Weighted Cluster Ensembles: Methods and Analysis

CARLOTTA DOMENICONI
and
MUNA AL-RAZGAN
George Mason University

17

Cluster ensembles offer a solution to challenges inherent to clustering arising from its ill-posed nature. Cluster ensembles can provide robust and stable solutions by leveraging the consensus across multiple clustering results, while averaging out emergent spurious structures that arise due to the various biases to which each participating algorithm is tuned. In this article, we address the problem of combining multiple *weighted clusters* that belong to different subspaces of the input space. We leverage the diversity of the input clusterings in order to generate a consensus partition that is superior to the participating ones. Since we are dealing with weighted clusters, our consensus functions make use of the weight vectors associated with the clusters. We demonstrate the effectiveness of our techniques by running experiments with several real datasets, including high-dimensional text data. Furthermore, we investigate in depth the issue of diversity and accuracy for our ensemble methods. Our analysis and experimental results show that the proposed techniques are capable of producing a partition that is as good as or better than the best individual clustering.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*; I.2.6 [**Artificial Intelligence**]: Learning; I.5.3 [**Pattern Recognition**]: Clustering

General Terms: Algorithms, Theory, Experimentation

Additional Key Words and Phrases: Cluster ensembles, subspace clustering, consensus functions, accuracy and diversity measures, text data, data mining

ACM Reference Format:

Domeniconi, C. and Al-Razgan, M. 2009. Weighted cluster ensembles: Methods and analysis. ACM Trans. Knowl. Discov. Data. 2, 4, Article 17 (January 2009), 40 pages. DOI = 10.1145/1460797.1460800 http://doi.acm.org/10.1145/1460797.1460800

This work was in part supported by NSF CAREER Award IIS-0447814.
Authors' addresses: Department of Computer Science, George Mason University, Fairfax, VA 22030; email: C. Domeniconi: carlotta@cs.gmu.edu; M. Al-Razgan: malrazga@gmu.edu.

1. INTRODUCTION

Recently, cluster ensembles have emerged as a technique for overcoming problems with clustering algorithms. It is well known that off-the-shelf clustering methods may discover different patterns in a given set of data. This is because each clustering algorithm has its own bias resulting from the optimization of different criteria. Furthermore, there is no ground truth against which the clustering result can be validated. Thus, no cross-validation technique can be carried out to tune input parameters involved in the clustering process. As a consequence, the user is equipped with no guidelines for choosing the proper clustering method for a given dataset.

A cluster ensemble consists of different partitions. Such partitions can be obtained from multiple applications of any single algorithm with different initializations, or on various bootstrap samples of the available data, or from the application of different algorithms to the same dataset. Cluster ensembles offer a solution to challenges inherent to clustering arising from its ill-posed nature: they can provide more robust and stable solutions by making use of the consensus across multiple clustering results, while averaging out emergent spurious structures that arise due to the various biases to which each participating algorithm is tuned, or to the variance induced by different data samples.

An orthogonal issue related to clustering is high dimensionality. High-dimensional data pose a difficult challenge to the clustering process. Various clustering algorithms can handle data with low dimensionality, but as the dimensionality of the data increases, these algorithms tend to break down. In high-dimensional spaces, it is highly likely that, for any given pair of points within the same cluster, there exist at least few dimensions on which the points are far apart from each other. As a consequence, distance functions that equally use all input features may not be effective. As a result, many different subspace clustering methods have been proposed [Parsons et al. 2004]. They all attempt to dodge the curse of dimensionality that affects any clustering algorithm in high-dimensional spaces.

A common scenario with high-dimensional data is that several clusters may exist in different subspaces comprised of different combinations of features. In many real-world problems, points in a given region of the input space may cluster along a given set of dimensions, while points located in another region may form a tight group with respect to different dimensions. Each dimension could be relevant to at least one of the clusters. Common global dimensionality reduction techniques are unable to capture such local structure of the data. Thus, a proper feature selection procedure should operate locally in input space. Local feature selection allows one to estimate to which degree features participate to the discovery of clusters. Such estimation is carried out using points within local neighborhoods, and it allows the embedding of adaptive distance measures in different regions of the input space.

To cope with the high-dimensionality of data, Domeniconi et al. [2004, 2007] proposed a *soft* feature selection procedure (called LAC) that depends on two input parameters. The first one is common to all clustering algorithms: the number of clusters k to be discovered in the data. The second one (called h) controls the strength of the incentive to cluster on more features. LAC assigns

weights to features according to the local variance of data along each dimension. Dimensions along which data are loosely clustered receive a small weight, which has the effect of elongating distances along that dimension. Features along which data manifest a small variance receive a large weight, which has the effect of constricting distances along that dimension. Thus, the learned weights perform a directional local reshaping of distances which allows a better separation of clusters, and therefore the discovery of different patterns in different subspaces of the original input space.

Although LAC proved to be an effective method for the discovery of subspace clusters [Domeniconi et al. 2004, 2007], the setting of the h parameter is particularly difficult, as no domain knowledge for its tuning is likely to be available. The setting of the h parameter is an open problem, and motivates the combination of LAC-based clusterings in cluster ensembles. Here we focus on setting the parameter h directly from the data. We utilize the diversity of the clusterings produced by LAC when different values of h are used, in order to generate a consensus clustering that is superior to the participating ones. The major challenge we face is to find a consensus partition from the output of the LAC algorithm to achieve an "improved" overall clustering of the data. Since we are dealing with subspace clusterings, we need to design a proper consensus function that makes use of the weight vectors associated with the input clusters.

In our previous work [Al-Razgan and Domeniconi 2006], we have designed two new consensus functions (WSPA and WBPA) for an ensemble of subspace clusterings obtained by means of the LAC algorithm. Our ensemble techniques reduce the problem of defining a consensus function to a graph partitioning problem. This article is a major extension of our prior research on cluster ensembles. Besides providing further motivation for the two previously proposed methods (WSPA and WBPA), here we introduce an additional cluster ensemble technique (WSBPA) that provides weighted clusters in output. The main advantage of WSBPA is that it provides in output, not only a partition of the data into k clusters, but also weight vectors that reflect the relevance of features within each cluster. In other words, the technique preserves the local nature of the structure discovered by LAC itself from the data (while also improving the overall quality of such local structure). This is important for the cluster prediction of future test points (especially in high dimensions).

Overall, our three techniques define a consensus function that takes into account not only how often points are grouped together across the various input clusterings, but also the degree of confidence of the groupings. LAC produces partitions, where each cluster is associated with a weight vector representative of the subspace the cluster belongs to. To build a consensus function, such weight vectors are embedded in the distance computation between points and clusters, so that individual features participate with the proper strength in the assignment of points to clusters. This characteristic is the main reason for the superior accuracy achieved by our weighted clustering ensemble algorithms (e.g., with respect to CSPA and MCLA [Strehl and Ghosh 2002]). To the best of our knowledge, our techniques provide a first attempt to improving subspace clustering results by means of ensemble systems.

Specifically, our contributions are as follows:

(1) We introduce and analyze three consensus functions for subspace clusterings. The ultimate goal of our consensus functions is to provide hard partitions of the data, along with weight vectors that convey information regarding the subspaces within which the individual clusters exist. This result is achieved by our WSBPA algorithm.
(2) We demonstrate the effectiveness of our three techniques by running experiments with several real datasets, including high-dimensional text data. Furthermore, we combine our techniques with both METIS and spectral clustering, to compute the k-way partition of the resulting graphs (previously only METIS was used [Al-Razgan and Domeniconi 2006]). Our results show the applicability of spectral clustering in conjunction with our ensemble techniques, thus enabling the use of our methods also with unbalanced data.
(3) We experimentally demonstrate the use of our subspace cluster ensemble technique for the categorization of unlabeled documents, spam/nonspam messages in particular. The analysis of relevance values credited to features (i.e., terms) reveals interesting findings, and provides insights on the nature of the spam filtering problem, and the general classification case.
(4) We investigate in great detail the issue of diversity and accuracy for our ensemble techniques. We consider two different measures of diversity: a pairwise diversity measure based on Normalized Mutual Information (NMI) that does not depend on the ensemble methodology, and a nonpairwise diversity measure based on the Adjusted Random Index (ARI) that depends on the ensemble methodology. Our objective is to determine which measure of diversity is the best indicator of good ensemble accuracy, and what is the preferred level of diversity. Such findings enable one to select, from a set of ensembles, the one that is most likely to provide good results. Our results reveal that a diversity measure based on ARI is more robust and consistent, and that high diversity signifies large accuracy.

The rest of the article is organized as follows. Section 2 discusses related work on clustering ensembles. Section 3 provides a brief description of the Locally Adaptive Clustering algorithm (LAC). Section 4 introduces and motivates our three cluster ensemble algorithms. In Section 5, a motivating example is discussed. In Section 6, we describe our experiments, and analyze the results. Section 7 investigates the use of our subspace ensemble technique for the categorization of unlabeled documents. Section 8 contains a discussion of diversity measures used in the literature, and presents our investigation and findings with respect to accuracy/diversity issues for cluster ensembles. Finally, Section 9 provides the final remarks and outlines future research directions.

2. RELATED WORK

A cluster ensemble technique is characterized by two components: the mechanism to generate diverse partitions, and the consensus function to combine the input partitions into a final clustering.

Diverse partitions are typically generated by using different clustering algorithms, or by applying a single algorithm with different parameter settings, possibly in combination with data or feature sampling. The k-means algorithm with random initializations [Fred and Jain 2002; Kuncheva et al. 2006], or with random number of clusters [Kuncheva and Hadjitodorov 2004] has been widely used in the literature to generate diverse clusterings. Topchy et al. [2003] introduce two techniques, called *weak clustering* algorithms, to produce different partitions. The first technique clusters random one-dimensional projections of multidimensional data; the second one splits the data using random hyperplanes. Random projection is used in Fern and Brodley [2003]. A different approach is proposed in Topchy et al. [2004], where the ensemble is modeled as a mixture of multivariate multinomial distributions. A unified framework for producing multiple partitions is presented in Topchy et al. [2005]. Greene et al. [2004] apply k-means, k-medoids, and fast *weak clustering* as strategies to generate diversity in clustering results, while Minaei-Bidgoli et al. [2004] propose a resampling technique that generates and then combines partitions of subsets of the data, to obtain results that reflect the entire dataset.

One popular methodology to build a consensus function utilizes a coassociation matrix [Fred and Jain 2002; Greene et al. 2004; Minaei-Bidgoli et al. 2004; Topchy et al. 2003]. Such matrix can be seen as a similarity matrix, and thus can be used with any clustering algorithm that operates directly on similarities (e.g., hierarchical clustering) [Topchy et al. 2003; Greene et al. 2004]. Kuncheva et al. [2006] have shown that good results can be obtained when the coassociation matrix is used as a data matrix in a new feature space, and k-means is ran on it. In alternative to the coassociation matrix, voting procedures have been considered to build consensus functions in Topchy et al. [2004] and in Dudoit and Fridlyand [2003]. Gondek and Hofmann [2005] derive a consensus function based on the Information Bottleneck principle: the mutual information between the consensus clustering and the individual input clusterings is maximized directly, without requiring approximation.

A different popular mechanism for constructing a consensus maps the problem onto a graph-based partitioning setting [Strehl and Ghosh 2002; Ayad and Kamel 2003; Hu 2004]. In particular, Strehl and Ghosh [2002] propose three graph-based approaches: Cluster-based Similarity Partitioning Algorithm (CSPA), HyperGraph Partitioning Algorithm (HGPA), and Meta-Clustering Algorithm (MCLA). In CSPA, a binary similarity matrix is constructed for each input clustering. Each column corresponds to a cluster: an entry has a value of one if the corresponding point belongs to the cluster, and zero otherwise. An entry-wise average of all the matrices gives an overall similarity matrix, utilized to recluster the data using a graph-partitioning based approach. The induced similarity graph, where vertices correspond to data and edge weights to similarities, is partitioned using METIS [Karypis and Kumar 1998]. HGPA seeks a partitioning of the hypergraph by cutting a minimal number of hyperedges. (Each hyperedge represents a cluster of an input clustering.) All hyperedges have the same weight. This algorithm looks for a hyperedge separator that partitions the hypergraph into k unconnected components of approximately the same size. It makes use of the package HMETIS [Karypis and

Kumar 1998]. MCLA is based on the clustering of clusters. It provides object-wise confidence estimates of cluster membership. Hyperedges are grouped, and each data point is assigned to the collapsed hyperedge in which it participates most strongly.

We observe that all the ensemble methods discussed above take hard clustering as input. A recent paper [Punera and Ghosh 2007] aims at combining soft partitionings of data (e.g., produced by fuzzy k-mean) without hardening the partitions before entering them into a consensus mechanism. The authors develop soft versions of CSPA, HGPA, and MCLA.

Our work on ensembles for subspace clusterings differs from all the previous approaches as it builds consensus functions that accept in input subspace clustering results. Our work is related to the recent techniques discussed in Punera and Ghosh [2007]. Our mapping, though, encodes information provided by subspace clusterings, rather than fuzzy clusterings. Fuzzy clustering, typically, produces overlapping clusters that coexist within the same space. On the other hand, LAC produces hard partitions, where each cluster is associated with a weight vector representative of the subspace the cluster belongs to. To build a consensus, such weight vectors are embedded in the distance computation between points and clusters, so that individual features participate with the proper strength in the assignment of points to clusters. The ultimate goal of our consensus functions is to provide hard partitions of the data, along with weight vectors that convey information regarding the subspaces within which the individual clusters exist. This result is achieved by our WSBPA algorithm.

3. LOCALLY ADAPTIVE CLUSTERING

In this section we briefly describe the Locally Adaptive Clustering (LAC) algorithm [Domeniconi et al. 2004, 2007]. Let us consider a set of n points in some space of dimensionality D. A *weighted cluster* is a subset of datapoints, together with a vector of weights $\mathbf{w} = (w_1, \ldots, w_D)^t$, such that the points in the cluster are close to each other according to the L_2 norm distance weighted using $\mathbf{w}$. The component w_j measures the degree of participation of feature j to the cluster. The problem is how to estimate the weight vector $\mathbf{w}$ for each cluster in the dataset.

In traditional clustering, the partition of a set of points is induced by a set of *representative* vectors, also called *centroids* or *centers*. The partition induced by discovering weighted clusters is formally defined as follows.

Definition. Given a set S of n points $\mathbf{x} \in \Re^D$, a set of k centers $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$, $\mathbf{c}_j \in \Re^D$, $j = 1, \ldots, k$, coupled with a set of corresponding weight vectors $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$, $\mathbf{w}_j \in \Re^D$, $j = 1, \ldots, k$, partition S into k sets:

$$S_j = \left\{\mathbf{x} | \left(\sum_{i=1}^{D} w_{ji}(x_i - c_{ji})^2\right)^{1/2} < \left(\sum_{i=1}^{D} w_{li}(x_i - c_{li})^2\right)^{1/2}, \forall l \neq j \right\}, j = 1, \ldots, k, \tag{1}$$

where w_{ji} and c_{ji} represent the ith components of vectors $\mathbf{w}_j$ and $\mathbf{c}_j$ respectively (ties are broken randomly).

The set of centers and weights is *optimal* with respect to the Euclidean norm, if they minimize the error measure:

$$E_1(P, W) = \sum_{j=1}^{k} \sum_{i=1}^{D} \left(w_{ji} \frac{1}{|S_j|} \sum_{\mathbf{x} \in S_j} (c_{ji} - x_i)^2 \right) \tag{2}$$

subject to the constraints: $\forall j, \sum_i w_{ji} = 1$. P and W are $(D \times k)$ matrices whose columns are $\mathbf{c}_j$ and $\mathbf{w}_j$ respectively, that is, $P = [\mathbf{c}_1 \ldots \mathbf{c}_k]$ and $W = [\mathbf{w}_1 \ldots \mathbf{w}_k]$. For shortness of notation, we set $X_{ji} = \frac{1}{|S_j|} \sum_{\mathbf{x} \in S_j} (c_{ji} - x_i)^2$, where $|S_j|$ is the cardinality of set S_j. X_{ji} represents the variance of the data in cluster j along dimension i. The solution

$$(P^*, W^*) = \arg \min_{(P,W)} E_1(P, W)$$

will discover one-dimensional clusters: it will put maximal (unit) weight on the feature with smallest dispersion X_{ji} within each cluster j, and zero weight on all other features. Our objective, instead, is to find weighted multidimensional clusters, where the unit weight gets distributed among all features according to the respective dispersion of data within each cluster. One way to achieve this goal is to add the regularization term $\sum_{i=1}^{D} w_{ji} log w_{ji}$, which represents the negative entropy of the weight distribution for each cluster. It penalizes solutions with maximal weight on the single feature with smallest variance within each cluster. The resulting error function is

$$E_2(P, W) = \sum_{j=1}^{k} \sum_{i=1}^{D} (w_{ji} X_{ji} + h w_{ji} log\, w_{ji}), \tag{3}$$

subject to the same constraints $\forall j, \sum_i w_{ji} = 1$. The coefficient $h \geq 0$ is a parameter of the procedure; it controls the relative differences between feature weights. In other words, h controls how much the distribution of weight values will deviate from the uniform distribution. This constrained optimization problem can be solved by introducing the Lagrange multipliers. It gives the solution [Domeniconi et al. 2004]:

$$w^*_{ji} = \frac{\exp(-X_{ji}/h)}{\sum_{i=1}^{D} \exp(-X_{ji}/h)} \tag{4}$$

$$c^*_{ji} = \frac{1}{|S_j|} \sum_{\mathbf{x} \in S_j} x_i. \tag{5}$$

Solution (4) puts increased weights on features along which the dispersion X_{ji} is smaller, within each cluster. The degree of this increase is controlled by the value h. Setting $h = 0$ places all weight on the feature i with smallest X_{ji}, whereas setting $h = \infty$ forces all features to be given equal weight for each cluster j.

We need to provide a search strategy to find a partition that identifies the solution clusters. We propose an approach that progressively improves the quality

of initial centroids and weights, by investigating the space near the centers to estimate the dimensions that matter the most. We start with *well-scattered* points in S as the k centroids. We initially set all weights to $1/D$. Given the initial centroids $\mathbf{c}_j$, for $j = 1, \ldots, k$, we compute the corresponding sets S_j as previously defined. We then compute the average distance X_{ji} along each dimension from the points in S_j to $\mathbf{c}_j$. The smaller X_{ji}, the stronger is the degree of participation of feature i to cluster j. We use the value X_{ji} in an exponential weighting scheme to credit weights to features (and to clusters), as given in Equation (4). The computed weights are used to update the sets S_j, and therefore the centroids' coordinates as given in Equation (5). The procedure is iterated until convergence is reached.

LAC has shown a highly competitive performance with respect to other state-of-the-art subspace clustering algorithms [Domeniconi et al. 2007]. Despite its strong performance, LAC's dependence on the setting of h is a liability. Because no domain knowledge is likely to be available, tuning h is difficult. Improving upon this aspect of LAC's performance is desirable, and we have sought such improvement through the development of cluster ensemble techniques, which is the focus of the following sections.

4. CLUSTERING ENSEMBLE TECHNIQUES

Consider a set $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ of n points. A clustering ensemble is a collection of m clustering solutions: $G = \{G_1, G_2, \ldots, G_m\}$. Each clustering solution G_L for $L = 1, \ldots, m$, is a partition of the set S, i.e. $G_L = \{G_L^1, G_L^2, \ldots, G_L^{K_l}\}$, where $\bigcup_K G_L^K = S$. Given a collection of clustering solutions C and the desired number of clusters k, the objective is to combine the different clustering solutions and compute a new partition of S into k disjoint clusters.

The challenge in cluster ensembles is the design of a proper consensus function that combines the component clustering solutions into an "improved" final clustering. In this section we introduce three consensus functions. In our ensemble techniques we reduce the problem of defining a consensus function to a graph partitioning problem. This approach has shown good results in the literature [Dhillon 2001; Strehl and Ghosh 2002; Fern and Brodley 2004]. Moreover, the weighted clusters computed by the LAC algorithm offer a natural way to define a similarity measure to be integrated in the weights associated to the edges of a graph. The overall clustering ensemble process is illustrated in Figure 1.

4.1 Weighted Similarity Partitioning Algorithm (WSPA)

LAC outputs a partition of the data, identified by the two sets $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ and $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$. Our aim here is to generate robust and stable solutions via a consensus clustering method. We can generate contributing clusterings by changing the parameter h (as illustrated in Figure 1). The objective is then to find a consensus partition from the output partitions of the contributing clusterings, so that an "improved" overall clustering of the data is obtained. Since LAC produces weighted clusters, we need to design a consensus function that makes use of the weight vectors associated with the clusters. The details of our approach are as follows.

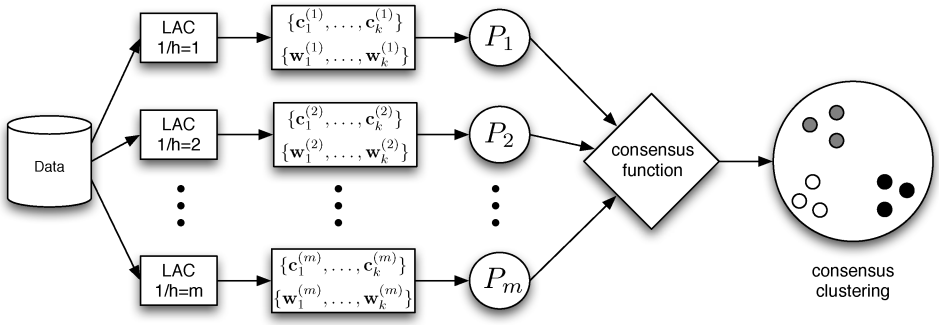

Fig. 1. The clustering ensemble process.

For each data point $\mathbf{x}_i$, the weighted distance from cluster C_l is given by

$$d_{il} = \sqrt{\sum_{s=1}^{D} w_{ls}(x_{is} - c_{ls})^2}.$$

Let $D_i = \max_l\{d_{il}\}$ be the largest distance of $\mathbf{x}_i$ from any cluster. We want to define the probability associated with cluster C_l given that we have observed $\mathbf{x}_i$. At a given point $\mathbf{x}_i$, the cluster label C_l is assumed to be a random variable from a distribution with probabilities $\{P(C_l|\mathbf{x}_i)\}_{l=1}^{k}$. We provide a nonparametric estimation of such probabilities based on the data and on the clustering result. We do not make any assumption about the specific form (e.g., Gaussian) of the underlying data distributions, thereby avoiding parameter estimations of models, which are problematic in high dimensions when the available data are limited.

In order to embed the clustering result in our probability estimations, the smaller the distance d_{il} is, the larger the corresponding probability credited to C_l should be. Thus, we can define $P(C_l|\mathbf{x}_i)$ as follows:

$$P(C_l|\mathbf{x}_i) = \frac{D_i - d_{il} + 1}{kD_i + k - \sum_l d_{il}}, \tag{6}$$

where the denominator serves as a normalization factor to guarantee $\sum_{l=1}^{k} P(C_l|\mathbf{x}_i) = 1$. We observe that $\forall l = 1, \ldots, k$ and $\forall i = 1, \ldots, n$ $P(C_l|\mathbf{x}_i) > 0$. In particular, the added value of 1 in (6) allows for a nonzero probability $P(C_L|\mathbf{x}_i)$ when $L = \arg\max_l\{d_{il}\}$. (Any small positive constant achieves this goal, with the normalization factor properly adjusted.) In this last case $P(C_l|\mathbf{x}_i)$ assumes its minimum value $P(C_L|\mathbf{x}_i) = 1/(kD_i + k - \sum_l d_{il})$. For smaller distance values d_{il}, $P(C_l|\mathbf{x}_i)$ increases proportionally to the difference $D_i - d_{il}$: the larger the deviation of d_{il} from D_i, the larger the increase. As a consequence, the corresponding cluster C_l becomes more likely, as it is reasonable to expect based on the information provided by the clustering process. Thus, Equation (6) provides a nonparametric estimation of the posterior probability associated to each cluster C_l.

We can now construct the vector P_i of posterior probabilities associated with $\mathbf{x}_i$:

$$P_i = (P(C_1|\mathbf{x}_i), P(C_2|\mathbf{x}_i), \ldots, P(C_k|\mathbf{x}_i))^t, \tag{7}$$

where t denotes the transpose of a vector. The transformation $\mathbf{x}_i \rightarrow P_i$ maps the D dimensional data points $\mathbf{x}_i$ onto a new space of *relative coordinates* with respect to cluster centroids, where each dimension corresponds to one cluster. This new representation embeds information from both the original input data and the clustering result.

To compute the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ we used both the cosine similarity and the Kullback-Leibler (KL) divergence. The cosine similarity between probability vectors associated to $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as:

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{P_i^t P_j}{\|P_i\| \|P_j\|}. \tag{8}$$

In alternative, we compute the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ using the symmetric KL divergence [Kullback and Leibler 1951]:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2} \sum_{l=1}^{k} P_{il} \log_2 \frac{P_{il}}{P_{jl}} + \frac{1}{2} \sum_{l=1}^{k} P_{jl} \log_2 \frac{P_{jl}}{P_{il}}. \tag{9}$$

We then transform the distance into a similarity measure: $s(\mathbf{x}_i, \mathbf{x}_j) = 1 - d(\mathbf{x}_i, \mathbf{x}_j)/(\max_{p,q} d(\mathbf{x}_p, \mathbf{x}_q))$. Both versions of WSPA (with cosine similarity and KL divergence) gave similar results. Thus, in this paper we report the results obtained with cosine similarity.

We combine all pairwise similarities (8) into an $(n \times n)$ similarity matrix S, where $S_{ij} = s(\mathbf{x}_i, \mathbf{x}_j)$. We observe that, in general, each clustering may provide a different number of clusters, with different sizes and boundaries. The size of the similarity matrix S is independent of the clustering approach, thus providing a way to align the different clustering results onto the same space, with no need to solve a label correspondence problem.

After running the LAC algorithm m times for different values of the h parameter, we obtain the m similarity matrices $S_1, S_2, \ldots, S_m$. The combined similarity matrix Ψ defines a *consensus function* that can guide the computation of a consensus partition:

$$\Psi = \frac{1}{m} \sum_{l=1}^{m} S_l. \tag{10}$$

Ψ_{ij} reflects the average similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ (through P_i and P_j) across the m contributing clusterings.

We now map the problem of finding a consensus partition to a graph partitioning problem. We construct a complete graph $G = (V, E)$, where $|V| = n$ and the vertex V_i identifies $\mathbf{x}_i$. The edge E_{ij} connecting the vertices V_i and V_j is assigned the weight value Ψ_{ij}. We run METIS [Karypis and Kumar 1998] on the resulting graph to compute a k-way partitioning of the n vertices that minimizes the edge weight-cut.[1] This gives the consensus clustering we seek. The size of the resulting graph partitioning problem is n^2. The steps of the algorithm, which we call WSPA (Weighted Similarity Partitioning Algorithm), are summarized in the following.

[1]In our experiments we also apply spectral clustering to compute a k-way partitioning of the n vertices.

Input: n points $\mathbf{x} \in R^D$, and k.

(1) Run LAC m times with different h values. Obtain m partitions: $\{\mathbf{c}_1^v, \ldots, \mathbf{c}_k^v\}$, $\{\mathbf{w}_1^v, \ldots, \mathbf{w}_k^v\}$, $v = 1, \ldots, m$

(2) For each partition $v = 1, \ldots, m$:
   (a) Compute $d_{il}^v = \sqrt{\sum_{s=1}^{D} w_{ls}^v (x_{is} - c_{ls}^v)^2}$
   (b) Set $D_i^v = \max_l \{d_{il}^v\}$
   (c) Compute $P(C_l^v|\mathbf{x}_i) = \frac{D_i^v - d_{il}^v + 1}{kD_i^v + k - \sum_l d_{il}^v}$
   (d) Set $P_i^v = (P(C_1^v|\mathbf{x}_i), P(C_2^v|\mathbf{x}_i), \ldots, P(C_k^v|\mathbf{x}_i))^t$
   (e) Compute the similarity

$$s^v(\mathbf{x}_i, \mathbf{x}_j) = \frac{P_i^v P_j^v}{\|P_i^v\| \|P_j^v\|}, \forall i, j$$

   (f) Construct the matrix S^v where $S_{ij}^v = s^v(\mathbf{x}_i, \mathbf{x}_j)$

(3) Build the *consensus function* $\Psi = \frac{1}{m} \sum_{v=1}^{m} S^v$

(4) Construct the complete graph $G = (V, E)$, where $|V| = n$ and $V_i \equiv \mathbf{x}_i$. Assign Ψ_{ij} as the weight value of the edge E_{ij} connecting the vertices V_i and V_j

(5) Run METIS (or spectral clustering) on the resulting graph G

Output: The resulting k-way partition of the n vertices

4.2 Weighted Bipartite Partitioning Algorithm (WBPA)

Our second approach (WBPA) maps the problem of finding a consensus partition to a bipartite graph partitioning problem. This mapping was first introduced in Fern and Brodley [2004]. In Fern and Brodley [2004], however, 0/1 weight values are used. Here we extend the range of weight values to [0,1].

The technique described here has a conceptual advantage with respect to WSPA. We observe that the consensus function ψ used in WSPA measures pairwise similarities which are solely instance-based. On the other hand, the bipartite graph partitioning problem, to which the WBPA technique reduces, partitions both cluster vertices and instance vertices simultaneously. Thus, it also accounts for similarities between clusters. Consider, for example, four instances $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_4$. Suppose that $\mathbf{x}_1$ and $\mathbf{x}_2$ are never clustered together in the input clusterings, and the same holds for $\mathbf{x}_3$ and $\mathbf{x}_4$. However, the groups to which $\mathbf{x}_1$ and $\mathbf{x}_2$ belong often share the same instances, but this is not the case for the groups $\mathbf{x}_3$ and $\mathbf{x}_4$ belong to. Intuitively, we would consider $\mathbf{x}_1$ and $\mathbf{x}_2$ more similar to each other than $\mathbf{x}_3$ and $\mathbf{x}_4$. But WSPA is unable to distinguish these two cases, and may assign low similarity values to both pairs. On the other hand, WBPA is able to differentiate the two cases by modeling both instance-based and cluster-based similarities.

The graph in WBPA models both instances (e.g., datapoints) and clusters, and the graph edges can only connect an instance vertex to a cluster vertex, forming a bipartite graph. In detail, we proceed as follows for the construction of the graph. Suppose, again, that we run the LAC algorithm m times for different values of the h parameter. For each instance $\mathbf{x}_i$, and for each clustering v=1, ..., m, we then can compute the vector of posterior probabilities P_i^v, as defined in

Equations (7) and (6). Using the P vectors, we construct the following matrix A:

$$A = \begin{pmatrix} (P_1^1)^t & (P_1^2)^t & \dots & (P_1^m)^t \\ (P_2^1)^t & (P_2^2)^t & \dots & (P_2^m)^t \\ \vdots & \vdots & & \vdots \\ (P_n^1)^t & (P_n^2)^t & \dots & (P_n^m)^t \end{pmatrix}. \tag{11}$$

Note that the $(P_i^\nu)^t$s are row vectors (t denotes the transpose). The dimensionality of A is therefore $n \times km$, under the assumption that each of the m clusterings produces k clusters. (We observe that the definition of A can be easily generalized to the case where each clustering may discover a different number of clusters.)

Based on A we can now define a bipartite graph to which our consensus partition problem maps. Consider the graph $G = (V, E)$ with V and E constructed as follows. $V = V^C \cup V^I$, where V^C contains km vertices, each representing a cluster of the ensemble, and V^I contains n vertices, each representing an input data point. Thus $|V| = km + n$. The edge E_{ij} connecting the vertices V_i and V_j is assigned a weight value defined as follows. If the vertices V_i and V_j represent both clusters or both instances, then $E(i, j) = 0$; otherwise, if vertex V_i represents an instance $\mathbf{x}_i$ and vertex V_j represents a cluster C_j^ν (or vice versa) then the corresponding entry of E is $A(i, k(\nu - 1) + j)$. More formally:

—$E(i, j) = 0$ when $((1 \le i \le km)$ and $(1 \le j \le km))$ or $((km + 1 \le i \le km + n)$ and $(km + 1 \le j \le km + n))$ (This is the case in which V_i and V_j are both clusters or both instances.)

—$E(i, j) = A(i - km, j)$ when $(km + 1 \le i \le km + n)$ and $(1 \le j \le km)$ (This is the case in which V_i is an instance and V_j is a cluster.)

—$E(i, j) = E(j, i)$ when $(1 \le i \le km)$ and $(km + 1 \le j \le km + n)$ (This is the case in which V_i is a cluster and V_j is an instance.)

Note that the dimensionality of E is $(km+n) \times (km+n)$, and E can be written as follows:

$$E = \begin{pmatrix} 0 & A^t \\ A & 0 \end{pmatrix}.$$

A partition of the bipartite graph G partitions the cluster vertices and the instance vertices simultaneously. The partition of the instances can then be output as the final clustering. Due to the special structure of the graph G (sparse graph), the size of the resulting bipartite graph partitioning problem is kmn. Assuming that $(km) \ll n$, this complexity is much smaller than the size n^2 of WSPA.

The steps of the algorithm, which we call WBPA (Weighted Bipartite Partitioning Algorithm), are summarized in the following.

Input: n points $\mathbf{x} \in R^D$, and k

(1) Run LAC m times with different h values. Obtain the m partitions: $\{\mathbf{c}_1^\nu, \dots, \mathbf{c}_k^\nu\}$,$\{\mathbf{w}_1^\nu, \dots, \mathbf{w}_k^\nu\}$, $\nu = 1, \dots, m$

(2) For each partition $\nu = 1, \dots, m$:

(a) Compute $d_{il}^v = \sqrt{\sum_{s=1}^{D} w_{ls}^v (x_{is} - c_{ls}^v)^2}$
(b) Set $D_i^v = \max_l \{d_{il}^v\}$
(c) Compute $P(C_l^v|\mathbf{x}_i) = \frac{D_i^v - d_{il}^v + 1}{kD_i^v + k - \sum_l d_{il}^v}$
(d) Set $P_i^v = (P(C_1^v|\mathbf{x}_i), P(C_2^v|\mathbf{x}_i), \ldots, P(C_k^v|\mathbf{x}_i))^t$

(3) Construct the matrix A as in (11)

(4) Construct the bipartite graph $G = (V, E)$, where $V = V^C \cup V^I$, $|V^I| = n$ and $V_i^I \equiv \mathbf{x}_i$, $|V^C| = km$ and $V_j^C \equiv C_j$ (a cluster of the ensemble). Set $E(i, j) = 0$ if V_i and V_j are both clusters or both instances. Set $E(i, j) = A(i - km, j) = E(j, i)$ if V_i and V_j represent an instance and a cluster

(5) Run METIS (or spectral clustering) on the resulting graph G

Output: The resulting k-way partition of the n vertices in V^I

We observe that WBPA captures instance-based similarity. Suppose, for example, that $\mathbf{x}_1$ and $\mathbf{x}_2$ are always clustered together in the m input clusterings. Then, the weights, $P(C_i^v|\mathbf{x}_1)$ and $P(C_i^v|\mathbf{x}_2)$, of the edges connecting $\mathbf{x}_1$ and $\mathbf{x}_2$ to the same cluster vertex C_i^v have high values, for $v = 1, \ldots, m$. As a consequence, the k-way partitioning of the n instances will not cut such edges. As a result, $\mathbf{x}_1$ and $\mathbf{x}_2$ will be grouped together in the final consensus clustering.

4.3 Weighted Subspace Bipartite Partitioning Algorithm (WSBPA)

The two algorithms WSPA and WBPA provide as output a partition of the data into k clusters, with no information regarding feature relevance for each of the clusters. Next, we discuss a clustering ensemble algorithm (WSBPA) that provides weighted clusters in output. Our approach represents the first attempt in the literature to produce subspace clustering results within the context of ensemble research. This technique advances the WBPA method (Section 4.2) by adding to the final partition weighted features associated with each cluster. By assigning a value to each dimension, WSBPA captures the local relevance of features within each cluster. Thus, the structure of the output provided by a single run of LAC is preserved. The output of WSBPA, then, becomes twofold, and has good potential to advance the research on the label assignment problem, which is a difficult and open research issue. For example, for text documents, the analysis of weights assigned to features (i.e., terms) can guide the identification of keywords representative of the topics discussed in the documents. Possibly, relevant keywords, combined with associated weight values, can be used to provide short summaries for clusters and to automatically annotate documents (e.g., for indexing purposes). We will demonstrate this further in Section 7.

As we mentioned in our discussion on WBPA, a partition of the bipartite graph G partitions the cluster and the instance vertices simultaneously. However, only the partition of the instance vertices is used to output the final result in WBPA; the partition of the cluster vertices is discarded. WSBPA also uses the partition of cluster vertices; such partition reflects cluster-based similarities. Specifically, WSBPA utilizes the information associated with the partitioned cluster vertices to compute weight vectors for the final clustering.

Let us consider the bipartite graph $G = (V, E)$ as constructed by the algorithm WBPA. We recall that $V = V^C \cup V^I$, where V^C contains km vertices, each representing a cluster of the ensemble, and V^I contains n vertices, each representing an input datapoint. A k-way partition of the bipartite graph G partitions the cluster vertices and the instance vertices simultaneously into k sets. Furthermore, the k-way partition of G provides a one-to-one correspondence between the k elements of the partition of V^C and the k elements of the partition of V^I. In symbols, let $P_{V^C} = \{V_1^C, V_2^C, \ldots, V_k^C\}$ be the partition of V^C into k sets, and let $P_{V^I} = \{V_1^I, V_2^I, \ldots, V_k^I\}$ be the partition of V^I into k sets. V_j^C and V_j^I, for $j = 1, \ldots, k$, are the sets of cluster vertices and instance vertices grouped together by the k-way partitioning of graph G.

As in WBPA, the partition P_{V^I} provides the resulting clustering of the n input datapoints $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Each element in P_{V^C} is a set of cluster vertices: $V_l^C = \{v_{l_1}^C, \ldots, v_{l_{|V_l^C|}}^C\}$, for $l = 1, \ldots, k$. Each element in V_l^C represents a cluster from a run of the LAC algorithm. Thus, it has an associated weight vector. Let $\mathbf{w}_{l_i}^C$ be the weight vector associated with the cluster vertex $v_{l_i}^C$. We average the weight vectors $\mathbf{w}_{l_i}^C$, for $i = 1, \ldots, |V_l^C|$, to obtain the weights for cluster V_l^I, for $l = 1, \ldots, k$:

$$\mathbf{w}_l = \frac{1}{|V_l^C|} \sum_{i=1}^{|V_l^C|} \mathbf{w}_{l_i}^C. \tag{12}$$

We therefore obtain k clusters along with the associated weight vectors: $\{(V_l^I, \mathbf{w}_l)\}_{l=1}^k$. We observe that a k-way partitioning of G that minimizes the edge weight-cut groups together instances $\mathbf{x}$ and clusters C with a high value for $P(C|\mathbf{x})$. This means that, according to LAC clustering, C is a likely cluster given that we have observed $\mathbf{x}$. Thus, the weight vector for the cluster containing $\mathbf{x}$ should be close to the weight vector associated with C. The averaging in (12) gives each cluster C (i.e., the corresponding weight) with high $P(C|\mathbf{x})$ equal importance for the computation of the weight of the cluster containing $\mathbf{x}$. The steps of the algorithm, which we call Weighted Subspace Bipartite Partitioning Algorithm (WSBPA) are summarized in the following.

Input: n points $\mathbf{x} \in R^D$, and k

(1) Run LAC m times with different h values. Obtain the m partitions: $\{\mathbf{c}_1^\nu, \ldots, \mathbf{c}_k^\nu\}$,$\{\mathbf{w}_1^\nu, \ldots, \mathbf{w}_k^\nu\}$, $\nu = 1, \ldots, m$

(2) For each partition $\nu = 1, \ldots, m$:
   (a) Compute $d_{il}^\nu = \sqrt{\sum_{s=1}^D w_{ls}^\nu (x_{is} - c_{ls}^\nu)^2}$
   (b) Set $D_i^\nu = \max_l \{d_{il}^\nu\}$
   (c) Compute $P(C_l^\nu|\mathbf{x}_i) = \frac{D_i^\nu - d_{il}^\nu + 1}{kD_i^\nu + k - \sum_l d_{il}^\nu}$
   (d) Set $P_i^\nu = (P(C_1^\nu|\mathbf{x}_i), P(C_2^\nu|\mathbf{x}_i), \ldots, P(C_k^\nu|\mathbf{x}_i))^t$

(3) Construct the A matrix as in (11)

(4) Construct the bipartite graph $G = (V, E)$ as in the algorithm WBPA

(5) Run METIS (or spectral clustering) on the resulting graph G. Consider the resulting partitions $P_{V^C} = \{V_1^C, V_2^C, \ldots, V_k^C\}$ and $P_{V^I} = \{V_1^I, V_2^I, \ldots, V_k^I\}$ of the cluster and instance vertices respectively

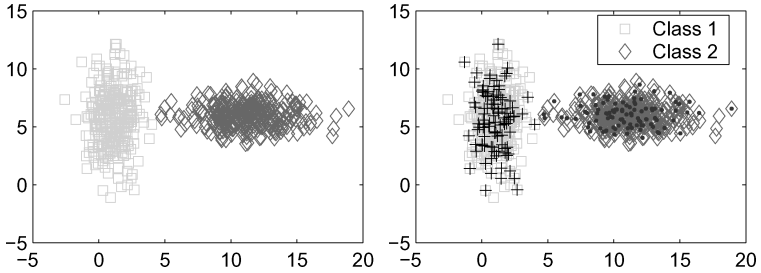

Fig. 2. (Left): Two-Gaussian data. (Right): Random sampling of 100 points (crosses and dots) from each cluster.

(6) Compute the average weight vector $\mathbf{w}_l$ for each element V_l^C in P_{VC}, as given in Equation (12)

Output: The resulting weight vectors coupled with the corresponding cluster centroids: $\{(\mathbf{c}_l^I, \mathbf{w}_l)\}_{l=1}^k$, where $\mathbf{c}_l^I$ is the centroid of cluster V_l^I

5. AN ILLUSTRATIVE EXAMPLE

Here we present and discuss an illustrative example to demonstrate that the relative coordinates $P(C|\mathbf{x})$ provide a suitable representation for the computation of pairwise similarities. We emphasize that this is an important point since the information provided by the subspace clustering is embedded into these coordinates, and, in turn, the proposed consensus function is constructed upon such representation of the data. Thus, the efficacy of the consensus function itself relies on the suitability of these coordinates.

We have designed one simulated dataset with two clusters distributed as bivariate Gaussians (Figure 2(Left)). The mean and standard deviation vectors for each cluster are as follows: $\mathbf{m}_1 = (0.5, 5)$, $\mathbf{s}_1 = (1, 9)$; $\mathbf{m}_2 = (12, 5)$, $\mathbf{s}_2 = (6, 2)$. Each cluster has 300 points. We ran the LAC algorithm on the Two-Gaussian dataset for two values of the $1/h$ parameter (7 and 12). For $(1/h) = 7$, LAC provides a perfect separation (the error rate is 0.0%); the corresponding weight vectors associated to each cluster are $\mathbf{w}_1^{(7)} = (0.81, 0.19)$, $\mathbf{w}_2^{(7)} = (0.18, 0.82)$. For $(1/h) = 12$, the error rate of LAC is 5.3%; the weight vectors in this case are $\mathbf{w}_1^{(12)} = (0.99, 0.01)$, $\mathbf{w}_2^{(12)} = (0.0002, 0.9998)$.

For the purpose of plotting the two-dimensional posterior probability vectors associated with each point $\mathbf{x}$, we consider a random sample of 100 points from each cluster (as shown in Figure 2(Right)). The probability vectors (computed as in equations (7) and (6)) of such sample points are plotted in Figure 3(Left) and Figure 3(Right), respectively for $(1/h) = 7$ and $(1/h) = 12$. We observe that in Figure 3 (Left) ($(1/h) = 7$) for points $\mathbf{x}$ of cluster 1 (green points square-shaped) $P(C_1|\mathbf{x}) > P(C_2|\mathbf{x})$, and for points $\mathbf{x}$ of cluster 2 (red points diamond-shaped) $P(C_2|\mathbf{x}) > P(C_1|\mathbf{x})$. Thus, there is no overlapping (in relative coordinate space) between points of the two clusters, and LAC achieves a perfect separation (the error rate is 0.0%). On the other hand, Figure 3(Right) ($(1/h) = 12$) demonstrates that for a few points $\mathbf{x}$ of cluster 1 (green points square-shaped) $P(C_1|\mathbf{x}) < P(C_2|\mathbf{x})$ (overlapping region in Figure 3(Right)). LAC misclassifies these points as members of cluster 2, which results in an error rate of 5.3%.

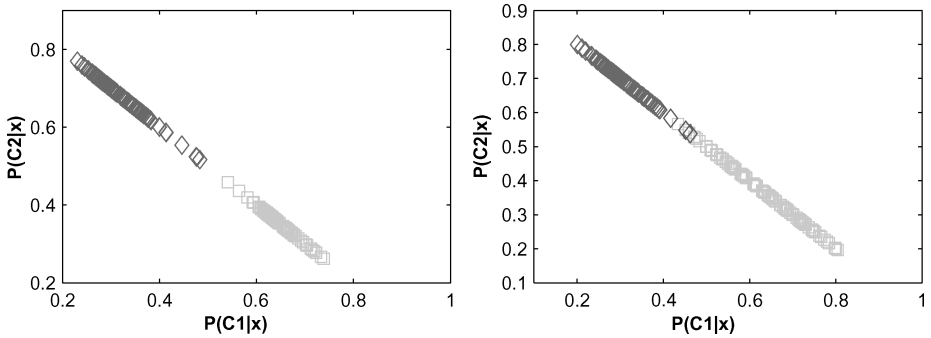

Fig. 3. (Left): Two dimensional probability vectors $P = (P(C_1|\mathbf{x}), P(C_2|\mathbf{x}))^t$, $(1/h) = 7$. LAC error rate is 0.0%. (Right): Two dimensional probability vectors $P = (P(C_1|\mathbf{x}), P(C_2|\mathbf{x}))^t$, $(1/h) = 12$.

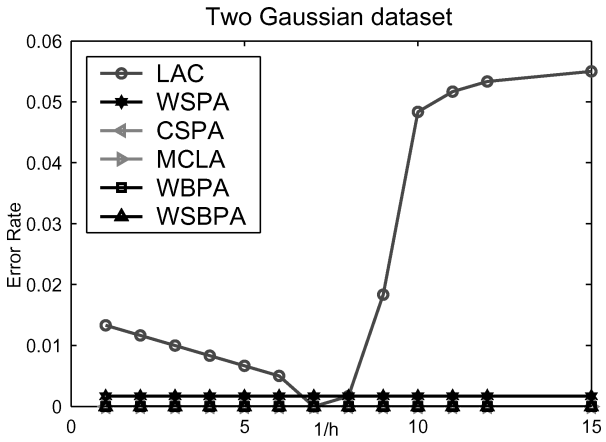

Fig. 4. Results on Two-Gaussian data. METIS was used in conjunction with WSPA, WBPA, and WSBPA.

Thus, the relative coordinates $P(C|\mathbf{x})$ provide a suitable representation to compute the pairwise similarity measure in our clustering ensemble approaches. By combining the clustering results in the relative coordinate space obtained by different runs of LAC, we aim at utilizing the consensus across multiple clusterings, while averaging out emergent spurious structures. The experimental results obtained for this dataset (presented in the next Section) corroborate our analysis. In fact, we anticipate here that our three clustering ensemble methods WSPA, WBPA, and WSBPA achieved 0.17%, 0.0%, and 0.0% error rates, respectively. Thus, they successfully separated the two clusters, as the best input clustering provided by LAC did (see Table III and Figure 4 for details).

6. EXPERIMENTAL DESIGN AND RESULTS

We have designed two simulated datasets to analyze the behavior of the proposed techniques in a controlled setting. These datasets contain two and three clusters, respectively, distributed as bivariate Gaussians (Figures 2(Left) and 5). The mean and standard deviation vectors for the Two-Gaussian dataset are

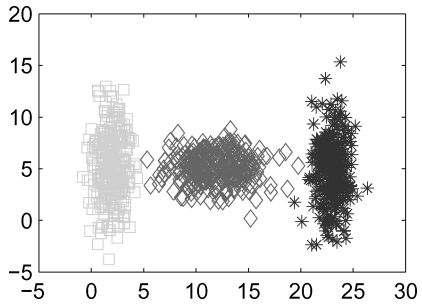

Fig. 5. Three Gaussian dataset.

Table I. Characteristics of the Datasets

Dataset	k	D	n (points-per-class)
Two-Gaussian	2	2	600 (300-300)
Three-Gaussian	3	2	900 (300-300-300)
Iris	3	4	150 (50-50-50)
WDBC	2	31	424 (212-212)
Breast	2	9	478 (239-239)
Letter(A,B)	2	16	1555 (789-766)
SatImage	2	36	2110 (1072-1038)
Spam2000	2	2000	1284 (642-642)
Spam5996	2	5996	1284 (642-642)

as described in Section 5. The mean and standard deviation vectors for the Three-Gaussian dataset are as follows: $\mathbf{m}_1 = (2, 5)$, $\mathbf{s}_1 = (1, 9)$; $\mathbf{m}_2 = (12, 5)$, $\mathbf{s}_2 = (6, 2)$; $\mathbf{m}_3 = (23, 5)$, $\mathbf{s}_3 = (1, 9)$. In our experiments, we also used seven real datasets. The characteristics of all datasets are given in Table I. Iris, Breast, Letter(A,B), and SatImage are from the UCI Machine Learning Repository [Asuncion and Newman 2007]. WDBC is the Wisconsin Diagnostic Breast Cancer dataset [Mangasarian and Wolberg 1990]. Spam2000 and Spam5996 are two high-dimensional text (spam) datasets. The documents in each dataset were preprocessed by eliminating stop words (based on a stop words list) and stemming words to their root source. As feature values in the vector space model we have used the frequency of the terms in the corresponding document. Both Spam2000 and Spam5996 belong to the Email-1431 dataset.[2] This dataset consists of emails falling into three categories: conference (370), jobs (272), and spam (786). We ran two different experiments with this dataset. In one case we reduced the dimensionality to 2000 terms (Spam2000), and in the second case to 5996 (Spam5996). In both cases we consider two clusters by merging the conference and jobs mails into one group (nonspam).

Since METIS [Karypis and Kumar 1998] requires balanced datasets, we performed random sampling on Breast, WDBC, Spam2000, and Spam5996. In each case, we subsampled the most populated class: from 357 to 212 for WDBC, from 444 to 239 for Breast, and from 786 to 642 for Spam2000 and Spam5996.

[2]The Email-1431 dataset was created by Finn Arup Nielsen. It is available at: http://www.imm.dtu.dk/∼rem/data/Email-1431.zip.

For the Letter dataset, we used the classes A and B (balanced), and for the SatImage we used classes 1 and 7 (again balanced).

Besides METIS, we also used spectral clustering[3] [Ng et al. 2002] to compute the k-way partitioning of the resulting graph, for the three techniques WSPA, WBPA, and WSBPA. The advantage of spectral clustering over METIS is that spectral clustering does not require balanced data. Here, for comparison purposes, we apply both METIS and spectral clustering on the same balanced data. Our objective is to demonstrate the applicability of spectral clustering in conjunction with our ensemble techniques, thus enabling the use of our methods also with unbalanced data.

We compared our weighted clustering ensemble techniques (WSPA, WBPA, and WSBPA) with the three methods CSPA, HGPA, and MCLA [Strehl and Ghosh 2002]. Like our methods, these three techniques transform the problem of finding a consensus clustering into a graph partitioning problem, and make use of METIS. Thus, it was a natural choice for us to compare our methods with these approaches. We consider the partitions provided by LAC (and discard the weights) in order to run CSPA, HGPA, and MCLA, since these methods are designed to accept clusterings (not subspace clusterings). In this paper we report the accuracy achieved by CSPA and MCLA, as HGPA was consistently the worst. The ClusterPack Matlab Toolbox was used.[4]

To further analyzing the benefits of diverse results generated by means of subspace clustering, we also considered a consensus function not based on a graph partitioning problem. The specific goals of these experiments are: (1) Test whether the diverse clusterings produced by LAC can be effectively combined using a consensus function based on a coassociation matrix; and (2) compare our approach of generating diversity with alternate approaches available in the literature (e.g., varying k-means). To this end, we ran LAC with different values of h as before. For each of the m resulting partitions (weights are discarded), we construct a coassociation matrix T of size $n \times n$, where $T_{ij}^{(l)} = 1$ if x_i and x_j are clustered together in partition l, $T_{ij}^{(l)} = 0$ otherwise. A final coassociation matrix T is derived by averaging the individual $T^{(l)}$, $l = 1, \ldots, m$: $T_{ij} = \frac{1}{m} \sum_{l=1}^{m} T_{ij}^{(l)}$, $i, j = 1, \ldots, n$. Previous work [Kuncheva et al. 2006; Pekalska 2005] has shown that good results can be obtained when T is used as a data matrix in a new feature space (rather then a similarity matrix). Thus, we used T as data, and ran k-means on it [Kuncheva et al. 2006]. We identify the resulting method as LAC+Co-as. To account for the subspace structure discovered by LAC, we also consider Ψ (as defined in (10)) as data matrix. We call this approach LAC+wCo-as. In addition, we ran the same consensus function on clusterings generated by k-means with random initializations. The resulting approach is denoted as k-means+Co-as. We observe that the consensus function has a random element (as it relies on k-means). Thus, we ran it 10 times, and report average accuracies. Finally, we consider another variation of k-means clustering, where we vary the number of clusters in each partition. Specifically, we ran the Evidence

[3]We used the Matlab Toolbox available at: http://www.cs.washington.edu/homes/sagarwal/code.html.

[4]Available at: www.lans.ece.utexas.edu/∼strehl/.

Accumulation Clustering (EAC) algorithm introduced in Fred and Jain [2005]. For each partition, k-means is ran with k uniformly distributed in the interval $[k_{min}, k_{max}]$, and with random initial centroids. To find the consensus partition, hierarchical clustering is used with the coassociation matrix as similarity matrix (applying both single and average link). To determine the final number of clusters, the algorithm looks for the number of clusters that has the longest lifetime on the dendogram, (i.e., the largest range of distance values on the dendogram that leads to the identification of the clusters). We consistently obtained better results with average link, and thus do not report the results for single link. We call this approach k-means-EAC-AL (k free). We also ran the same algorithm with the final number of clusters set equal to the number of classes k in the data. We call the resulting approach k-means-EAC-AL (k fixed). In our experiments, we set $k_{min} = 2$ and $k_{max} = 20$. As suggested in Fred and Jain [2005], we make sure that the range (k_{min},k_{max}) is not completely below the minimum k value.

Evaluating the quality of clustering is in general a difficult task. Since class labels are available for the datasets used here, we evaluate the results by computing the error rate, which is computed according to the confusion matrix.

We observe that the algorithm WSBPA outputs weight vectors coupled with the corresponding cluster centroids: $\{(\mathbf{c}_l^I, \mathbf{w}_l)\}_{l=1}^k$. In order to compute the corresponding partition, we assign each point to the closest centroid according to the locally weighted Euclidean distance.

6.1 Analysis of the Results

For each dataset, we ran the LAC algorithm for several values of the input parameter h. The clustering results of LAC are then given as input to the consensus clustering techniques being compared. (As the value of k, we input both LAC and the ensemble algorithms with the actual number of classes in the data.) Figures 4 and 7 plot the error rate (%) achieved by LAC as a function of the $1/h$ parameter, for each dataset considered. The error rates of our weighted clustering ensemble methods (WSPA, WBPA, and WSBPA in conjunction with METIS), and of the CSPA and MCLA techniques are also reported. Each figure clearly shows the sensitivity of the LAC algorithm to the value of h. The trend of the error rate clearly depends on the data distribution. Detailed results for all data are provided in Tables II–XI, where we report the error rate (ER) of the ensembles, and the maximum, minimum, and average error rate values for the input clusterings. Thirteen methods are being compared: our three methods WSPA, WBPA, WSBPA, each combined with both METIS and spectral clustering (SPEC is short for spectral clustering), CSPA and MCLA, and the five techniques based on a coassociation matrix. The value in parenthesis reported for k-means-EAC-AL (k free) corresponds to the number of clusters in the consensus clustering.

We further illustrate the sensitivity of the LAC algorithm to the value of h for the Three-Gaussian data (Figure 5). Figures 6(Left) and 6(Right) depict the clustering results of LAC for $(1/h) = 1$ and $(1/h) = 4$, respectively. Figure 6 (Left) clearly shows that for $(1/h) = 1$, LAC is unable to discover the structure

Table II. Average Error Rates on all Real Datasets

	Avg-Error
WSPA-METIS	7.07
WSPA-SPEC	**6.04**
WBPA-METIS	6.67
WBPA-SPEC	6.14
WSBPA-METIS	8.51
WSBPA-SPEC	8.32
CSPA	8.59
MCLA	9.36
LAC	13.59
LAC+Co-as.	10.07
LAC+wCo-as.	9.7
k-means+Co-as.	16.78
k-means+EAC-AL (k fixed)	29.8
k-means+EAC-AL (k free)	28.4

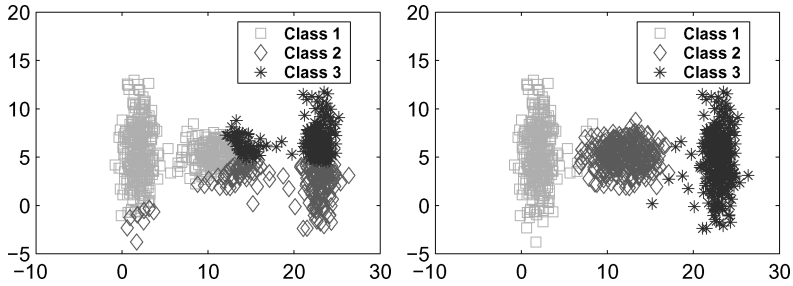

Fig. 6. (Left): LAC: Clustering results for Three-Gaussian data, $(1/h) = 1$. The error rate is 34.6%. (Right): LAC: Clustering results for Three-Gaussian data, $(1/h) = 4$. The error rate is 1.3%

of the three clusters, and gives an error rate of 34.6%. On the other hand, LAC achieves a nearly perfect separation for $(1/h) = 4$, as shown in Figure 6 (Right). The error rate in this case is 1.3%, which is also the minimum achieved in all the runs of the algorithm. Results for the ensemble techniques on the Three-Gaussian data are given in Figure 7 and in Table IV. We observe that the WSPA(-METIS) technique perfectly separates the data (0.0% error), and that WBPA(-METIS) gives a 0.44% error rate. In both cases, the error rate achieved is lower than the minimum error rate among the input clusterings (1.3%). Moreover, WSBPA gives an error rate of 1.3%, which is equal to the lowest error rate achieved by LAC. We note that WSBPA(-METIS) and MCLA provide the same error rate for this problem. However, WSBPA produces not only a partition of points as the final result, but also relevance values of features associated with each cluster. In this regard, WSBPA provides more information, and is therefore superior to MCLA.

In general, all three of our ensemble techniques were able to filter out spurious structures identified by individual runs of LAC, and provided a better error rate than (or equal to) LAC's minimum error rate. For all seven real datasets either WBPA, WSPA, or WSBPA provided the lowest error rate among the

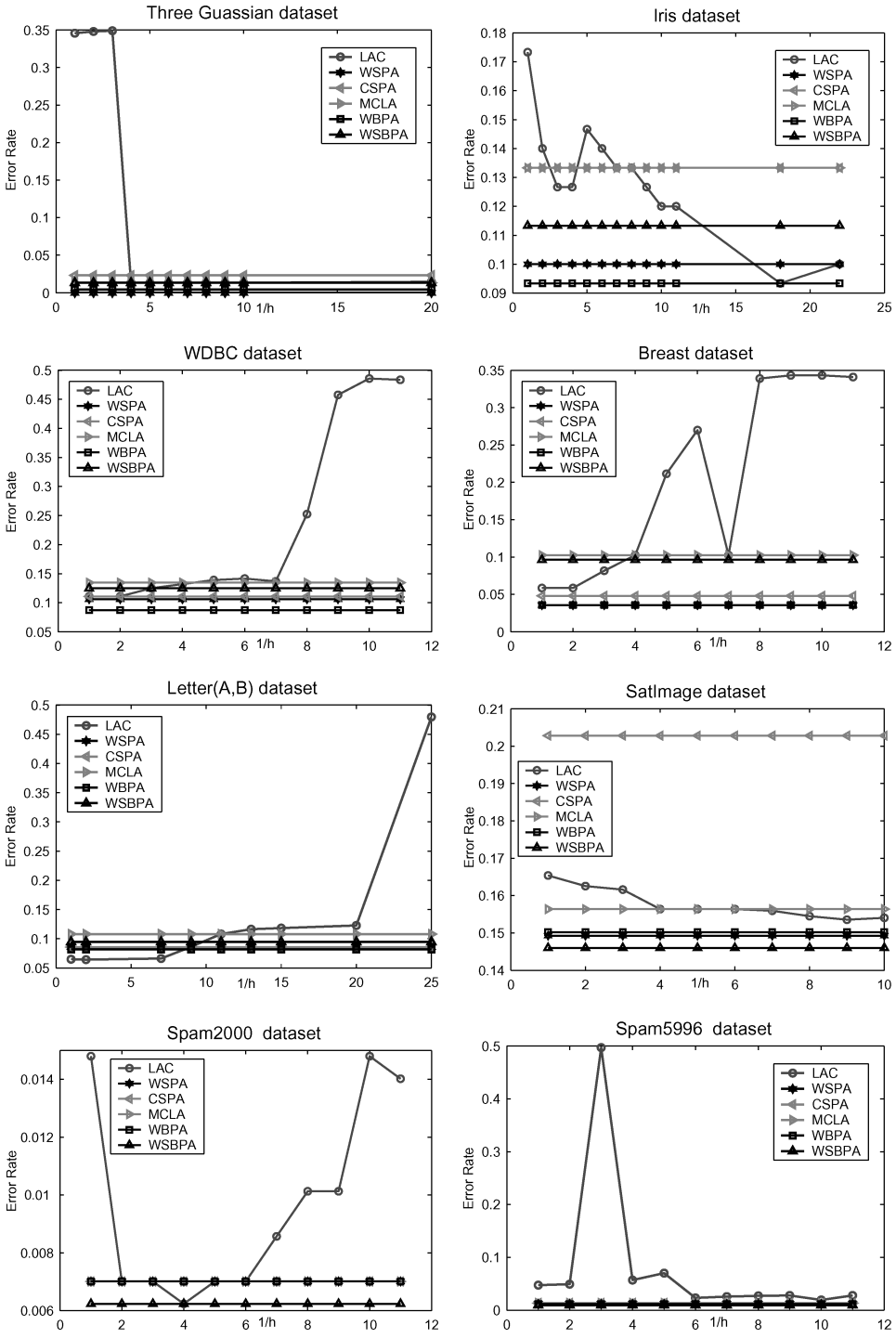

Fig. 7. Clustering Ensemble Results. METIS was used in conjunction with WSPA, WBPA, and WSBPA.

Table III. Results on Two-Gaussian Data

	Ens-ER	Max-ER	Min-ER	Avg-ER
WSPA-METIS	0.17	5.5	0	2.2
WSPA-SPEC	1.3	5.5	0	2.2
WBPA-METIS	**0**	5.5	0	2.2
WBPA-SPEC	1.3	5.5	0	2.2
WSBPA-METIS	**0**	5.5	0	2.2
WSBPA-SPEC	**0**	5.5	0	2.2
CSPA	**0**	5.5	0	2.2
MCLA	**0**	5.5	0	2.2
LAC+Co-as.	**0**	5.5	0	2.2
LAC+wCo-as.	0.18	5.5	0	2.2
k-means+Co-as.	1.3	1.3	1.3	1.3
k-means+EAC-AL (k fixed)	0.5	0.8	0.0	0.45
k-means+EAC-AL (k free)	**0** ($k = 217$)	0.8	0.0	0.45

Table IV. Results on Three Gaussian Data

	Ens-ER	Max-ER	Min-ER	Avg-ER
WSPA-METIS	**0**	34.9	1.3	10.5
WSPA-SPEC	2.2	34.9	1.3	10.5
WBPA-METIS	0.44	34.9	1.3	10.5
WBPA-SPEC	1.3	34.9	1.3	10.5
WSBPA-METIS	1.3	34.9	1.3	10.5
WSBPA-SPEC	1.56	34.9	1.3	10.5
CSPA	2.3	34.9	1.3	10.5
MCLA	1.3	34.9	1.3	10.5
LAC+Co-as.	17.3	34.9	1.3	10.5
LAC+wCo-as.	2.7	34.9	1.3	10.5
k-means+Co-as.	17.2	1.2	1.1	1.17
k-means+EAC-AL (k fixed)	0.3	33.3	0.2	4.0
k-means+EAC-AL (k free)	0.3 ($k = 3$)	33.3	0.2	4.0

methods being compared. For the Iris, WDBC, Breast, SatImage, and Spam5996 datasets (five out of seven total), the error rate provided by the WBPA technique is as good or better than the best individual input clustering. For the Letter(A,B) and Spam2000 datasets, the error rate of WBPA is still below the average error rate of the input clusterings. WSPA gave excellent results as well. For Iris, WDBC, Breast, SatImage, and Spam5996 the error rate provided by WSPA is lower than the best individual input clustering. For Spam2000 (with METIS) and Letter(A,B) the error rate of WSPA is well below the average error rate of the input clusterings.

Also WSBPA performed quite well. It produced error rates comparable with, and sometime better than, the other techniques. In addition, WSBPA provides information on the relevance of features associated with each cluster. In each dataset, WSBPA achieved a result far superior to the average error rate of the input clusterings. Furthermore, we note that for Iris, SatImage, Spam2000, and Spam5996 (four out of seven total) WSBPA has provided a result superior to both the results provided by CSPA and MCLA. In particular, WSBPA (both with METIS and SPEC) produced excellent results for the high-dimensional

Table V. Results on Iris Data

	Ens-ER	Max-ER	Min-ER	Avg-ER
WSPA-METIS	10.00	17.3	9.3	12.9
WSPA-SPEC	**6.00**	17.3	9.3	12.9
WBPA-METIS	9.3	17.3	9.3	12.9
WBPA-SPEC	6.6	17.3	9.3	12.9
WSBPA-METIS	11.3	17.3	9.3	12.9
WSBPA-SPEC	9.3	17.3	9.3	12.9
CSPA	13.3	17.3	9.3	12.9
MCLA	13.3	17.3	9.3	12.9
LAC+Co-as.	20.8	17.3	9.3	12.9
LAC+wCo-as.	15.7	17.3	9.3	12.9
k-means+Co-as.	19.9	33.3	10.6	17.6
k-means+EAC-AL (k fixed)	12.0	12.7	2.7	7.1
k-means+EAC-AL (k free)	12.0 ($k = 4$)	12.7	2.7	7.1

Table VI. Results on WDBC Data

	Ens-ER	Max-ER	Min-ER	Avg-ER
WSPA-METIS	10.6	48.5	11.1	23.4
WSPA-SPEC	10.3	48.5	11.1	23.4
WBPA-METIS	**8.7**	48.5	11.1	23.4
WBPA-SPEC	10.3	48.5	11.1	23.4
WSBPA-METIS	12.5	48.5	11.1	23.4
WSBPA-SPEC	12.7	48.5	11.1	23.4
CSPA	11.1	48.5	11.1	23.4
MCLA	13.4	48.5	11.1	23.4
LAC+Co-as.	12.9	48.5	11.1	23.4
LAC+wCo-as.	12.7	48.5	11.1	23.4
k-means+Co-as.	49.7	49.7	49.7	49.7
k-means+EAC-AL (k fixed)	49.8	46.2	33.7	38.9
k-means+EAC-AL (k free)	46.2 ($k = 4$)	46.2	33.7	38.9

data Spam2000 and Spam5996. In these two cases, WSBPA produced better results than the four competing techniques, and achieved a lower error rate than (or equal to) the minimum error rate among the input clusterings.

Clearly, our weighted clustering ensemble techniques are capable of achieving superior accuracy results with respect to the CSPA and MCLA techniques on the tested datasets. This result is summarized in Table II, where we report the average error rate on all real datasets. We observe that, on average, SPEC performed better than METIS. We also report the average values for the LAC algorithm to emphasize the large improvements obtained by the ensembles across the real datasets. Given the competitive behavior shown by LAC in the literature [Domeniconi et al. 2007], this is a significant result.

We observe that the consensus function Ψ defined in (10) measures the similarity of points in terms of how close the "patterns" captured by the corresponding probability vectors are. As a consequence, Ψ (as well as the matrix A for the WBPA and WSBPA techniques) takes into account not only how often the points are grouped together across the various input clusterings, but also the degree of confidence of the groupings. On the other hand, the CSPA and

Table VII. Results on Breast Data

	Ens-ER	Max-ER	Min-ER	Avg-ER
WSPA-METIS	**3.6**	34.1	5.9	20.5
WSPA-SPEC	3.77	34.1	5.9	20.5
WBPA-METIS	**3.6**	34.1	5.9	20.5
WBPA-SPEC	3.77	34.1	5.9	20.5
WSBPA-METIS	9.6	34.1	5.9	20.5
WSBPA-SPEC	10.0	34.1	5.9	20.5
CSPA	4.8	34.1	5.9	20.5
MCLA	10.3	34.1	5.9	20.5
LAC+Co-as.	8.2	34.1	5.9	20.5
LAC+wCo-as.	11.0	34.1	5.9	20.5
k-means+Co-as.	5.2	5.2	4.8	5.1
k-means+EAC-AL (k fixed)	5.0	4.8	2.1	3.4
k-means+EAC-AL (k free)	5.0 ($k = 2$)	4.8	2.1	3.4

Table VIII. Results on Letter(A,B) Data

	Ens-ER	Max-ER	Min-ER	Avg-ER
WSPA-METIS	8.6	47.9	6.4	13.6
WSPA-SPEC	**6.6**	47.9	6.4	13.6
WBPA-METIS	8.2	47.9	6.4	13.6
WBPA-SPEC	**6.6**	47.9	6.4	13.6
WSBPA-METIS	9.9	47.9	6.4	13.6
WSBPA-SPEC	9.4	47.9	6.4	13.6
CSPA	8.6	47.9	6.4	13.6
MCLA	10.8	47.9	6.4	13.6
LAC+Co-as.	10.8	47.9	6.4	13.6
LAC+wCo-as.	10.0	47.9	6.4	13.6
k-means+Co-as.	11.9	18.0	7.3	12.6
k-means+EAC-AL (k fixed)	24.9	19.0	2.1	7.5
k-means+EAC-AL (k free)	19.2 ($k = 3$)	19.0	2.1	7.5

MCLA approaches take as input the partitions provided by each contributing clustering algorithm. That is, $\forall \nu$ and $\forall i$, $P(C_l^\nu|\mathbf{x}_i) = 1$ for a given l, and 0 otherwise. Thus, the information concerning the degree of confidence associated with the clusterings is lost. This is likely the reason for the superior performance achieved by our weighted clustering ensemble algorithms.

In some cases, the WBPA technique gives a lower error rate compared to the WSPA technique (WBPA-METIS performs slightly better than WSPA-METIS, on average). This result may be due to the conceptual advantage of WBPA with respect to WSPA discussed at the beginning of Section 4.2. The consensus function ψ used in WSPA measures pairwise similarities which are solely instance-based. On the other hand, the bipartite graph partitioning problem, to which the WBPA technique reduces, partitions both cluster vertices and instance vertices simultaneously. Thus, it also accounts for similarities between clusters.

The results obtained for LAC+Co-as. and LAC+wCo-as. show that the diverse clusterings produced by LAC can be effectively combined using also a consensus function based on a coassociation matrix. LAC+wCo-as. gives on

Table IX. Results on SatImage Data

	Ens-ER	Max-ER	Min-ER	Avg-ER
WSPA-METIS	14.9	16.5	15.4	15.8
WSPA-SPEC	**13.2**	16.5	15.4	15.8
WBPA-METIS	15.0	16.5	15.4	15.8
WBPA-SPEC	**13.2**	16.5	15.4	15.8
WSBPA-METIS	14.5	16.5	15.4	15.8
WSBPA-SPEC	15.2	16.5	15.4	15.8
CSPA	20.3	16.5	15.4	15.8
MCLA	15.6	16.5	15.4	15.8
LAC+Co-as.	15.6	16.5	15.4	15.8
LAC+wCo-as.	16.1	16.5	15.4	15.8
k-means+Co-as.	15.7	15.7	15.6	15.7
k-means+EAC-AL (k fixed)	17.5	19.0	1.0	6.3
k-means+EAC-AL (k free)	17.5 ($k = 2$)	19.0	1.0	6.3

Table X. Results on Spam2000 Data

	Ens-ER	Max-ER	Min-ER	Avg-ER
WSPA-METIS	0.7	1.5	0.6	0.9
WSPA-SPEC	1.4	1.5	0.6	0.9
WBPA-METIS	0.7	1.5	0.6	0.9
WBPA-SPEC	1.4	1.5	0.6	0.9
WSBPA-METIS	**0.6**	1.5	0.6	0.9
WSBPA-SPEC	**0.6**	1.5	0.6	0.9
CSPA	0.7	1.5	0.6	0.9
MCLA	0.7	1.5	0.6	0.9
LAC+Co-as.	0.7	1.5	0.6	0.9
LAC+wCo-as.	0.8	1.5	0.6	0.9
k-means+Co-as.	9.7	47.9	5.4	22.3
k-means+EAC-AL (k fixed)	49.3	48.6	2.2	18.5
k-means+EAC-AL (k free)	49.3 ($k = 2$)	48.6	2.2	18.5

average lower error rates than LAC+Co-as. This is expected since LAC+wCo-as. embeds the subspace structure discovered by LAC into the consensus function. The coassociation matrix is also effective when combined with k-means (note that the high error rate of k-means+Co-as. on the WDBC data is due to the fact that k-means gave the same high error rate on each single run. See Table VI.) Overall, though, LAC provides better accuracy/diversity trade-offs, which lead to more accurate ensembles (see Table II). k-means-EAC-AL (both with k fixed and k free) provides very good results on the Gaussian data (see Tables III and IV). This demonstrates the advantage of using k-means with larger values of k when the clusters are shaped as elongated Gaussians. (We observe that, although k-means-EAC-AL (k free) achieves zero error rate on the Two-Gaussian data, the algorithm identifies 217 clusters, and most of the clusters contains just few points.) k-means-EAC-AL provides the poorest results on average on the real datasets. We observe that the error rate of the consensus clustering in each case is very close to the largest error rate among the components. For k fixed, this happens in part because the algorithm is forced to find a partition with a number of clusters equal to the number of classes. For k free,

Table XI. Results on Spam5996 Data

	Ens-ER	Max-ER	Min-ER	Avg-ER
WSPA-METIS	1.17	49.7	1.9	7.9
WSPA-SPEC	**0.93**	49.7	1.9	7.9
WBPA-METIS	1.12	49.7	1.9	7.9
WBPA-SPEC	**0.93**	49.7	1.9	7.9
WSBPA-METIS	**0.93**	49.7	1.9	7.9
WSBPA-SPEC	**0.93**	49.7	1.9	7.9
CSPA	1.3	49.7	1.9	7.9
MCLA	1.3	49.7	1.9	7.9
LAC+Co-as.	1.5	49.7	1.9	7.9
LAC-wCo-as.	1.6	49.7	1.9	7.9
k-means+Co-as.	5.4	49.7	5.4	41.2
k-means+EAC-AL (k fixed)	49.9	49.3	2.5	37.5
k-means+EAC-AL (k free)	49.9 ($k = 2$)	49.3	2.5	37.5

the algorithm itself settles for a consensus partition with a number of clusters very close or equal to the number of classes. High error rates are obtained on WDBC, Spam2000, and Spam5996. We observe that, in these cases, the algorithm groups almost the entire collection of data in a single cluster. Although k-means-EAC-AL (k free) identifies the correct number of clusters for the Spam data, it fails to discover any structure (almost all points populate a single cluster). We tested a variety of ranges for (k_{min},k_{max}) on WDBC, including those suggested in Fred and Jain [2005] (e.g., [2,20], [10,30], [60,90], [10,50],[2,10]), but the results did not improve.

We finally tested how the size of the ensemble affects the error rate. Figure 8 shows the results for WSPA-METIS and WBPA-METIS on the real data sets. Each point corresponds to an average of ten ensembles of the corresponding size. Ensemble components are randomly chosen from a collection of 50 partitions obtained by running LAC with $1/h = 1, \ldots, 50$. Ensemble sizes between 10 and 45 are considered. Overall, the error rate slowly decreases as the ensemble size increases. An ensemble size of 25–30 components seems to be a reasonable choice in general.

7. CATEGORIZATION OF UNLABELED DOCUMENTS: AN APPLICATION

Here we investigate the use of our subspace cluster ensemble technique (WSBPA) for the categorization of unlabeled documents. The output of WSBPA is twofold: it provides a partition of the data and a measure of local feature relevance for each identified group of data. For text documents, the analysis of relevance values (i.e., weights) credited to features (i.e., terms) can assist the identification of descriptive words representative of topics discussed in the documents.

To demonstrate these concepts we performed experiments with two datasets: spam Email-1431 and 20 Newsgroups. To reduce the dimensionality of the data, we followed the procedure presented in Kang et al. [2005]. Documents were first preprocessed by eliminating stop and rare words, and by stemming words to their root source. A global unsupervised feature selection procedure, based on frequent itemset mining, was then applied. The objective of this step is to

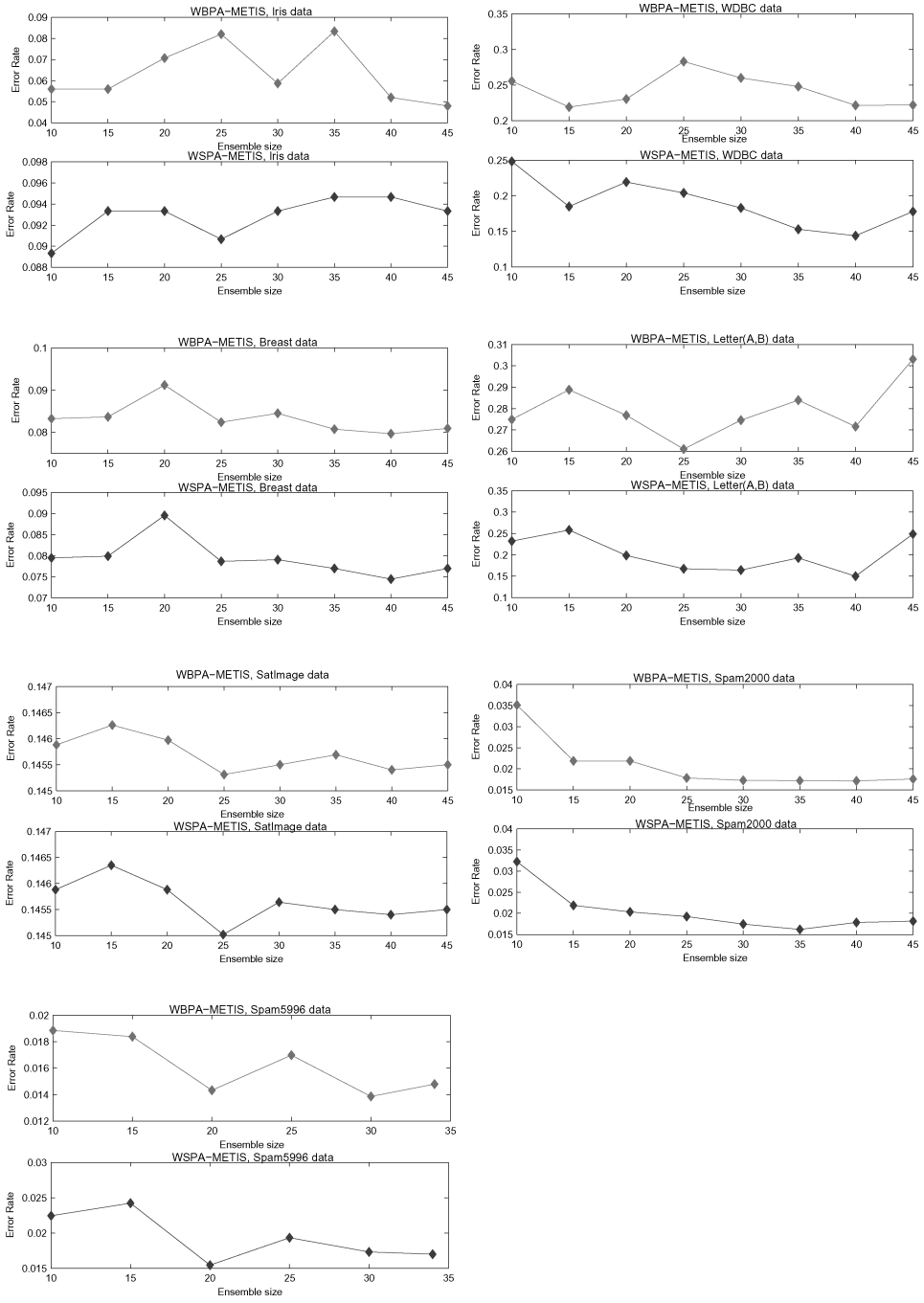

Fig. 8. Error rate as a function of the ensemble size.

Table XII. Results on Email-1431

	Ens-ER	Min-ER	Max-ER	Avg-ER
WSPA-METIS	5.3	1.5	2.2	1.95
WBPA-METIS	5.3	1.5	2.2	1.95
WSBPA-METIS	1.6	1.5	2.2	1.95
WSPA-SPEC	1.5	1.5	2.2	1.95
WBPA-SPEC	1.6	1.5	2.2	1.95
WSBPA-SPEC	1.6	1.5	2.2	1.95

Table XIII. Results on 20 Newsgroups (electronic, medical)

	Ens-ER	Min-ER	Max-ER	Avg-ER
WSPA-METIS	18.16	16.79	46.17	20.37
WBPA-METIS	16.95	16.79	46.17	20.37
WSBPA-METIS	16.89	16.79	46.17	20.37
WSPA-SPEC	17.15	16.79	46.17	20.37
WBPA-SPEC	17.09	16.79	46.17	20.37
WSBPA-SPEC	17.19	16.79	46.17	20.37

identify sets of terms that co-occur frequently in the given corpus of documents. Such terms become the features used in the final representation of documents.

Email-1431 is the same dataset used in the experiments described in Section 6. The original size of the dictionary is 38,713. After processing the data as described previously, the dictionary size was reduced to 285. As before, we ran a two-class classification problem by merging the conference and jobs emails into one group (nonspam). 20 Newsgroups is a collection of 20,000 messages collected from 20 different netnews newsgroups. One thousand messages from each of the 20 newsgroups were chosen at random and partitioned by newsgroups name. In our experiments we consider the categories medical (990) and electronics (981). The original size of the dictionary is 24,546; after processing the data, the dictionary size was reduced to 321.

Tables XII and XIII report the results we obtained for these two datasets. We ran our three methods (WSPA, WBPA, and WSBPA) using both METIS and spectral clustering. We report the ensemble error rate, and minimum, maximum and average error rates of the input clusterings. (Figure 9 shows the ranges of values for the parameter h used to construct the ensembles.)

WSBPA gives good results in both cases. We observe that for the Email-1431 dataset, WSBPA gives the same error rate (1.6%) when combined with either METIS or spectral clustering (as shown in Figure 9(Left) and in Table XII). Such error rate is very close to the minimum error rate provided by the runs of LAC. Moreover, WSBPA significantly outperforms WSPA and WBPA when METIS is used. With SPEC, all three methods provide similar results. The fact that SPEC performs better than METIS might be due to the slightly unbalanced data (786 spams vs. 642 non-spams). Also for the 20 Newsgroups dataset (electronic, medical), WSBPA gives an error rate that is very close to the minimum error rate provided by LAC (for both METIS and SPEC) (see Figure 9(Right) and Table XIII). In this case, METIS and SPEC give similar results (the dataset is balanced).

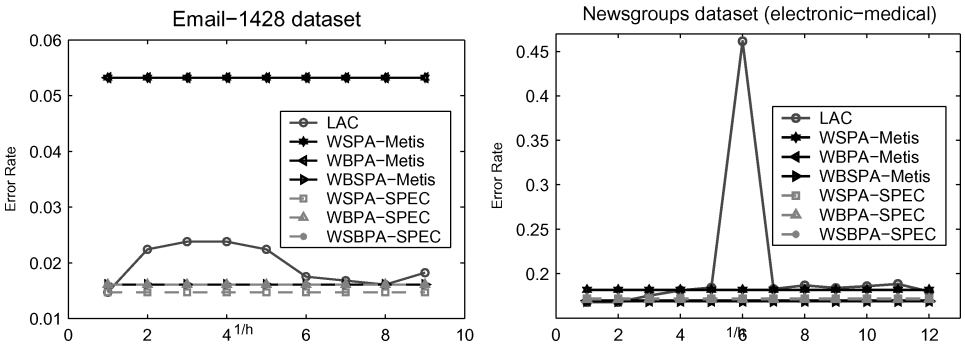

Fig. 9. Results on text datasets. (Left): Email-1431 dataset. (Right): 20 Newsgroups dataset (electronic-medical).

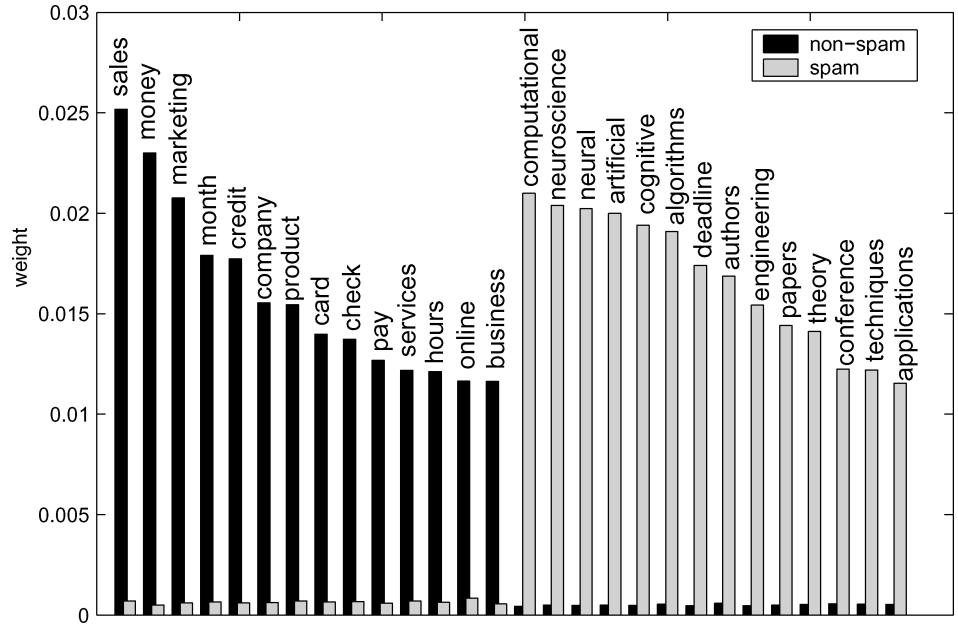

Fig. 10. Email-1431: Words and Corresponding Weight Values.

7.1 Analysis of Weights

We analyzed the weights credited to features by the algorithm WSBPA (combined with METIS). The analysis of weights assigned to words provides some insights on the nature of the spam filtering problem and the general classification case. As Figures 10 and 11 show, the selected words (i.e., those words that receive largest weight values) are representative of the underlying categories, which provides evidence that our subspace cluster ensemble technique is capable of sifting relevant words, while discarding (i.e., assigning a low weight value) spurious ones.

Let us consider the distribution of weights obtained for the Email-1431 dataset. Figure 10 shows the weight values and corresponding words for the

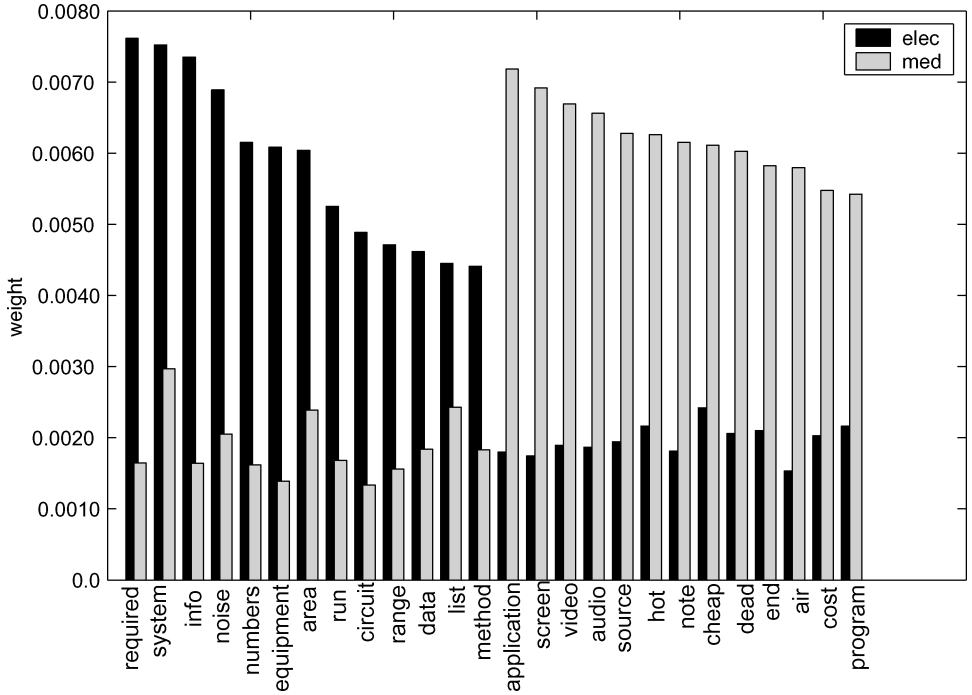

Fig. 11. 20 Newsgroups (electronic, medical): Words and Corresponding Weight Values.

two-class case (the nonspam class corresponds to both conference and jobs emails). Here we plot the top words that received highest weight for each class (discarding those without a clear meaning, e.g., abbreviations, acronyms, etc.). We observe that words reflecting the topic of a category receive a larger weight in the *other class*. For example, the words *sales*, *money*, *marketing*, *credit*, etc. get a larger weight in the non-spam category (their weights in the spam class are very close to zero). Similarly, the words *computational*, *neuroscience*, *neural*, *algorithms*, *deadline*, etc. receive larger weights in the spam category. The weights for these words in the nonspam class are very close to zero. While surprising at first, this trend may be due to the nature of the spam and nonspam email distributions. Each of these two categories is actually a combination of subclasses. The nonspam class in this case is the union of conference and jobs emails (by construction). Likewise, the spam messages can be very different in nature (sales, jokes, diets, fraud, etc.), and therefore different in their word content. As a consequence, the variance of feature values for words reflecting the general topic of a category is larger within the same category than in the other one (e.g., the word *sales* appears only in half of the spam messages, and does not appear in any of the nonspam emails). Since the weights computed by the LAC algorithm are inversely proportional to a measure of such variance of values (i.e., X_{ji}), we obtain the "swapping phenomenon" depicted in Figure 10. This analysis can be interpreted as the fact that the absence of a certain term (e.g., absence of the word *sales* within the nonspam messages) is a characteristic shared across the emails of a given category;

whereas the presence of certain words shows a larger variability across emails of a given category (e.g., the word *sales* appears only in half of the spam messages).

Figure 11 shows the weight values and corresponding words for the 20 Newsgroups (electronic, medical) dataset. In this case words receive largest weights within the representative class (e.g., *system*, *noise*, *circuit*, *range*, for the electronic class; *screen*, *hot*, *dead*, *cost*, *program* for the medical class). In this case, categories represent focused topics, and therefore words reflecting the content of documents show a small variance (e.g., the word *system* appears in all documents on electronics, and thus its variance is zero).

For this dataset, we also analyzed the dictionary of the corpus, and noticed that the majority of words are descriptive of the electronic category, while the medical domain is underrepresented. This bias was also reflected within the words that received larger weights: we could easily identify many words of the electronic domain, while words from the medical domains were less in number. Given the biased dictionary, this result is expected.

These results provide evidence that the weights computed by the WSBPA algorithm are meaningful, that is the averaging of weights performed by Equation (12) properly captures the local relevance of features. This is important for the cluster prediction of future data. Local weights also provide information regarding the subspace each cluster belongs to, thus allowing data interpretation, and possibly data compression. Specifically, for text categorization, the analysis of weights can be informative of the nature of the categorization problem, and can be used to guide the process of text interpretation. Of course, we are not advocating that local weights alone can solve the problem of automatic document annotation. Our results simply show that they are useful for the identification of descriptive words. Local weights alone, though, are not able to account for all possible configurations and words' distributions. For example, a word that appears in all documents of one class and in zero documents of the other, receives large weight in both (its variance is zero in both cases). Considering the frequency of occurrence within each class, may clarify which class the word is descriptive of. While this phenomenon was not observed in our data, one has to account for such instances in general. Considering relative frequencies of words that receive large weight in both classes is a viable solution.

8. MEASURES OF DIVERSITY AND ACCURACY

Diversity is an important aspect in building clustering ensembles. It is expected that the accuracy of the ensemble improves when a larger number of input clusterings is given, provided that the clusterings are diverse. Diversity in clustering ensembles is under investigation by many researchers. Here we study the interplay between accuracy and diversity for our ensemble techniques.

Fern and Brodley [2003] illustrate the importance of diversity for cluster ensemble accuracy. They measure diversity using NMI, a pairwise similarity measure that quantifies the information shared between two partitions. Let A and B be two partitions of n points into c_A and c_B clusters, respectively. Let's assume n_i^A represent the number of points in cluster i of A, n_j^B represent the

number of points in cluster j of B, and n_{ij} is the number of points shared by cluster i of A and cluster j of B. The NMI between A and B is a value in [0, 1] and is defined as follows [Strehl and Ghosh 2002]:

$$NMI(A, B) = \frac{\sum_{i=1}^{c_A} \sum_{j=1}^{c_B} n_{ij} \log \frac{n_{ij} n}{n_i^A n_j^B}}{\sqrt{\sum_{i=1}^{c_A} n_i^A \log \frac{n_i^A}{n} \sum_{j=1}^{c_B} n_j^B \log \frac{n_j^B}{n}}}. \tag{13}$$

Since NMI measures the similarity between two partitions, $(1 - NMI)$ gives the pairwise diversity. The pairwise measure of diversity, based on NMI, of an ensemble of L partitions is then defined as follows:

$$D_{NMI} = \frac{2}{L(L-1)} \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} (1 - NMI(P_i, P_j)), \tag{14}$$

where P_i and P_j are two of the L partitions.

Kuncheva and Hadjitodorov [2004] and Hadjitodorov et al. [2006] discuss diversity and accuracy measures in great depth. In particular, Hadjitodorov et al. [2006] investigate which diversity measure gives more accurate results. In all, six measures were examined. One is based on the Adjusted Rand Index (ARI), which measures the amount of departure from the assumption that any two clustering results have occurred by chance. ARI is a measure of similarity between two partitions, and is defined as follows:

$$t_1 = \sum_{i=1}^{c_A} \binom{n_i^A}{2}, \; t_2 = \sum_{j=1}^{c_B} \binom{n_j^B}{2}, \; t_3 = \frac{2t_1 t_2}{n(n-1)},$$

$$ar(A, B) = \frac{\sum_{i=1}^{c_A} \sum_{j=1}^{c_B} \binom{n_{ij}}{2} - t_3}{\frac{1}{2}(t_1 + t_2) - t_3}, \tag{15}$$

where $\binom{a}{b}$ is the binomial coefficient. A and B are two partitions of a dataset with n points, c_A and c_B are the number of clusters in partitions A and B respectively, n_i^A is the number of points in cluster i of partition A, n_j^B is the number of points in cluster j of partition B, and n_{ij} is the number of points cluster i of A and cluster j of B have in common. Since $ar()$ measures the similarity between two partitions, to compute the pairwise diversity one would consider $(1 - ar())$. Therefore, the measure of diversity, based on ARI, of an ensemble is defined as follows:

$$D_p = \frac{2}{L(L-1)} \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} (1 - ar(P_i, P_j)).$$

D_p measures the diversity of an ensemble with L partitions, where P_i, P_j are two such partitions.

Other measures evaluate individual (i.e, nonpairwise) diversities, by comparing individual clustering results with the ensemble result. One such measure is:

$$D_{np_1} = \frac{1}{L}\sum_{i=1}^{L}(1 - ar(P_i, P^*)),$$

where P_i and P^* are the individual clustering result and the ensemble result respectively, and L is the number of clustering members.

An additional measure focuses on the spread of diversity (with respect to P^*) of individual clusterings. It is defined as follows:

$$D_{np_2} = \sqrt{\frac{1}{L-1}\sum_{i=1}^{L}(1 - ar(P_i, P^*) - D_{np_1})^2}.$$

Using this measure, Hadjitodorov et al. [2006] discover that a larger spread is not strongly related to the ensemble accuracy. To take this result into account, another measure was introduced:

$$D_{ARI} = \frac{1}{2}(1 - D_{np_1} + D_{np_2}), \tag{16}$$

which considers both variability and accuracy. Assuming that the ensemble result is close to the true labeling, we can measure the accuracy of individual clusterings by measuring how close they are to the ensemble result. Thus, a larger value of $(1 - D_{np_1})$ means higher accuracy. At the same time, variability within the ensemble can be measured using D_{np_2}. Equation (16) achieves a trade-off between accuracy and variability.

Hadjitodorov et al. [2006] indicate that the most stable measures are D_{np_1} and D_{ARI}. The study focuses on the coassociation approach to construct consensus functions. The authors conclude that an ensemble selected through medium diversity will fare better than either randomly selected ensembles or those selected through maximum diversity.

Based on the findings discussed above, we investigate here the issue of diversity and accuracy in more detail for our ensemble techniques (WSPA, WBPA, and WSBPA). Our objective is to investigate which measure of diversity is the best indicator for a good ensemble accuracy, and what is the preferred level of diversity (high, medium, or low). Such findings would enable one to select, from a set of ensembles, the one that is most likely to provide good results. We consider two measures of diversity, one based on NMI as defined in (14), and one based on ARI as defined in (16). We observe that D_{NMI} is a pairwise diversity measure that does not depend on the ensemble methodology, while D_{ARI} is a nonpairwise diversity measure that depends on the ensemble methodology. Furthermore, we experiment with two methods to build a cluster ensemble: we run LAC with different values of h in one case, and with initial random centroids in the second case. In the following, we provide the details of the experiments, and discuss the results. The results obtained with random centroids are consistent with those obtained by varying h. Therefore, in the following, we omit the accuracy/diversity plots for random centroids.

8.1 Building Cluster Ensembles by Varying h

To study how accuracy relates with the chosen measures of diversity, we created 50 ensembles of size 15 by varying the value of h. As clustering algorithm, we always used LAC. For each of the 50 ensembles, we computed both measures of diversity D_{NMI} and D_{ARI}, and corresponding accuracy values. In details, we ran the following procedure:

(1) Run the LAC algorithm for $1/h = 1, \ldots, 50$;
(2) Repeat the following 50 times:
   (a) Sample 15 clusterings out of the 50 generated in 1;
   (b) Run WBPA, WSPA, WSBPA (using METIS) on the selected 15 clusterings;
   (c) Compute the diversity measures D_{NMI} as in (14) and D_{ARI} as in (16), for $L = 15$;
   (d) Compute the average accuracy of the ensemble components, both based on NMI and ARI, as follows:

$$Acc_{NMI} = \frac{1}{15}\sum_{i=1}^{15} NMI(P_i, P^T) \tag{17}$$

$$Acc_{ARI} = \frac{1}{15}\sum_{i=1}^{15} ar(P_i, P^T), \tag{18}$$

   where P^T is the target partition (according to the ground truth);
   (e) Compute the accuracy of the ensemble decision, both based on NMI and ARI, as follows:

$$Acc^*_{NMI} = NMI(P^*, P^T) \tag{19}$$

$$Acc^*_{ARI} = ar(P^*, P^T), \tag{20}$$

   where P^* is the ensemble partition, and P^T is the target partition.

Figures 12–20 show the results of accuracy vs. diversity for our nine datasets. To construct the plots, we proceeded as follows. We sorted the 50 D_{NMI} values in increasing order. Each D_{NMI} value was associated with the corresponding Acc_{NMI} and Acc^*_{NMI} values. We plotted the collection of two dimensional points (D_{NMI}, Acc_{NMI}) and (D_{NMI}, Acc^*_{NMI}), and connected them with a line. We proceeded similarly for the measures based on ARI. This procedure was performed for each of the three ensemble techniques WSPA, WBPA, and WSBPA. In Figures 12–20, the points marked with a "$*$" symbol correspond to (D_{NMI}, Acc_{NMI}) and (D_{ARI}, Acc_{ARI}). The points marked with an "open square" symbol correspond to (D_{NMI}, Acc^*_{NMI}) and (D_{ARI}, Acc^*_{ARI}). From the plots, we observe the following:

(1) Larger D_{NMI} (D_{ARI}) values give larger Acc_{NMI} (Acc_{ARI}) values and larger Acc^*_{NMI} (Acc^*_{ARI}) values, for all datasets and all the three ensemble methods. This result suggests that, to obtain good ensemble accuracy, a high level of diversity should be preferred. (The same trend was obtained when diversity was generated by means of random centroids.)

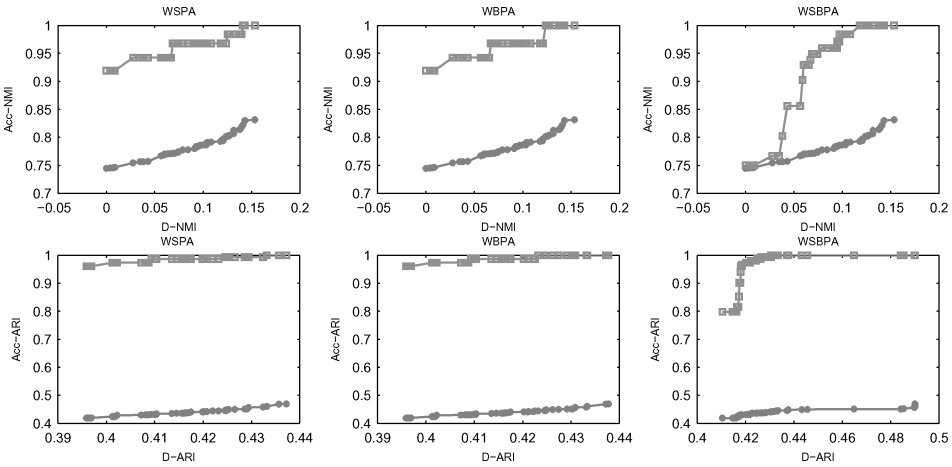

Fig. 12. Two Gaussian dataset: accuracy vs. diversity.

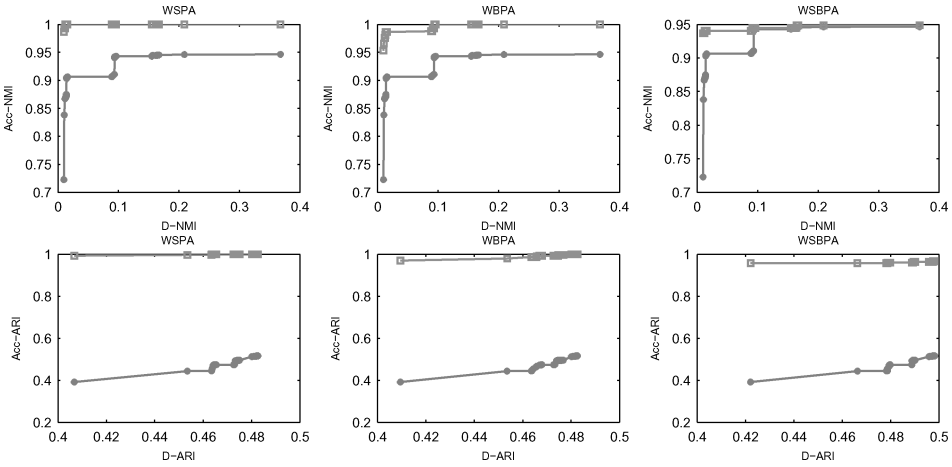

Fig. 13. Three Gaussian dataset: accuracy vs. diversity.

(2) For a given value of diversity D_{NMI} (D_{ARI}), the accuracy of the ensemble decision, Acc^*_{NMI} (Acc^*_{ARI}), is typically larger than the average accuracy of the ensemble components, Acc_{NMI} (Acc_{ARI}), for all three methods and all datasets (with few exceptions discussed below). This demonstrates the efficacy of our ensemble methods. Furthermore, the gain in accuracy, $Acc^*_{NMI} - Acc_{NMI}$ ($Acc^*_{ARI} - Acc_{ARI}$), in many cases is larger for larger diversity values (D_{NMI} and D_{ARI}, respectively). Again, this confirms that a high level of diversity should be preferred.

(3) *WDBC dataset and WSBPA ensemble method*: For lower values of diversity (both based on NMI and ARI), the accuracy of the ensemble decision is very low, and slightly below the average accuracy of the ensemble components. As diversity increases, the ensemble accuracy improves rapidly, and achieves significant improvement upon the components. This case stresses

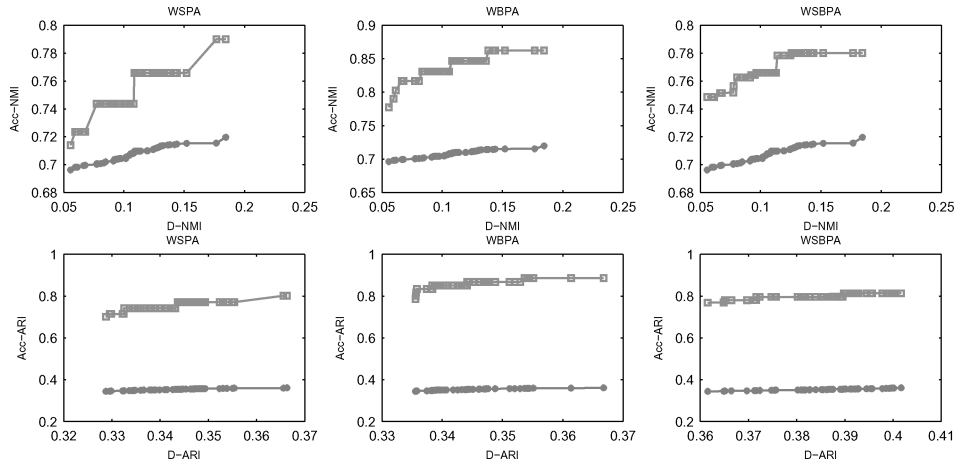

Fig. 14. Iris dataset: accuracy vs. diversity.

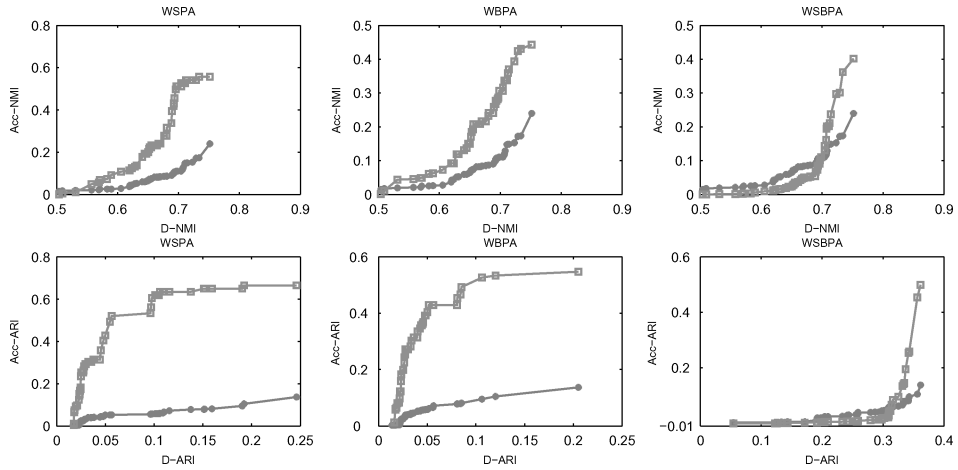

Fig. 15. WDBC dataset: accuracy vs. diversity.

the importance of high diversity. Note that, for this dataset, also the WSPA and WBPA techniques show a much larger accuracy gain for larger diversity values.

(4) Results similar to WDBC are observed for the Letter(A,B) dataset, and accuracy/diversity measures based on NMI.

(5) In general, given an ensemble of partitions, the average accuracy value of the components computed according to NMI (Acc_{NMI}) is higher than the average accuracy value computed according to ARI (Acc_{ARI}). This is because $NMI() \in [0, 1]$, while $ar() \in [-1, 1]$. Thus, the summation in (18) may contain negative values, which lead to smaller averages than in (17) (where the smallest components are zeros). On the other hand, the values Acc^*_{NMI} and Acc^*_{ARI}, which measure the accuracy of the ensemble partitions, are in general closer to each other. This happens because the largest value both

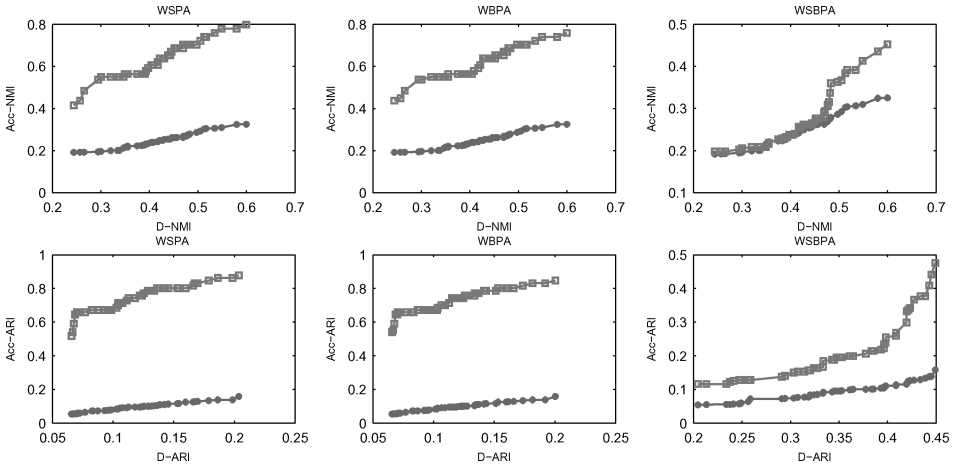

Fig. 16. Breast dataset: accuracy vs. diversity.

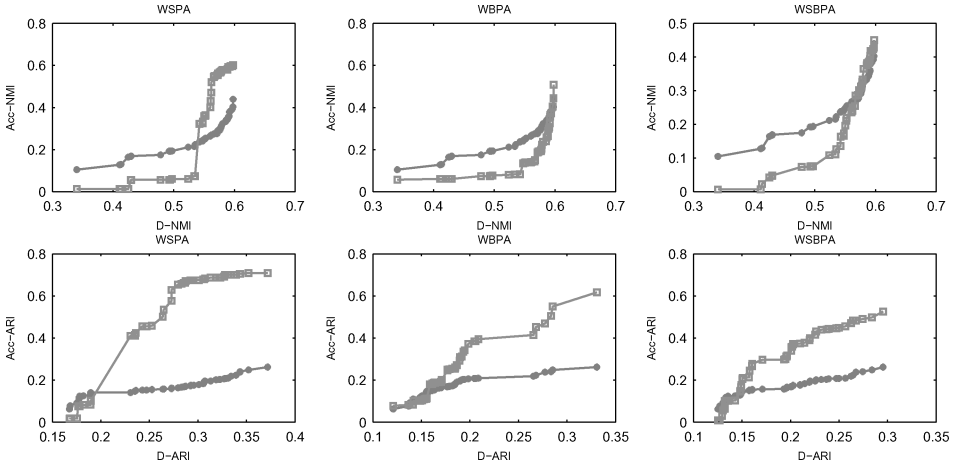

Fig. 17. Letter(A,B) dataset: accuracy vs. diversity.

for $NMI()$ and $ar()$, in (19) and (20) respectively, is 1. This different scaling of the accuracy/diversity measures causes the values (D_{NMI}, Acc^*_{NMI}) to lie below the (D_{NMI}, Acc_{NMI}) values for the SatImage dataset (while the opposite trend is observed for the measures based on ARI) (see Figure 18). We also observe that the range for the diversity values is very narrow in this case, suggesting the presence of correlated partitions in the ensembles. According to Table IX, our three ensemble techniques provide a smaller error rate than the minimum error rate of the input clusterings. This suggests that a measure of accuracy/diversity based on ARI might be more robust and consistent than a measure based on NMI. Nevertheless, it is important to keep in mind that D_{ARI} depends on the ensemble methodology. Thus, our findings are not necessarily applicable to other ensemble techniques or datasets.

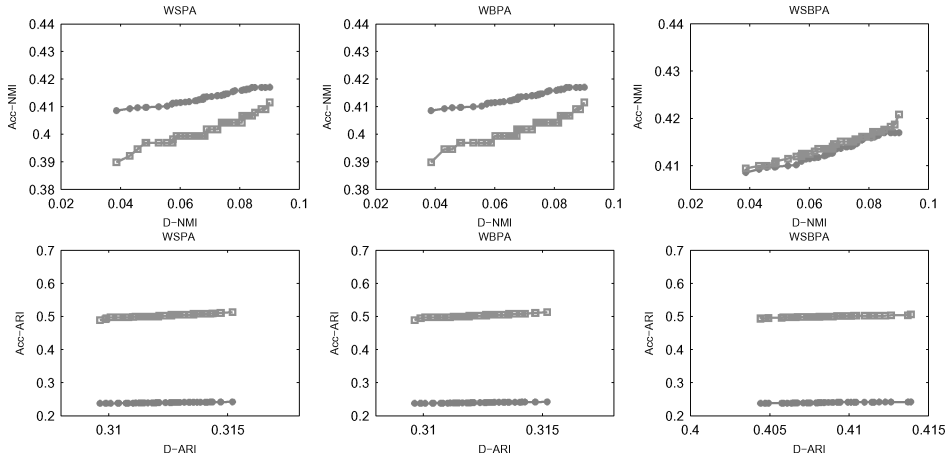

Fig. 18. SatImage dataset: accuracy vs. diversity.

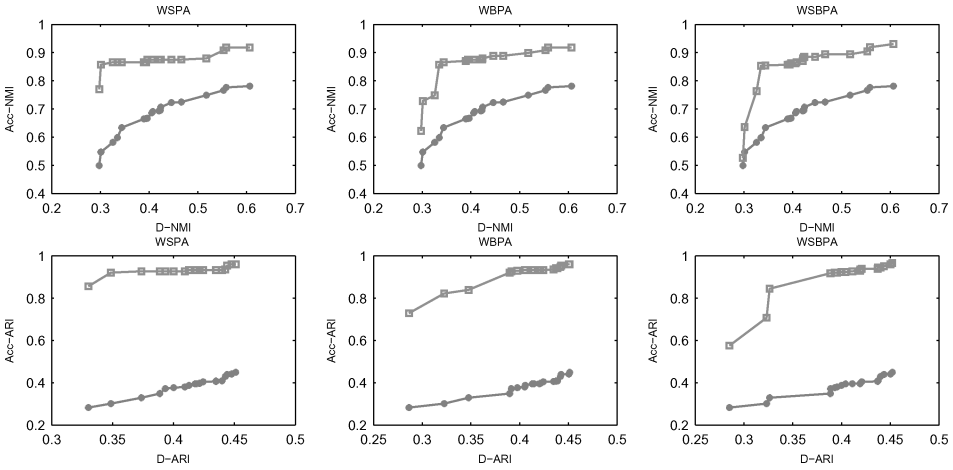

Fig. 19. Spam2000 dataset: accuracy vs. diversity.

9. CONCLUSIONS

This article discusses the challenges related to clustering due to its ill-posed nature. In particular, we address problems which arise from high-dimensional data, and issues due to parameter tuning. Our solutions make use of the ensemble methodology.

We have introduced three cluster ensemble techniques for subspace clustering. The experimental results show that our weighted clustering ensembles can provide solutions that are as good as or better than the best individual clustering, provided that the input clusterings are diverse. We have also demonstrated the use of our methods for the categorization of unlabeled documents. Furthermore, we addressed in depth the issue of diversity and accuracy. Our findings show that, typically, a high level of diversity should be preferred. Moreover, our results reveal that a diversity measure based on ARI is more robust and

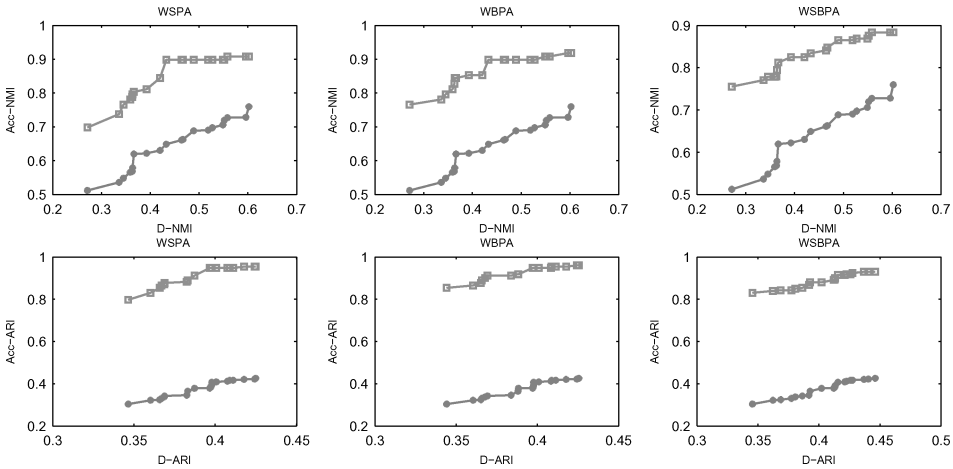

Fig. 20. Spam5996 dataset: accuracy vs. diversity.

consistent. We finally note that "universal" rules for choosing the preferred level of diversity should be used with caution, as the "optimal" level clearly depends on the consensus function and on the dataset. Our future research effort will focus on achieving a better understanding on which consensus function and which diversity-based ensemble selection method is more appropriate for which dataset.

ACKNOWLEDGMENTS

The authors would like to thank Ana Fred for providing the Matlab implementation of the EAC algorithm.

REFERENCES

AL-RAZGAN, M. AND DOMENICONI, C. 2006. Weighted clustering ensembles. In *Proceedings of the SIAM International Conference on Data Mining*. 258–269.

ASUNCION, A. AND NEWMAN, D. 2007. UCI Machine Learning Repository. http://www.ics.uci.edu/~mlearn/{MLR}epository.html.

AYAD, H. AND KAMEL, M. 2003. Finding natural clusters using multi-clusterer combiner based on shared nearest neighbors. In *Proceedings of the International Workshop on Multiple Classifier Systems*. 166–175.

DHILLON, I. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 269–274.

DOMENICONI, C., GUNOPULOS, D., MA, S., YAN, B., AL-RAZGAN, M., AND PAPADOPOULOS, D. 2007. Locally adaptive metrics for clustering high-dimensional data. *Data Min. Knowl. Discov. J. 14,* 1, 63–97.

DOMENICONI, C., PAPADOPOULOS, D., GUNOPULOS, D., AND MA, S. 2004. Subspace clustering of high-dimensional data. In *Proceedings of the SIAM International Conference on Data Mining*. 517–520.

DUDOIT, S. AND FRIDLYAND, J. 2003. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics 19,* 9, 1090–1099.

FERN, X. AND BRODLEY, C. 2003. Random projection for high-dimensional data clustering: A cluster ensemble approach. In *Proceedings of the International Conference on Machine Learning*. 63–74.

FERN, X. AND BRODLEY, C. 2004. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the International Conference on Machine Learning*. 281–288.

FRED, A. AND JAIN, A. 2002. Data clustering using evidence accumulation. In *Proceedings of the International Conference on Pattern Recognition*. 276–280.

FRED, A. AND JAIN, A. 2005. Combining multiple clusterings using evidence accumulation. *IEEE Trans. Patt. Analy. Mach. Intell. 27,* 6, 835–850.

GONDEK, D. AND HOFMANN, T. 2005. Non-redundant clustering with conditional ensembles. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 70–77.

GREENE, D., TSYMBAL, A., BOLSHAKOVA, N., AND CUNNINGHAM, P. 2004. Ensemble clustering in medical diagnostics. In *Proceedings of the 17th IEEE Symposium on Computer-Based Medical Systems*. 576–581.

HADJITODOROV, S., KUNCHEVA, L., AND TODOROVA, L. 2006. Moderate diversity for better cluster ensembles. *Inform. Fusion 7,* 3, 264–275.

HU, X. 2004. Integration of cluster ensemble and text summarization for gene expression analysis. In *Proceedings of the 4th IEEE Symposium on Bioinformatics and Bioengineering*. 251–258.

KANG, N., DOMENICONI, C., AND BARBARA, D. 2005. Categorization and keyword identification of unlabeled documents. In *Proceedings of the 5th IEEE International Conference on Data Mining*. 677–680.

KARYPIS, G. AND KUMAR, V. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Scient. Comput. 20,* 1, 359–392.

KULLBACK, S. AND LEIBLER, R. A. 1951. On information and sufficiency. *Annals Math. Statist. 22,* 1, 79–86.

KUNCHEVA, L. AND HADJITODOROV, S. 2004. Using diversity in cluster ensembles. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*. Vol. 2. 1214–1219.

KUNCHEVA, L. I., HADJITODOROV, S. T., AND TODOROVA, L. P. 2006. Experimental comparison of cluster ensemble methods. In *Proceedings of the International Conference on Information Fusion*. 1–7.

MANGASARIAN, O. L. AND WOLBERG, W. H. 1990. Cancer diagnosis via linear programming. *SIAM News 23,* 5, 1–18.

MINAEI-BIDGOLI, B., TOPCHY, A., AND PUNCH, W. 2004. A comparison of resampling methods for clustering ensembles. In *Proceedings of the International Conference on Machine Learning: Models, Technologies and Applications*. 939–945.

NG, A. Y., JORDAN, M. I., AND WEISS, Y. 2002. On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems*. Vol. 14. 849–856.

PARSONS, L., HAQUE, E., AND LIU, H. 2004. Subspace clustering for high-dimensional data: a review. *ACM SIGKDD Explor. Newslet. 6,* 1, 90–105.

PEKALSKA, E. 2005. The dissimilariy representations in pattern recognition. concepts, theory and applications. Ph.D. thesis, Delft University of Technology, Delft.

PUNERA, K. AND GHOSH, J. 2007. Soft cluster ensembles. In *Advances in Fuzzy Clustering and its Applications*, J. V. de Oliveira and W. Pedrycz, Eds. John Wiley & Sons, Ltd., 69–90.

STREHL, A. AND GHOSH, J. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Resea. 3,* 3, 583–617.

TOPCHY, A., JAIN, A., AND PUNCH, W. 2003. Combining multiple weak clusterings. In *Proceedings of the IEEE International Conference on Data Mining*. 331–338.

TOPCHY, A., JAIN, A., AND PUNCH, W. 2004. A mixture model for clustering ensembles. In *Proceedings of the SIAM International Conference on Data Mining*. 379–390.

TOPCHY, A., JAIN, A., AND PUNCH, W. 2005. Clustering ensembles: Models of consensus and weak partitions. *IEEE Trans. Patt. Anal. Mach. Intell. 27,* 12, 1866–1881.

Received August 2007; revised June 2008; accepted August 2008