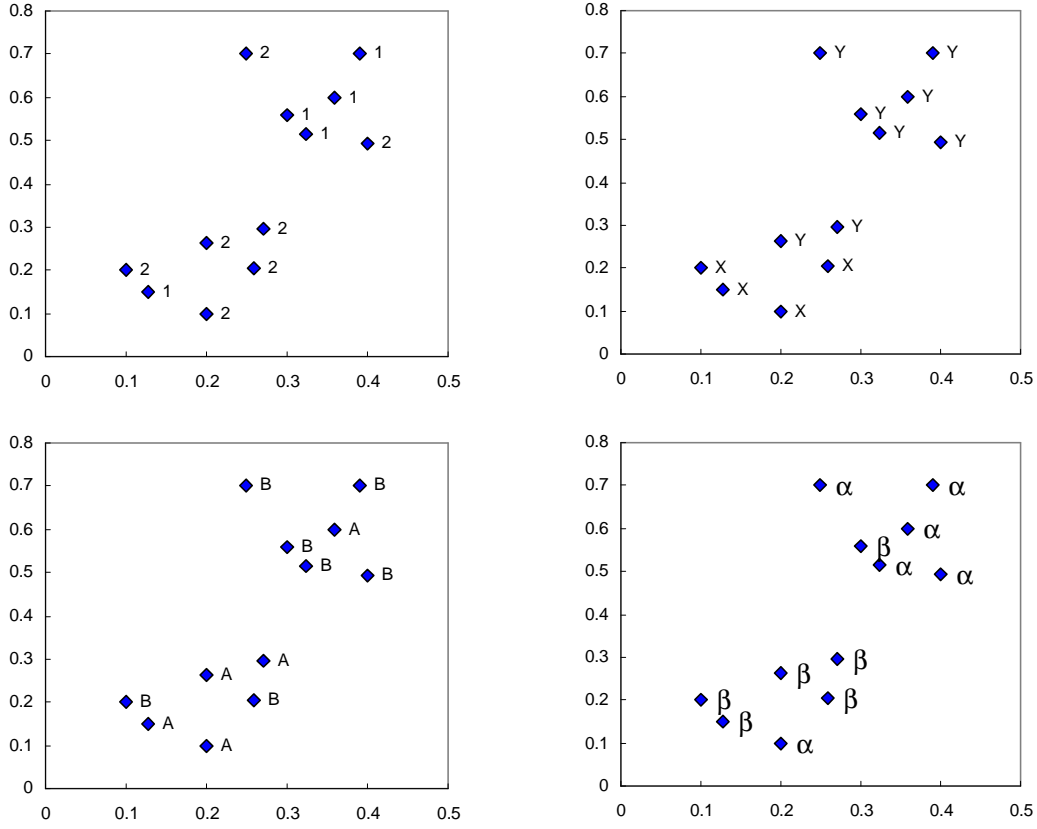
$$\alpha_m = \frac{\sum_{i=1}^N E[z_{im}]}{\sum_{i=1}^N \sum_{m=1}^M E[z_{im}]} . \quad (10)$$

$$\vartheta_{jm}(k) = \frac{\sum_{i=1}^N \delta(y_{ij}, k) E[z_{im}]}{\sum_{i=1}^N \sum_{k=1}^{K(j)} \delta(y_{ij}, k) E[z_{im}]} . \quad (11)$$

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  $\mathbf{y}_i$  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  $\mathbf{z}_i$ .

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, a stable combination appears as early as 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  $\mathbf{y}_i$ .



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

Table 1: Clustering ensemble and consensus solution

|          | $\pi_1$ | $\pi_2$ | $\pi_3$ | $\pi_4$  | $E[z_{i1}]$ | $E[z_{i2}]$ | Consensus |
|----------|---------|---------|---------|----------|-------------|-------------|-----------|
| $y_1$    | 2       | B       | X       | $\beta$  | 0.999       | 0.001       | 1         |
| $y_2$    | 2       | A       | X       | $\alpha$ | 0.997       | 0.003       | 1         |
| $y_3$    | 2       | A       | Y       | $\beta$  | 0.943       | 0.057       | 1         |
| $y_4$    | 2       | B       | X       | $\beta$  | 0.999       | 0.001       | 1         |
| $y_5$    | 1       | A       | X       | $\beta$  | 0.999       | 0.001       | 1         |
| $y_6$    | 2       | A       | Y       | $\beta$  | 0.943       | 0.057       | 1         |
| $y_7$    | 2       | B       | Y       | $\alpha$ | 0.124       | 0.876       | 2         |
| $y_8$    | 1       | B       | Y       | $\alpha$ | 0.019       | 0.981       | 2         |
| $y_9$    | 1       | B       | Y       | $\beta$  | 0.260       | 0.740       | 2         |
| $y_{10}$ | 1       | A       | Y       | $\alpha$ | 0.115       | 0.885       | 2         |
| $y_{11}$ | 2       | B       | Y       | $\alpha$ | 0.124       | 0.876       | 2         |
| $y_{12}$ | 1       | B       | Y       | $\alpha$ | 0.019       | 0.981       | 2         |

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 [20], namely missing cluster labels for some of the data points. In these situations, each vector  $\mathbf{y}_i$  in  $\mathbf{Y}$  can be split into observed and missing components  $\mathbf{y}_i = (\mathbf{y}_i^{\text{obs}}, \mathbf{y}_i^{\text{mis}})$ . Incorporation of a missing data leads to a slight

modification of the computation of E and M steps. First, the expected values  $E[z_{im} | \mathbf{y}_i^{\text{obs}}, \Theta']$  are now inferred from the observed components of vector  $\mathbf{y}_i$ , i.e. the products in Eq. (9) are taken over known labels:  $\prod_{j=1}^H \rightarrow \prod_{j:\mathbf{y}^{\text{obs}}}$ . Additionally, one must compute the expected values  $E[z_{im} \mathbf{y}_i^{\text{mis}} | \mathbf{y}_i^{\text{obs}}, \Theta']$  and substitute them, as well as  $E[z_{im} | \mathbf{y}_i^{\text{obs}}, \Theta']$ , in the M-step for re-estimation of parameters  $\vartheta_{jm}(k)$ . More details on handling missing data can be found in [20].

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 [46]. 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. Note that any clustering algorithm can be used to generate ensemble instead of the k-means algorithm shown in this pseudocode:

---

**begin**

**for**  $i=1$  **to**  $H$  //  $H$  - number of clusterings

    cluster a dataset  $X$ :  $\pi \leftarrow k\text{-means}(X)$

    add partition  $\pi$  to the ensemble  $\Pi = \{\Pi, \pi\}$

**end**

initialize model parameters  $\Theta = \{\alpha_1, \dots, \alpha_M, \theta_1, \dots, \theta_M\}$

**do until** convergence criterion is satisfied

    compute expected values  $E[z_{im}], i=1..N, m=1..M$

    compute  $E[z_{im} \mathbf{y}_i^{\text{mis}}]$  for missing data (if any)

    re-estimate parameters  $\vartheta_{jm}(k), j=1..H, m=1..M, \forall k$

**end**

$\pi_C(\mathbf{x}_i) = \text{index of component of } \mathbf{z}_i \text{ with the largest expected value, } i=1..N$

```
    return  $\pi_c$  // consensus partition
end
```

---

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 model in unsupervised classification greatly facilitates estimation of the true number of clusters [13]. 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 [4]), which has rigorous statistical means for quantifying plausibility of a candidate mixture model.

Whereas the finite mixture model may not be valid for the patterns in the 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 [5] seeks spherical clusters after the kernel transformation that corresponds to more complex cluster shapes in the original pattern space.

## 4 Information-Theoretic Consensus of Clusterings

Another candidate consensus function is based on the notion of median partition. A median partition  $\sigma$  is the best summary of existing partitions in  $\Pi$ . In contrast to the co-association approach, median partition is derived from estimates of similarities between attributes<sup>1</sup> (i.e., partitions in  $\Pi$ ), rather than from similarities between objects. A well-known example of this approach is implemented in the COBWEB algorithm in the context of conceptual clustering [48]. COBWEB clustering criterion estimates the partition utility, which is the sum of category utility functions introduced by Gluck and Corter [21]. In our terms, the category utility function  $U(\sigma, \pi_i)$  evaluates the quality of a

---

<sup>1</sup> Here “attributes” (features) refer to the partitions of an ensemble, while the objects refer the original data points.

candidate median partition  $\pi_C = \{C_1, \dots, C_K\}$  against some other partition  $\pi_i = \{L_1^i, \dots, L_{K(i)}^i\}$ , with labels  $L_j^i$  for  $j$ -th cluster:

$$U(\pi_C, \pi_i) = \sum_{r=1}^K p(C_r) \sum_{j=1}^{K(i)} p(L_j^i | C_r)^2 - \sum_{j=1}^{K(i)} p(L_j^i)^2, \quad (12)$$

with the following notations:  $p(C_r) = |C_r| / N$ ,  $p(L_j^i) = |L_j^i| / N$ , and  $p(L_j^i | C_r) = |L_j^i \cap C_r| / |C_r|$ .

The function  $U(\pi_C, \pi_i)$  assesses the agreement between two partitions as the difference between the expected number of labels of partition  $\pi_i$  that can be correctly predicted both with the knowledge of clustering  $\pi_C$  and without it. The category utility function can also be written as Goodman-Kruskal index for the contingency table between two partitions [22, 39]. The overall utility of the partition  $\pi_C$  with respect to all the partitions in  $\Pi$  can be measured as the sum of pair-wise agreements:

$$U(\pi_C, \Pi) = \sum_{i=1}^H U(\pi_C, \pi_i). \quad (13)$$

Therefore, the best median partition should maximize the value of overall utility:

$$\pi_C^{\text{best}} = \arg \max_{\pi_C} U(\pi_C, \Pi). \quad (14)$$

Importantly, Mirkin [39] has proved that maximization of partition utility in Eq. (13) is equivalent to minimization of the square-error clustering criterion if the number of clusters  $K$  in target partition  $\pi_C$  is fixed. This is somewhat surprising in that the partition utility function in Eq. (14) uses only the between-attribute similarity measure of Eq.(12), while square-error criterion makes use of distances between objects and prototypes. Simple standardization of categorical labels in  $\{\pi_1, \dots, \pi_H\}$  effectively transforms them to quantitative features [39]. This allows us to compute real-valued distances and cluster centers. This transformation replaces the  $i$ -th partition  $\pi_i$  assuming  $K(i)$  values by  $K(i)$  binary features, and standardizes each binary feature to a zero mean. In other words, for each object  $x$  we can compute the values of the new features  $\tilde{y}_{ij}(x)$ , as following:

$$\tilde{y}_{ij}(x) = \mathcal{D}(L_j^i, \pi_i(x)) - p(L_j^i), \text{ for } j=1 \dots K(i), \ i=1 \dots H. \quad (15)$$

Hence, the solution of median partition problem in Eq. (4) can be approached by  $k$ -means clustering algorithm operating in the space of features  $\tilde{y}_{ij}$  if the number of target clusters is predetermined. We use this heuristic as a part of empirical study of consensus functions.

Let us consider the information-theoretic approach to the median partition problem. In this framework, the quality of the consensus partition  $\pi_C$  is determined by the amount of information  $I(\pi_C, \Pi)$  it shares with the given partitions in  $\Pi$ . Strehl and Ghosh [47] suggest an objective function that is based on the classical Shannon definition of mutual information:

$$\pi_C^{\text{best}} = \arg \max_{\pi_C} I(\pi_C, \Pi), \text{ where } I(\pi_C, \Pi) = \sum_{i=1}^H I(\pi_C, \pi_i), \quad (16)$$

$$I(\pi_C, \pi_i) = \sum_{r=1}^K \sum_{j=1}^{K(i)} p(C_r, L_j^i) \log \left( \frac{p(C_r, L_j^i)}{p(C_r)p(L_j^i)} \right). \quad (17)$$

Again, an optimal median partition can be found by solving this optimization problem. However, it is not clear how to directly use these equations in a search for consensus.

We show that another information-theoretic definition of entropy will reduce the mutual information criterion to the category utility function discussed before. We proceed from the generalized entropy of degree  $s$  for a discrete probability distribution  $P=(p_1, \dots, p_n)$  [23]:

$$H^s(P) = (2^{1-s} - 1)^{-1} \left( \sum_{i=1}^n p_i^s - 1 \right), \quad s > 0, \quad s \neq 1 \quad (18)$$

Shannon's entropy is the limit form of Eq.(18):

$$\lim_{s \rightarrow 1} H^s(P) = - \sum_{i=1}^n p_i \log_2 p_i. \quad (19)$$

Generalized mutual information between  $\sigma$  and  $\pi$  can be defined as:

$$I^s(\pi, \pi_C) = H^s(\pi) - H^s(\pi | \pi_C). \quad (20)$$

Quadratic entropy ( $s=2$ ) is of particular interest, since it is known to be closely related to classification error, when used in the probabilistic measure of inter-class distance. When  $s=2$ , generalized mutual information  $I(\pi_C, \pi_i)$  becomes:

$$\begin{aligned}
I^2(\pi_c, \pi_i) &= -2 \left( \sum_{j=1}^{K(i)} p(L_j^i)^2 - 1 \right) + 2 \sum_{r=1}^K p(C_r) \left( \sum_{j=1}^{K(i)} p(L_j^i | C_r)^2 - 1 \right) = \\
&= 2 \sum_{r=1}^K p(C_r) \sum_{j=1}^{K(i)} p(L_j^i | C_r)^2 - 2 \sum_{j=1}^{K(i)} p(L_j^i)^2 = 2U(\pi_c, \pi_i).
\end{aligned} \tag{21}$$

Therefore, generalized mutual information gives the same consensus clustering criterion as category utility function in Eq. (13). Moreover, traditional Gini-index measure for attribute selection also follows from Eqs. (12) and (21). In light of Mirkin's result, all these criteria are equivalent to within-cluster variance minimization, after simple label transformation. Quadratic mutual information, mixture model and other interesting consensus functions have been used in our comparative empirical study.

## 5 Combination of Weak Clusterings

The previous sections addressed the problem of clusterings combination, namely how to formulate the consensus function regardless of the nature of individual partitions in the combination. We now turn to the issue of generating different clusterings for the combination. There are several principal questions. Do we use the partitions produced by numerous clustering algorithms available in the literature? Can we relax the requirements for the clustering components? There are several existing methods to provide diverse partitions:

1. Use different clustering algorithms, e.g.  $k$ -means, mixture of Gaussians, spectral, single-link, etc. [47].
2. Exploit built-in randomness or different parameters of some algorithms, e.g. initializations and various values of  $k$  in  $k$ -means algorithm [35, 15, 16].
3. Use many subsamples of the data set, such as bootstrap samples [10, 38].

These methods rely on the clustering algorithms, which are powerful on their own, and as such are computationally involved. We argue that it is possible to generate the partitions using weak, but less expensive, clustering algorithms and still achieve comparable or better performance. Certainly, the



key motivation is that the synergy of many such components will compensate for their weaknesses.

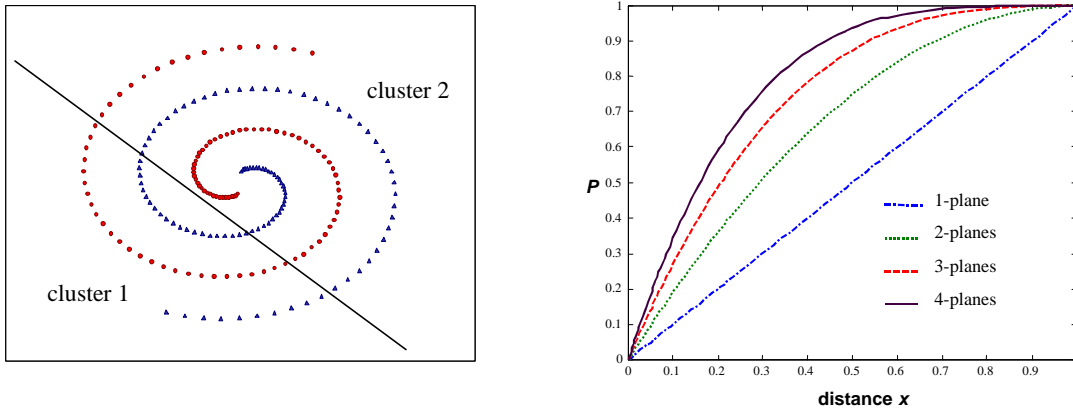
We consider two simple clustering algorithms:

1. Clustering of the data projected to a random subspace. In the simplest case, the data is projected on 1-dimensional subspace, a random line. The  $k$ -means algorithm clusters the projected data and gives a partition for the combination.
2. Random splitting of data by hyperplanes. For example, a single random hyperplane would create a rather trivial clustering of  $d$ -dimensional data by cutting the hypervolume into two regions.

We will show that both approaches are capable of producing high quality consensus clusterings in conjunction with a proper consensus function.

## 5.1 Splitting by Random Hyperplanes

Direct clustering by use of a random hyperplane illustrates how a reliable consensus emerges from low-informative components. The random splits approach pushes the notion of weak clustering almost to an extreme. The data set is cut by random hyperplanes dissecting the original volume of  $d$ -dimensional space containing the points. Points separated by the hyperplanes are declared to be in different clusters. Hence, the output clusters are convex. In this situation, a co-association consensus function is appropriate since the only information needed is whether the patterns are in the same cluster or not. Thus the contribution of a hyperplane partition to the co-association value for any pair of objects can be either 0 or 1. Finer resolutions of distance are possible by counting the number of hyperplanes separating the objects, but for simplicity we do not use it here. Consider a random line dissecting the classic 2-spiral data shown in Fig. 2(a). While any one such partition does little to reveal the true underlying clusters, analysis of the hyperplane generating mechanism shows how multiple such partitions can discover the true clusters.



**Figure 2.** Clustering by a random hyperplane: (a) An example of splitting 2-spiral data set by a random line. Points on the same side of the line are in the same cluster. (b) Probability of splitting two one-dimensional objects for different number of random thresholds as a function of distance between objects.

Consider first the case of one-dimensional data. Splitting of objects in 1-dimensional space is done by a random threshold in  $\mathbf{R}^1$ . In general, if  $r$  thresholds are randomly selected, then  $(r+1)$  clusters are formed. It is easy to derive that, in 1-dimensional space, the probability of separating two objects whose inter-point distance is  $x$  is exactly:

$$P(\text{split}) = 1 - (1 - x/L)^r, \quad (22)$$

where  $L$  is the length of the interval containing the objects, and  $r$  threshold points are drawn at random from uniform distribution on this interval. Fig. 2(b) illustrates the dependence for  $L=1$  and  $r=1,2,3,4$ . If a co-association matrix is used to combine  $H$  different partitions, then the expected value of co-association between two objects is  $H(1 - P(\text{split}))$ , that follows from the binomial distribution of the number of splits in  $H$  attempts. Therefore, the co-association values found after combining many random split partitions are generally expected to be a non-linear and a monotonic function of respective distances. The situation is similar for multidimensional data, however, the generation of random hyperplanes is a bit more complex. To generate a random hyperplane in  $d$  dimensions, we should first draw a random point in the multidimensional region that will serve as a point of origin. Then we randomly choose a unit normal vector  $\mathbf{u}$  that defines the hyperplane. The two objects characterized by vectors  $\mathbf{p}$  and  $\mathbf{q}$  will be in the same cluster if  $(\mathbf{u}\mathbf{p})(\mathbf{u}\mathbf{q}) > 0$  and will be separated otherwise (here  $\mathbf{a}\mathbf{b}$  denotes a scalar product of  $\mathbf{a}$  and  $\mathbf{b}$ ). If  $r$  hyperplanes are generated, then the total probability that two objects remain in the same cluster is just the product of

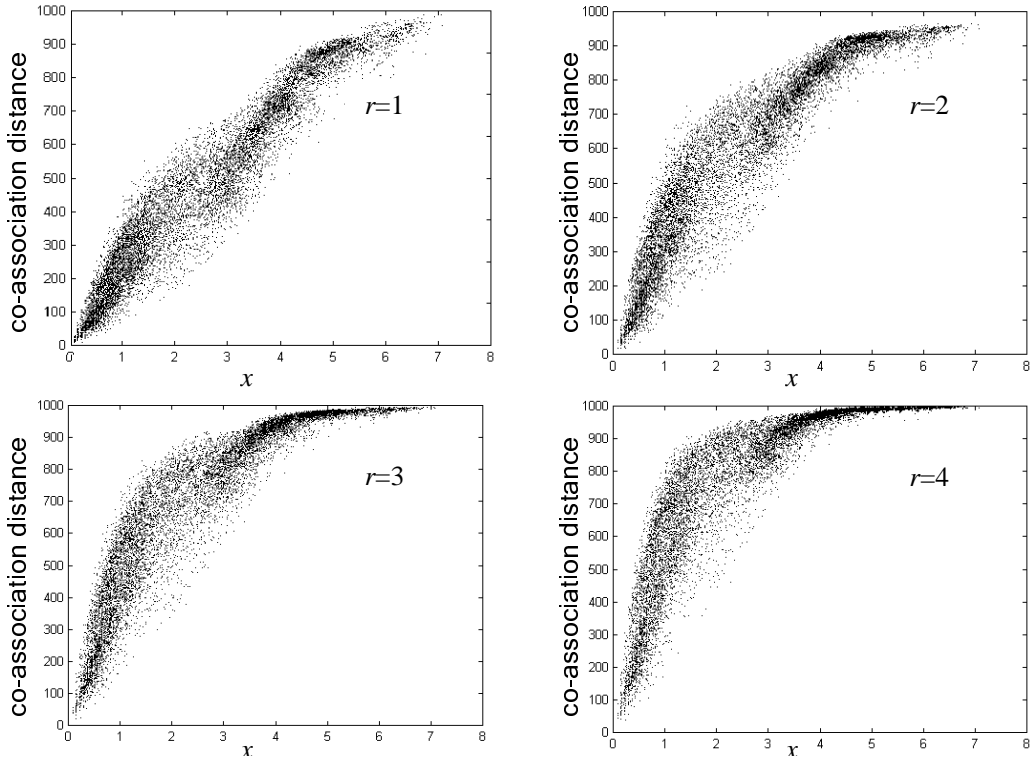
probabilities that each of the hyperplanes does not split the objects. Thus we can expect that the law governing the co-association values is close to what is obtained in 1-dimensional space in Eq. (22).

Let us compare the actual dependence of co-association values with the function in Eq. (22). Fig. 3 shows the results of experiments with 1000 different partitions by random splits of the Iris data set. The Iris data is 4-dimensional and contains 150 points. There are 11,175 pair-wise distances between the data items. For all the possible pairs of points, each plot in Fig. 3 shows the number of times a pair was split. The observed dependence of the inter-point “distances” derived from the co-association values vs. the true Euclidean distance, indeed, can be described by the function in Eq. (22).

Clearly, the inter-point distances dictate the behavior of respective co-association values. The probability of a cut between any two given objects does not depend on the other objects in the data set. Therefore, we can conclude that any clustering algorithm that works well with the original inter-point distances is also expected to work well with co-association values obtained from a combination of multiple partitions by random splits. However, this result is more of theoretical value when true distances are available, since they can be used directly instead of co-association values. It illustrates the main idea of the approach, namely that the synergy of multiple weak clusterings can be very effective. We present an empirical study of the clustering quality of this algorithm in the experimental section.

## **5.2 Combination of Clusterings in Random Subspaces**

Random subspaces are an excellent source of clustering diversity that provides different views of the data. Projective clustering is an active topic in data mining. For example, algorithms such as CLIQUE [2] and DOC [42] can discover both useful projections as well as data clusters. Here, however, we are only concerned with the use of random projections for the purpose of clustering combination.

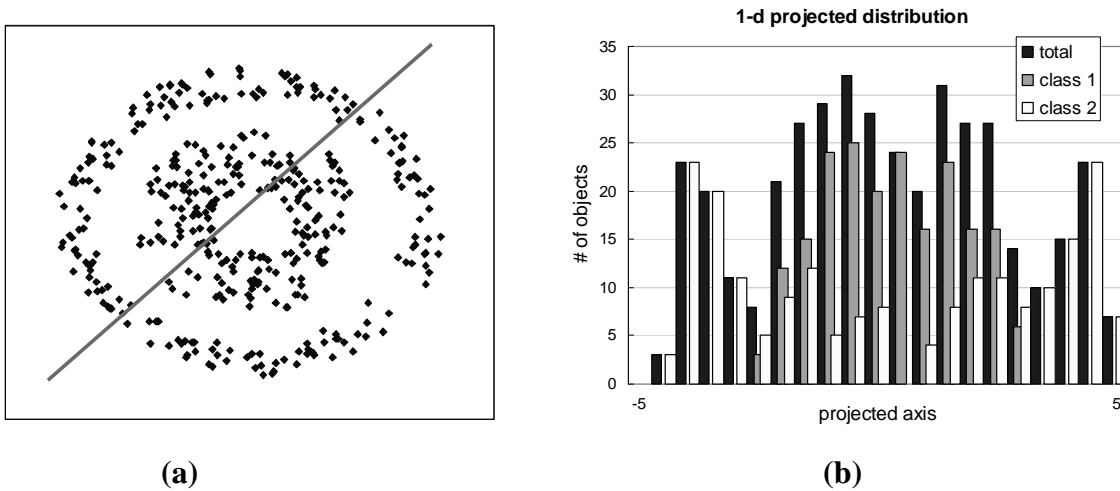


**Figure 3.** Dependence of distances derived from the co-association values vs. the actual Euclidean distance  $x$  for each possible pair of objects in Iris data. Co-association matrices were computed for different numbers of hyperplanes  $r = 1, 2, 3, 4$ .

Each random subspace can be of very low dimension and it is by itself somewhat uninformative. On the other hand, clustering in 1-dimensional space is computationally cheap and can be effectively performed by  $k$ -means algorithm. The main subroutine of  $k$ -means algorithm – distance computation – becomes  $d$  times faster in 1-dimensional space. The cost of projection is linear with respect to the sample size and number of dimensions  $O(Nd)$ , and is less than the cost of one  $k$ -means iteration.

The main idea of our approach is to generate multiple partitions by projecting the data on a random line. A fast and simple algorithm such as  $k$ -means clusters the projected data, and the resulting partition becomes a component in the combination. Afterwards, a chosen consensus function is applied to the components. We discuss and compare several consensus functions in the experimental section.

It is instructive to consider a simple 2-dimensional data and one of its projections, as illustrated in Fig. 4(a). There are two natural clusters in the data. This data looks the same in any 1-



**Figure 4.** Projecting data on a random line: (a) A sample data with two identifiable natural clusters and a line randomly selected for projection. (b) Histogram of the distribution of points resulting from data projection onto a random line.

dimensional projection, but the actual distribution of points is different in different clusters in the projected subspace. For example, Fig. 4(b) shows one possible histogram distribution of points in 1-dimensional projection of this data. There are three identifiable modes, each having a clear majority of points from one of the two classes. One can expect that clustering by  $k$ -means algorithm will reliably separate at least a portion of the points from the outer ring cluster. It is easy to imagine that projection of the data in Fig. 4(a) on another random line would result in a different distribution of points and different label assignments, but for this particular data set it will always appear as a mixture of three bell-shaped components. Most probably, these modes will be identified as clusters by  $k$ -means algorithm. Thus each new 1-dimensional view correctly helps to group some data points. Accumulation of multiple views eventually should result in a correct combined clustering.

The major steps for combining the clusterings using random 1-d projections are described by the following procedure:

---

```

begin
  for  $i=1$  to  $H$  //  $H$  is the number of clusterings in the combination
    generate a random vector  $\mathbf{u}$ , s.t.  $|\mathbf{u}|=1$ 
    project all data points  $\{\mathbf{x}_j\}$ :  $\{y_j\} \leftarrow \{\mathbf{u}\mathbf{x}_j\}$ ,  $j=1..N$ 
    cluster projections  $\{y_j\}$ :  $\pi(i) \leftarrow k\text{-means}(\{y_j\})$ 
  end
  combine clusterings via a consensus function:  $\sigma \leftarrow \{\pi(i)\}$ ,  $i=1..H$ 

```

```

return  $\sigma$  // consensus partition
end

```

---

The important parameter is the number of clusters in the component partition  $\pi_i$  returned by  $k$ -means algorithm at each iteration, i.e. the value of  $k$ . If the value of  $k$  is too large then the partitions  $\{\pi_i\}$  will overfit the data set which in turn may cause unreliability of the co-association values. Too small a number of clusters in  $\{\pi_i\}$  may not be enough to capture the true structure of data set. In addition, if the number of clusterings in the combination is too small then the effective sample size for the estimates of distances from co-association values is also insufficient, resulting in a larger variance of the estimates. That is why the consensus functions based on the co-association values are more sensitive to the number of partitions in the combination (value of  $H$ ) than consensus functions based on hypergraph algorithms.

## 6 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 the mixture model. We explored the datasets using five different consensus functions.

**6.1 Datasets.** Table 2 summarizes the details of the datasets. Five datasets of different nature have been used in the experiments. “Biochemical” and “Galaxy” data sets are described in [1] and [40], respectively.

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                     |

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.

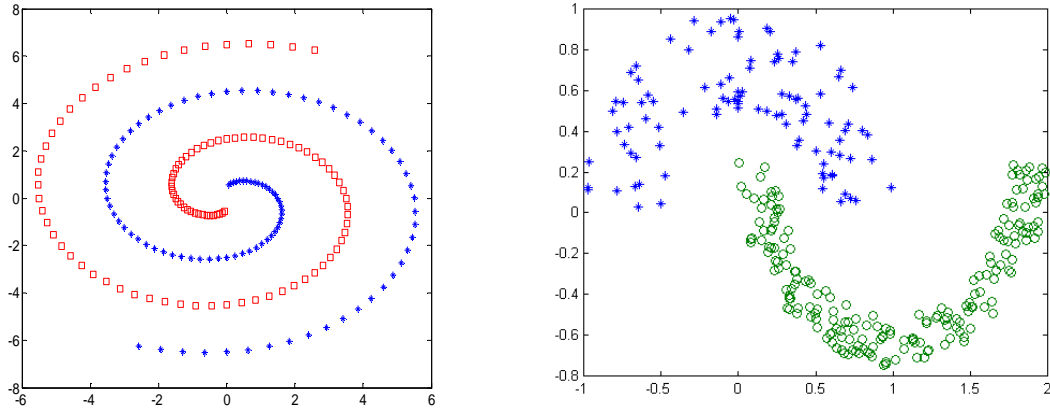
**6.2 Selection of Parameters and Algorithms.** Accuracy of the QMI and EM consensus algorithms has been compared to six other consensus functions:

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” [40] and “Biochemical” [1] data. For the same reason, we did not use other co-association methods, such as single-link clustering. The performance of these methods was already analyzed in [14,15].
2. HGPA for hypergraph partitioning.
3. MCLA, that modifies HGPA via extended set of hyperedge operations and additional heuristics.
4. Consensus functions operated on the co-association matrix, but with three different hierarchical clustering algorithms for obtaining the final partition, namely single-linkage, average-linkage, and complete-linkage.

First three methods (CSPA, HGPA and MCLA) were introduced in [47] 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 a random initialization of the algorithm. The following parameters of the clustering ensemble are especially important:

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



**Figure 5:** “2 spirals” and “Half-rings” datasets are difficult for any centroid based clustering algorithms.

- ii.  $k$  – the number of clusters in the component clusterings  $\{\pi_1, \dots, \pi_H\}$  produced by  $k$ -means algorithm was taken in the range [2..10].
- iii.  $r$  – the number of hyperplanes used for obtaining clusterings  $\{\pi_1, \dots, \pi_H\}$  by random splitting algorithm.

Both, the EM and QMI algorithms 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 and within-cluster variance is a criterion for the QMI algorithm.

**6.3 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 relatively large biochemical and astronomical data sets and from 20 runs for the 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 is superior 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 an increase in the 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.

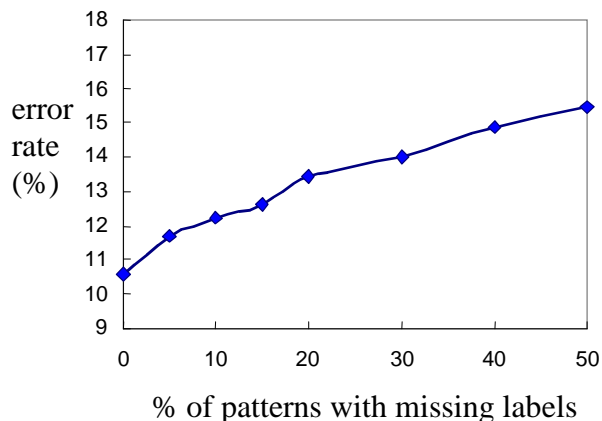
Clustering combination accuracy also depends on the number of clusters  $M$  in the ensemble partitions, or more precisely, on its ratio to the target number of clusters, i.e.  $k/M$ . For example, the EM algorithm worked best with  $k=3$  for Iris dataset,  $k=3,4$  for “Galaxy” dataset and  $k=2$  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 7 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 2) 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 [1] was the same as found by supervised classifiers applied to this dataset [45].

**6.4 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

**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 |



**Figure 6:** Consensus clustering error rate as a function of the number of missing labels in the ensemble for the Iris dataset,  $H=5$ ,  $k=3$ .

such an ensemble. The number of missing labels in each partition was varied between 10% to 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 6 ( $H=5$ ,  $k=3$ ).

One can note that combination accuracy decreases only insignificantly for “Biochemistry” 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  $H=5$ .

### 6.5 Results of Random Subspaces Algorithm

Let us start by demonstrating how the combination of clusterings in projected 1-dimensional subspaces outperforms the combination of clusterings in the original multidimensional space. Fig. 8(a) shows the learning dynamics for Iris data and  $k=4$ , using average-link consensus function based

on co-association values. Note that the number of clusters in each of the components  $\{\pi_1, \dots, \pi_H\}$  is set to  $k=4$ , and is different from the true number of clusters ( $=3$ ). Clearly, each individual clustering in full multidimensional space is much stronger than any 1-dim partition, and therefore with only a small number of partitions ( $H < 50$ ) the combination of weaker partitions is not yet effective. However, for larger numbers of combined partitions ( $H > 50$ ), 1-dim projections together better reveal the true structure of the data. It is quite unexpected, since the  $k$ -means algorithm with  $k=3$  makes, on average, 19 mistakes in original 4-dim space and 25 mistakes in 1-dim random subspace. Moreover, clustering in the projected subspace is  $d$  times faster than in multidimensional space. Although, the cost of computing a consensus partition  $\sigma$  is the same in both cases.

The results regarding the impact of value of  $k$  are reported in Fig. 8(b), which shows that there is a critical value of  $k$  for the Iris data set. This occurs when the average-linkage of co-association distances is used as a consensus function. In this case the value  $k=2$  is not adequate to separate the true clusters. The role of the consensus function is illustrated in Fig. 9. Three consensus functions are compared on the Iris data set. They all use similarities from the co-association matrix but cluster the objects using three different criterion functions, namely, single link, average link and complete link. It is clear that the combination using single-link performs significantly worse than the other two consensus functions. This is expected since the three classes in Iris data have hyperellipsoidal shape. More results were obtained on “half-rings” and “2 spirals” data sets in Fig. 5, which are traditionally difficult for any partitional centroid-based algorithm. Table 9 reports the error rates for the “2 spirals” data using seven different consensus functions, different number of component partitions  $H = [5..500]$  and different number of clusters in each component  $k = 2, 4, 10$ . We omit similar results for “half-rings” data set under the same experimental conditions and some intermediate values of  $k$  due to space limitations. As we see, the single-link consensus function performed the best and was able to identify both the ‘half-rings’ clusters as well as spirals. In contrast to the results for Iris data, average-link and complete-link consensus were not suitable for these data sets.

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

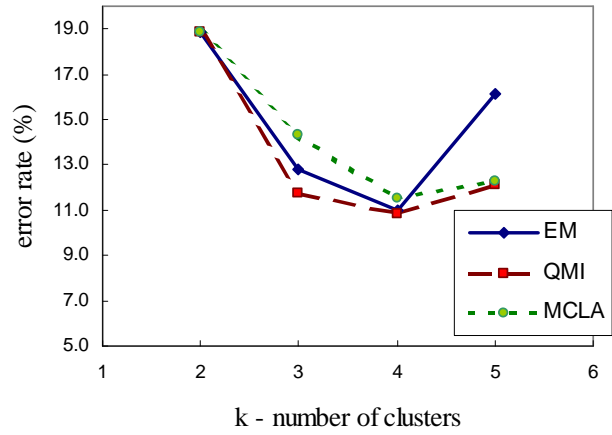
| $H$ | $k$ | Type of Consensus Function |      |      |
|-----|-----|----------------------------|------|------|
|     |     | EM                         | QMI  | MCLA |
| 5   | 2   | 44.8                       | 44.8 | 44.8 |
| 5   | 3   | 43.2                       | 48.8 | 44.7 |
| 5   | 4   | 42.0                       | 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.

| $H$ | $k$ | Type of Consensus Function |      |      |      |      |
|-----|-----|----------------------------|------|------|------|------|
|     |     | 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.

| $H$ | $k$ | Type of Consensus Function |      |      |      |      |
|-----|-----|----------------------------|------|------|------|------|
|     |     | 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 |



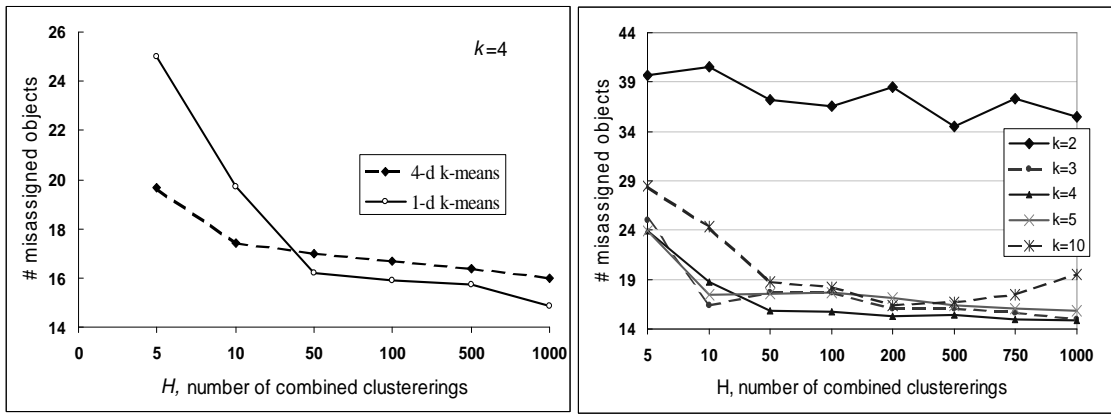
**Figure 7:** Consensus error as a function of the number of clusters in the contributing partitions for Galaxy data and ensemble size  $H=20$ .

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

| $H$ | $k$ | Type of Consensus Function |      |      |      |      |
|-----|-----|----------------------------|------|------|------|------|
|     |     | 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 |

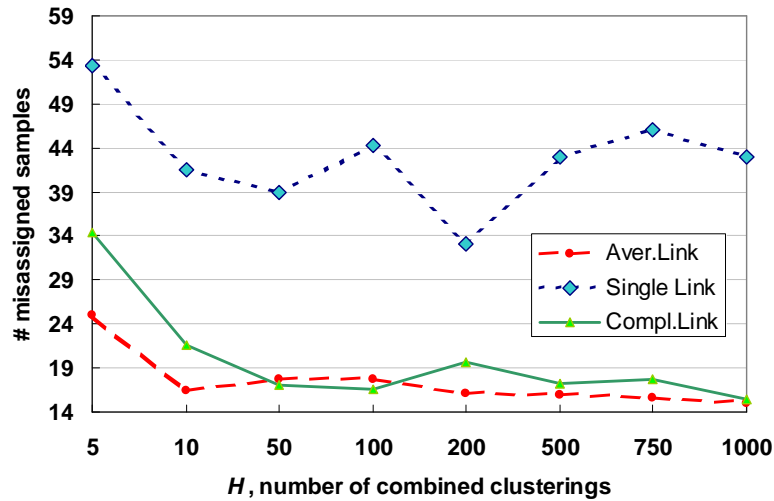
**Table 8:** Clustering error rate of EM algorithm as a function of the number of missing labels for the large datasets

| $H$ | $k$ | Missing    | "Galaxy"  | "Biochem." |
|-----|-----|------------|-----------|------------|
|     |     | 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      |



(a) (b)

**Figure 8.** Performance of random subspaces algorithm on Iris data. (a) Number of errors by the combination of  $k$ -means partitions ( $k=4$ ) in multidimensional space and projected to 1-d subspaces. Average-link consensus function was used. (b) Accuracy of projection algorithm as a function of the number of components and the number of clusters  $k$  in each component.



**Figure 9.** Dependence of accuracy of the projection algorithm on the type of consensus function for Iris data set.  $k=3$ .

This observation supports the idea that the accuracy of the consensus functions based on co-association values is sensitive to the choice of data set. In general, one can expect that average-link (single-link) consensus will be appropriate if standard average-link (single-link) agglomerative clustering works well for the data and vice versa. Moreover, none of the three hypergraph consensus functions could find a correct combined partition. This is somewhat surprising given that the hypergraph algorithms performed well on the Iris data. However, the Iris data is far less problematic

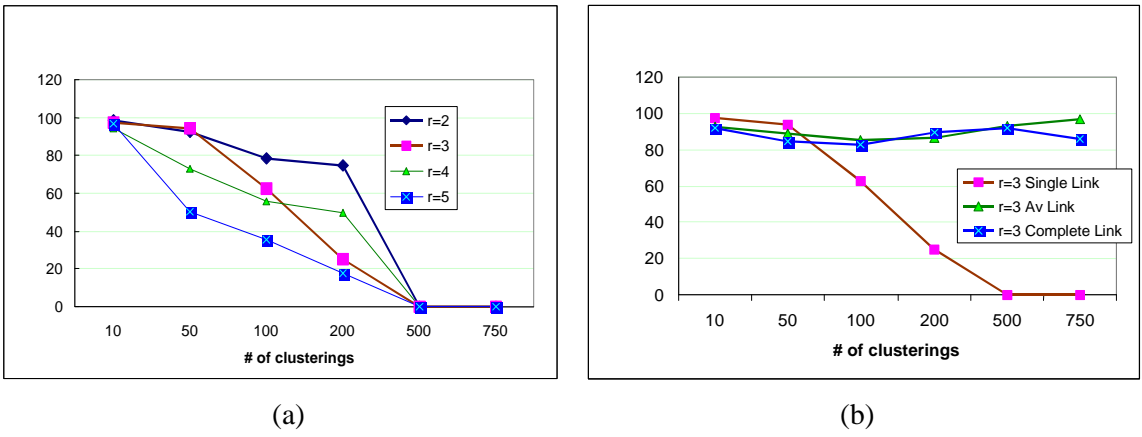
**Table 9.** “2 Spirals” data experiments. Average error rate (% over 20 runs) of clustering combination using the random 1-d projections algorithm with different number of components,  $H$ , in combination, different resolutions of components  $k$  and seven types of consensus functions.

| $H$ , # of components | $k$ , # of clusters in component | Type of Consensus Function |              |               |                    |      |      |                  |
|-----------------------|----------------------------------|----------------------------|--------------|---------------|--------------------|------|------|------------------|
|                       |                                  | Co-association methods     |              |               | Hypergraph methods |      |      | Median partition |
|                       |                                  | Single link                | Average link | Complete link | CSPA               | HGPA | MCLA | QMI              |
| 5                     | 2                                | 47.8                       | 45.0         | 44.9          | 41.7               | 50.0 | 40.8 | 40.0             |
| 10                    | 2                                | 48.0                       | 44.3         | 43.1          | 41.6               | 50.0 | 40.4 | 40.0             |
| 50                    | 2                                | 44.2                       | 43.5         | 41.6          | 43.1               | 50.0 | 41.1 | 39.3             |
| 100                   | 2                                | 46.2                       | 43.9         | 42.1          | 46.2               | 50.0 | 43.2 | 42.6             |
| 200                   | 2                                | 44.5                       | 42.5         | 41.9          | 43.3               | 50.0 | 40.1 | 40.7             |
| 500                   | 2                                | 41.6                       | 43.5         | 39.6          | 46.4               | 50.0 | 44.0 | 43.3             |
| 5                     | 4                                | 48.6                       | 45.9         | 46.8          | 43.4               | 44.9 | 43.8 | 44.7             |
| 10                    | 4                                | 47.4                       | 47.5         | 48.7          | 44.1               | 43.8 | 42.6 | 44.0             |
| 50                    | 4                                | 35.2                       | 48.5         | 46.2          | 44.9               | 39.9 | 42.3 | 44.2             |
| 100                   | 4                                | 29.5                       | 49.0         | 47.0          | 44.2               | 39.2 | 39.5 | 42.9             |
| 200                   | 4                                | 27.8                       | 49.2         | 46.0          | 47.7               | 38.3 | 37.2 | 39.0             |
| 500                   | 4                                | 4.4                        | 49.5         | 44.0          | 48.1               | 39.4 | 45.0 | 43.4             |
| 5                     | 10                               | 48.0                       | 42.6         | 45.3          | 42.9               | 42.4 | 42.8 | 45.3             |
| 10                    | 10                               | 44.7                       | 44.9         | 44.7          | 42.4               | 42.6 | 42.8 | 44.2             |
| 50                    | 10                               | 9.4                        | 47.0         | 44.3          | 43.5               | 42.4 | 42.2 | 42.8             |
| 100                   | 10                               | 0.9                        | 46.8         | 47.4          | 41.8               | 41.1 | 42.3 | 44.2             |
| 200                   | 10                               | 0.0                        | 47.0         | 45.8          | 42.4               | 38.9 | 44.5 | 40.0             |
| 500                   | 10                               | 0.0                        | 47.3         | 43.4          | 43.3               | 35.2 | 44.8 | 37.4             |

because one of the clusters is linearly separable, and the other classes are well described as a mixture of two multivariate normal distributions.

Perfect separation of natural clusters was achieved with a large number of partitions in clustering combination ( $H > 200$ ) and for values of  $k > 3$  for “half-rings” and “2 spirals”. Again, it indicates that for each problem there is a critical value of resolution of component partitions that guarantees good clustering combination. This further supports the work of Fred and Jain [15,16] who showed that a random number of clusters in each partition ensures a greater diversity of components. We see that the minimal required value of resolution for the Iris data is  $k=3$ , for “half-rings” it is  $k=2$  and for “2 spirals” it is  $k=4$ . In general, the value of  $k$  should be larger than the true number of clusters.

The number of partitions affects the relative performance of the consensus functions. With large values of  $H$  ( $>100$ ), co-association consensus becomes stronger, while with small values of  $H$  it is preferable to use hypergraph algorithms or  $k$ -means median partition algorithm.



**Figure 10.** Number of misassigned points by random hyperplanes algorithm in clustering of “2 spiral” data: (a) for different number of hyperplanes (b) for different consensus functions.

It is interesting to compare the combined clustering accuracy with the accuracy of some of the classical clustering algorithms. For example, for Iris data the EM algorithm has the best average error rate of 6.17%. In our experiments, the best performers for Iris data were the hypergraph methods, with an accuracy as low as 3%, with  $H > 200$  and  $k > 5$ . For the “half-rings” data, the best standard result is 5.25% error by the average-link algorithm, while the combined clustering using the single-link co-association algorithm achieved a 0% error with  $H > 200$ . Also, for the “2 spirals” data the clustering combination achieves 0% error, the same as by regular single-link clustering. Hence, with an appropriate choice of consensus function, clustering combination outperforms many standard clustering algorithms. However, the choice of a good consensus function is similar to the problem of choice of a good conventional clustering algorithm. Perhaps good alternative to guessing the right consensus function is simply to run all the available consensus functions and then pick the final consensus partition according to the partition utility criteria in Eq. (4) or Eq. (6). We hope to address this in future applications of the method.

Another set of experiments was performed on the “Galaxy” dataset which has significantly larger number of samples  $N = 4192$  and number of features  $d = 14$ . The task is to separate patterns of galaxies from stars. We used most difficult set of “faint” objects from the original data [40]. True labels for the objects were provided manually by experts. Even though computation of component partitions is  $d$  times faster due to projection, the overall computational effort can be

dominated by the complexity of computing the consensus partition. Quadratic computational complexity effectively prohibits co-association based consensus functions from being used on large data sets, due to  $O(N^2)$  complexity of building co-association matrix for  $N$  objects. Therefore, for large datasets we do not use three hierarchical agglomerative methods as well as CSPA hypergraph algorithm by Strehl and Ghosh. The  $k$ -means algorithm for median partition via QMI is the most attractive in terms of speed with a complexity  $O(kNH)$ . In addition, we also used two other hypergraph based consensus functions, since they worked fast in practice. Table 10 reports the achieved error rate using these consensus functions. We also limited the number of components in the combination to  $H=20$  because of the large data size. The results show that  $k$ -means algorithm for median partition has the best performance. HGPA did not work well due to its bias toward balanced cluster sizes, as it also happened in the case of the “half-rings” data set. We see that the accuracy improves when the number of partitions and clusters increases.

It is important to note that the average error rate of the standard  $k$ -means algorithm for the “Galaxy” data is about 20%, and the best known solution has an error rate of 18.6%. It is quite noticeable that  $k$ -means median partition algorithm and MCLA obtained much better partition with an error rate of only around 13% for  $k>3$ .

**6.6 Results of Random Splitting Algorithm** The same set of experiments was performed with clustering combination via splits by random hyperplanes as in section 5.1. Here we would like to emphasize only the most interesting observations, because the results in many details are close to what have been obtained by using random subspaces. There is a little difference in terms of absolute performance: the random hyperplanes algorithm is slightly better on “half-rings” data using single-link consensus function, about the same on “2 spirals”, and worse on Iris data set.

It is important to distinguish the number of clusters  $k$  in the component partition and the number of hyperplanes  $r$ , because hyperplanes intersect randomly and form varying number of clusters. For example, 3 lines can create anywhere between 4 and 7 distinct regions in a plane.



Table 10. Average error rate (in %, over 20 runs) of combination clustering using random projections algorithm on “Galaxy/star” data set.

| $H$ , # of component | $k$ , # of cl. component | Type of Consensus  |      |                  |
|----------------------|--------------------------|--------------------|------|------------------|
|                      |                          | Hypergraph methods |      | Median partition |
|                      |                          | HGPA               | MCLA | QMI              |
| 5                    | 2                        | 49.7               | 20.0 | 20.4             |
| 10                   | 2                        | 49.7               | 23.5 | 21.1             |
| 20                   | 2                        | 49.7               | 21.0 | 18.0             |
| 5                    | 3                        | 49.7               | 22.0 | 21.7             |
| 10                   | 3                        | 49.7               | 17.7 | 13.7             |
| 20                   | 3                        | 49.7               | 15.8 | 13.3             |
| 5                    | 4                        | 49.7               | 19.7 | 16.7             |
| 10                   | 4                        | 49.7               | 16.9 | 15.5             |
| 20                   | 4                        | 49.7               | 14.1 | 13.2             |
| 5                    | 5                        | 49.7               | 22.0 | 22.5             |
| 10                   | 5                        | 49.7               | 17.7 | 17.4             |
| 20                   | 5                        | 49.6               | 15.2 | 12.9             |

The results for the “2 spirals” data set also demonstrate the convergence of consensus clustering to a perfect solution when  $H$  reaches 500 and for the values of  $r = 2, \dots, 5$ . See Fig. 10(a). A larger number of hyperplanes ( $r=5$ ) improves the convergence. Fig 10(b) illustrates that the choice of consensus function is crucial for successful clustering. In the case of “2-spirals” data, only single-link consensus is able to find correct clusters.

## 7 Conclusion and Future Work

This study extended previous research on clustering ensembles in several respects. First, we have introduced a unified representation for multiple clusterings and formulated the corresponding categorical clustering problem. Second, 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. Third, it is shown that another consensus function can be related to classical intra-class variance criterion using the generalized mutual information definition. Consensus solution based on quadratic mutual information can be efficiently found by k-means algorithm in the space of specially transformed labels. Experimental results indicate good performance of the approach for several datasets and favorable comparison with other consensus

functions. Among the advantages of these algorithms are their low computational complexity and well-grounded statistical model.

We have also considered combining weak clustering algorithms that use data projections and random data splits. A simple explanatory model is offered for the behavior of combination of such weak components. We have analyzed combination accuracy as a function of parameters, which control the power and resolution of component partitions as well as the learning dynamics vs. the number of clusterings involved. Empirical study compared the effectiveness of several consensus functions applied to weak partitions.

It is interesting to continue the study of clustering combination in several directions: 1) design of effective search heuristics for consensus functions, 2) more precise and quantifiable notion of weak clustering, and 3) an improved understanding of the effect of component resolution for overall performance. This research can be extended in order to take into account non-independence of partitions in the ensemble. The consensus function presented here 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 in proportional to their strength. We hope to address these issues in our future work.

## References

- [1] E. Abola, F. Bernstein, S. Bryant, T. 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 Intl. Union of Crystallography, Bonn/Cambridge/Chester, 107–132, 1987.
- [2] R.Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, *Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications*, In Proceedings of ACM-SIGMOD 1998 Int. Conf. on Management of Data: 94-105, 1998.
- [3] J.-P. Barthélemy 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.
- [4] D. J. Bartholomew and M. Knott, *Latent variable models and factor analysis*, 2nd ed, Kendall's Library of Statistics 7. London: Arnold, 1999.
- [5] A. Ben-Hur, D. Horn, H.T. Siegelmann, and V.Vapnik, *Support vector clustering*, Journal of Machine Learning Research, 2:125-137, 2001.

- [6] L. Breiman, *Arcing classifiers*, The Annals of Statistics, 26(3): 801-849, 1998.
- [7] 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.
- [8] 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.
- [9] P. Domingos and M. Pazzani, *On the optimality of the simple Bayesian classifier under zero-one loss*, Machine Learning, 29: 103–130, 1997.
- [10] S. Dudoit and J. Fridlyand, *Bagging to improve the accuracy of a clustering procedure*, Bioinformatics, 19 (9): 1090-1099, 2003
- [11] X. Z. Fern and C. E. Brodley, *Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach*, Proc. of 20th Intl. Conf. on Machine learning, 2003.
- [12] B. Fischer, J.M. Buhmann, *Bagging for Path-Based Clustering*, IEEE Trans. on Pattern Analysis and Machine Intelligence, 25 (11), 1411-1415, 2003.
- [13] M. Figueiredo and A.K. Jain, *Unsupervised learning of finite mixture models*, IEEE Transaction on Pattern Analysis and Machine Intelligence 24:381--396, 2002.
- [14] 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.
- [15] A.L.N. Fred and A.K. Jain, *Data Clustering using Evidence Accumulation*, In Proc. of the 16th Intl. Conference on Pattern Recognition, ICPR 2002 ,Quebec City: 276 – 280, 2002.
- [16] A.L.N. Fred and A.K. Jain, *Evidence Accumulation Clustering based on the K-means algorithm*, In T.Caelli et al., eds, Structural, Syntactic, and Statistical Pattern Recognition, LNCS 2396, Springer-Verlag, 442-451, 2002.
- [17] A. Fred and A. K. Jain. *Robust data clustering*, In Proc. of IEEE Computer Society Conf, on Computer Vision and Pattern Recognition, CVPR 2003, Wisconsin, USA, June 2003.
- [18] Y. Freund and R.E. Schapire, *Experiments with a New Boosting Algorithm*, in Proc. of the Thirteenth Intl. Conference on Machine Learning, Morgan Kaufmann, 148-156, 1996.
- [19] 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.
- [20] 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.
- [21] M.A. Gluck and J.E. Corter, *Information, uncertainty, and the utility of categories*. In Proc. of the Seventh Annual Conference of the Cognitive Science Society, Hillsdale, NJ: Lawrence Erlbaum, 283-287, 1985.
- [22] L.A. Goodman and W.H. Kruskal, *Measures of association for cross classifications*. Journal of Am. Stat. Assoc.; 49: 732-64, 1954.
- [23] J. Havrda and F. Charvát, *Quantification Method of Classification Processes: Concept of Structural  $\alpha$ -Entropy*, Kybernetika, 3: 30-35, 1967.
- [24] A. Hinneburg, D. A. Keim, and M. Wawryniuk, *Using Projections to visually cluster high-dimensional Data*, Computing in Science and Engineering, 5 (2): 12-25, 2003.
- [25] T. K. Ho, *The random subspace method for constructing decision forests*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8): 832-844, 1998.
- [26] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, New Jersey, 1988.
- [27] A.K. Jain, M. N. Murty, and P. Flynn, *Data clustering: A review*, ACM Computing Surveys, 31(3): 264–323, 1999.

- [28] 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.
- [29] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, *Multilevel Hypergraph Partitioning: Applications in VLSI Design*, In Proc. ACM/IEEE Design Automation Conf., 526-529, 1997
- [30] 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.
- [31] 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 Pharmacology, 56-62, 2001.
- [32] E.M. Kleinberg, *Stochastic Discrimination*, Annals of Mathematics and Artificial Intelligence, 1: 207-239, 1990.
- [33] J. Kleinberg, *An Impossibility Theorem for Clustering*, In Proceedings of Advances in Neural Information Processing Systems (NIPS 2002) 15, 2002.
- [34] P. Langley, W. Iba, and K. Thompson, *An analysis of Bayesian classifiers*, In Proc. of the Tenth National Conf. on Artificial Intelligence, San Jose, CA, AAAI Press, 399–406, 1992.
- [35] 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.
- [36] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, *Consensus Clustering: A Resampling Based Method for Class Discovery and Visualization of Gene Expression Microarray Data*, Journal on Machine Learning, 52(1-2), 2003
- [37] R.S. Michalski and R.E Stepp, *Automated construction of classifications: Conceptual clustering versus numerical taxonomy*, IEEE Trans. Pattern Analysis and Machine Intelligence, 5, 396-410, 1983.
- [38] B. Minaei-Bidgoli, A. Topchy, and W. Punch, *Ensembles of Partitions via Data Resampling*, Proc. of IEEE Intl. Conf. on Information Technology: Coding and Computing, vol.2, pp. 188-192, 2004.
- [39] B. Mirkin, *Reinterpreting the Category Utility Function*, Machine Learning, 45(2): 219-228, 2001.
- [40] 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.
- [41] B.H. Park and H. Kargupta, *Distributed Data Mining*, In The Handbook of Data Mining, Ed. Nong Ye, Lawrence Erlbaum Associates, 2003.
- [42] C. Procopiuc, M. Jones, P. Agarwal, and T. M. Murali, *A Monte Carlo algorithm for fast projective clustering*. In Proc. 2002 SIGMOD Conference, 418-427, 2002.
- [43] Y. Qian and C. Suen, *Clustering Combination Method*, International Conference on Pattern Recognition (ICPR'00)-Volume 2, 2000.
- [44] J. R. Quinlan, *Bagging, boosting, and C4.5*. In Proceedings of the 13<sup>th</sup> AAAI Conference on Artificial Intelligence, AAAI Press, Menlo Park, CA, 725-30, 1996.
- [45] 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.
- [46] D.B. Rubin, *Inference with Missing Data*, Biometrika, 63: 581-592, 1976.
- [47] A. Strehl and J. Ghosh, *Cluster ensembles - a knowledge reuse framework for combining multiple partitions*, Journal of Machine Learning Research, 3: 583-617, 2002.
- [48] D. H. Fisher, *Knowledge acquisition via incremental conceptual clustering*, Machine Learning, 2: 139-172, 1987
- [49] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, 1982