Using Diversity in Cluster Ensembles

Ludmila I. Kuncheva
School of Informatics
University of Wales, Bangor
LL57 1UT, United Kingdom
mas00a@bangor.ac.uk

Stefan T. Hadjitodorov
CLBME, Bulgarian Academy of Sciences
Acad. G. Bonchev Street, Block 105
1113 Sofia, Bulgaria
sthadj@argo.bas.bg

Abstract − *The pairwise approach to cluster ensembles uses multiple partitions, each of which constructs a coincidence matrix between all pairs of objects. The matrices for the partitions are then combined and a final clustering is derived thereof. Here we study the diversity within such cluster ensembles. Based on this, we propose a variant of the generic ensemble method where the number of overproduced clusters is chosen randomly for every ensemble member (partition). Using three artificial sets we show that this approach increases the spread of the diversity within the ensemble thereby leading to a better match with the known cluster labels. Experimental results with three real data sets are also reported.*

Keywords: Pattern recognition, multiple classifier systems, cluster ensembles, diversity

1 Introduction

Combining the results of several clustering methods has recently appeared as one of the branches of multiple classifier systems [1, 4–9, 13, 15, 16]. The aim of combining partitions is to improve the quality and robustness of the results. According to the popular "no-free-lunch" allegory, there is no single clustering algorithm which performs best for all data sets. Choosing a single clustering algorithm for the problem at hand requires both expertise and insight, and this choice might be crucial for the success of the whole study. Selecting a clustering algorithm is more difficult than selecting a classifier. The difficulty comes from the fact that no ground truth is available against which to match the results. Therefore, instead of running the risk of picking an unsuitable clustering algorithm, a *cluster ensemble* can be used [15].

Diversity plays a significant, albeit difficult to quantify, role in classifier ensembles [11]. Recently, diversity of cluster ensembles has been looked at. Fern and Brodley [4] note that more diverse ensembles offer larger improvement on the individual accuracy than less diverse ensembles. (We call "accuracy" of a clustering algorithm the degree of match between the produced labels and some known cluster labels.)Here we use the Jaccard index to measure diversity between a pair of partitions. The lower the value, the higher the disagreement between the partitions. We develop further Fern and Brodley's result by looking at the relationship between diversity and the accuracy of the ensemble. Going a step further we propose a variant of the generic pairwise cluster ensemble approach which enforces diversity in the ensemble.

The next section gives the background of cluster ensembles. Section 3 explains four well known measures of similarity between two partitions which can also be used as diversity measures. Section 4 illustrates the possible benefit of larger diversity in the ensemble. We propose a variant of the traditional algorithm in Section 5. Conclusion remarks are given in Section 6.

2 Cluster ensembles

Let $P_1, \ldots, P_L$ be a set of partitions of a data set $\mathbf{Z}$. The aim is to find a resultant partition P^* which best represents the structure of $\mathbf{Z}$. There are two approaches to building cluster ensembles studied in the recent literature.

Pairwise cluster ensembles. Figure 1 shows a generic pairwise cluster ensemble algorithm. Given is a data set $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$. L ensemble members are generated by clustering $\mathbf{Z}$ or a subsample of it. For each clustering, a *co-association matrix* of size $N \times N$ is formed [1, 5–7], denoted $M^{(k)}$, termed sometimes connectivity matrix [13] or similarity matrix [15]. Its (i, j)th entry is 1 if $\mathbf{z}_i$ and $\mathbf{z}_j$ are in the same cluster in partition k, and 0, otherwise. A final matrix $\mathbf{M}$ is derived from $M^{(k)}$, $k = 1, \ldots, L$, called a 'consensus matrix' in [13]. The final clustering is decided using $\mathbf{M}$. The number of clusters may be pre-specified or found through further analysis of $\mathbf{M}$. Fern and Brodley [4] propose to use "soft" partitions and the resulting probabilities as the entries of $M^{(k)}$. The (i, j)th entry in $M^{(k)}$ is the probability that points $\mathbf{z}_i$ and $\mathbf{z}_j$ come from the same cluster in partition k. These probabilities are

*0-7803-8566-7/04/$20.00 ©2004 IEEE

calculated as

$$P^{(k)}(i,j) = \sum_{t=1}^{c} P(t|\mathbf{z}_i, k) \times P(t|\mathbf{z}_j, k)$$

where c is the number of clusters and $P(t|\mathbf{z}_i, k)$ is the probability for cluster t given object $\mathbf{z}_i$ in partition k. $P(t|\mathbf{z}_i, k)$ are produced by the clustering algorithm (e.g., EM for identifying Gaussian mixtures). $\mathbf{M}$ is obtained as the average across the k matrices $M^{(k)}$.

(PAIRWISE) CLUSTER ENSEMBLE ALGORITHM

1. Given is a data set $\mathbf{Z}$ with N elements. Pick the ensemble size L and the number of clusters c. Usually c is larger than the suspected number of clusters so there is "overproduction" of clusters.
2. Generate L partitions of $\mathbf{Z}$ with c clusters in each partition. Note that any of the itemized ways of building the individual partitions can be used here. For example, k-means clustering can be run from different initializations or with different subsets of features.
3. Form a co-association matrix for each partition, $M^{(k)} = \left\{m_{ij}^{(k)}\right\}$, of size $N \times N$, $k = 1, \ldots, L$, where

$$m_{ij}^{(k)} = \begin{cases} 1, & \text{if } \mathbf{z}_i \text{ and } \mathbf{z}_j \text{ are in the same cluster in partition } k, \\ 0, & \text{if } \mathbf{z}_i \text{ and } \mathbf{z}_j \text{ are in different clusters in partition } k \end{cases}$$

4. Form a final co-association matrix $\mathbf{M}$ (consensus matrix) from $M^{(k)}$, $k = 1, \ldots, L$, and derive the final clustering using this matrix.

Figure 1: A generic cluster ensemble algorithm.

Perhaps the simplest implementation of the generic cluster ensemble algorithm is as follows (called "voting c-means algorithm" in [5] and "evidence accumulation algorithm" in [6])

1. Pick the number of overproduced clusters c, the ensemble size L, and a threshold $\theta, 0 < \theta < 1$.
2. Run k-means L times, with c clusters, and form $M^{(1)}, \ldots, M^{(L)}$.
3. Calculate $\mathbf{M} = \frac{1}{L}\left(M^{(1)} + \ldots + M^{(L)}\right)$.
4. "Cut" $\mathbf{M}$ at threshold θ. Join in the same cluster all the points whose pairwise entry in $\mathbf{M}$ is greater than θ. For all the remaining points form single-element clusters.

This implementation is based on the majority vote. For $\theta = 0.5$, if points $\mathbf{z}_i$ and $\mathbf{z}_j$ have been in the same cluster in the majority of the L partitions, then they will be assigned to the same cluster in the final partition. The final number of clusters is not pre-specified; it depends on the threshold θ and on the number of overproduced clusters c. The combination of these two parameters is crucial for discovering the structure of the data. The rule of thumb is $c = \sqrt{N}$. Fred and Jain [6] also consider fitting a mixture of Gaussians to $\mathbf{Z}$ and taking the identified number of components as c. Neither of the two heuristics works in all the cases. Fred and Jain conclude that the algorithm can find clusters of any shape but it is not very successful if the clusters are touching.

The consensus matrix $\mathbf{M}$ can be regarded as a similarity matrix between the points on $\mathbf{Z}$. Therefore, it can be used with any clustering algorithm which operates directly upon a similarity matrix. In fact, "cutting" $\mathbf{M}$ at a certain threshold is equivalent to running the single link algorithm and cutting the dendrogram obtained from the hierarchical clustering at similarity θ. Viewed in this context, cluster ensemble is a type of *stacked clustering* whereby we can generate layers of similarity matrices and apply clustering algorithms on them.

Direct optimization in cluster ensembles. The labels that we assign to the c clusters are arbitrary. Thus two identical partitions might have permuted labels and be perceived as different partitions. Suppose that we can solve this correspondence problem between the partitions. Then the voting between the clusterers would be straightforward: just count the number of votes for the respective cluster. The problem is that there are $c!$ permutations of the labels and an exhaustive experiment might not be feasible for large c. Cluster ensemble methods with direct optimization have been proposed in [15, 17].

3 Diversity between partitions

3.1 Rand index.

Rand [14] proposes a simple measure of agreement between two partitions A and B. Denote by n_{11} the number of *pairs* of objects from $\mathbf{Z}$ which are both in the same cluster in A and are also both in the same cluster in B. Let n_{00} be number of *pairs* of objects from $\mathbf{Z}$ which are in different clusters in A and are also in different clusters in B. Both n_{00} and n_{11} are agreement quantities as in both partitions the pair of objects have been found to be either similar enough so as to be placed in the same cluster or dissimilar enough so as to be placed in different clusters. Accordingly, we can define the two disagreement quantities n_{01} and n_{10}. Rand index is

$$r(A,B) = \frac{n_{00} + n_{11}}{n_{00} + n_{11} + n_{01} + n_{10}} = \frac{2(n_{00} + n_{11})}{N(N-1)}. \quad (1)$$

Rand index takes value 1 if the partitions agree completely (regardless of the permutation of the labels) but does not have a constant value for the case when both partitions are drawn at random.

3.2 Jaccard index.

Using the same notation as for the Rand index, the *Jaccard index* between partitions A and B is [3]

$$J(A,B) = \frac{n_{11}}{n_{11}+n_{01}+n_{10}}. \quad (2)$$

3.3 Adjusted Rand index.

The adjusted Rand index corrects for the lack of a constant value of the Rand index when the partitions are selected at random [10]. Consider the confusion matrix for partitions A and B where the rows correspond to the clusters in A and the columns correspond to the clusters in B. Denote by N_{ij} the (i,j)th entry in this confusion matrix, where N_{ij} is the number of objects in both cluster i of partition A and cluster j in partition B. Denote by $N_{i.}$ the sum of all columns for row i; thus $N_{i.}$ is the number of objects in cluster i of partition A. Define $N_{.j}$ to be the sum of all rows for column i, i.e. $N_{.j}$ is the number of objects in cluster j in partition B. Suppose that the two partitions A and B are drawn randomly with a fixed number of clusters and a fixed number of objects in each cluster (generalized hypergeometric distribution). There is no requirement that the number of clusters in A and B should be the same. Let c_A be the number of clusters in A and c_B be the number of clusters in B. The expected value of the adjusted Rand index for this case is zero. The adjusted Rand index, ar, is calculated from the values N_{ij} of the confusion matrix for the two partitions as follows

$$t_1 = \sum_{i=1}^{c_A}\binom{N_{i.}}{2}; \quad t_2 = \sum_{j=1}^{c_B}\binom{N_{.j}}{2}; \quad (3)$$

$$t_3 = \frac{2t_1t_2}{N(N-1)}; \quad (4)$$

$$ar(A,B) = \frac{\sum_{i=1}^{c_A}\sum_{j=1}^{c_B}\binom{N_{ij}}{2} - t_3}{\frac{1}{2}(t_1+t_2)-t_3}, \quad (5)$$

where $\binom{a}{b}$ is the binomial coefficient $\frac{a!}{b!(a-b)!}$.

3.4 Mutual information.

A way to avoid relabeling is to treat the two partitions as (nominal) random variables, say X and Y, and calculate the mutual information between them [7, 15, 16]. The mutual information between partitions A and B is[1]

$$MI(A,B) = \sum_{i=1}^{c_A}\sum_{j=1}^{c_B}\frac{N_{ij}}{N}\log\left(\frac{N_{ij}N}{N_{i.}N_{.j}}\right) \quad (6)$$

A measure of similarity between partitions A and B is the *normalized* mutual information [7]

$$NMI(A,B) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B}N_{ij}\log\left(\frac{N_{ij}N}{N_{i.}N_{.j}}\right)}{\sum_{i=1}^{c_A}N_{i.}\log\left(\frac{N_{i.}}{N}\right)+\sum_{j=1}^{c_B}N_{.j}\log\left(\frac{N_{.j}}{N}\right)}$$

[1]By convention, $0\times\log(0)=0$.

If A and B are identical, then NMI takes its maximum value of 1. If A and B are independent, i.e., having complete knowledge of partition A we still know nothing about partition B and vice versa, then $NMI(A,B)\to 0$.

4 Diversity in classifier ensembles: an example

We consider two clustering algorithms: the standard k-means and a pairwise cluster ensemble with the following experimental protocols.

k-means. The number of clusters, c, was varied from 2 to 10. For each c we ran the clustering algorithm from 10 different initializations and chose the labeling with the minimum sum-of-squares criterion, J_e [2]. To determine the final number of clusters we took the minimum of the Xie-Beni index, $u_{XB}(c)$, across $c=2,\ldots,10$

$$u_{XB}(c) = \frac{\sum_{j=1}^{c}\sum_{\mathbf{z}\in C_j}||\mathbf{z}-\mathbf{v}_j||^2}{N(\min_{j\neq l}||\mathbf{v}_j-\mathbf{v}_l||^2)} \quad (7)$$

where $\mathbf{v}_j$ is the centroid of cluster C_j, $j=1,\ldots,c$, and $\mathbf{z}\in\mathbf{Z}$.

Pairwise cluster ensemble. The number of overproduced clusters, c, was set to 20 and the ensemble size was $L=25$. The individual ensemble members performed k-means starting from different initializations. Single link algorithm was run on the final co-association matrix $\mathbf{M}$. The largest "jump" in the distance at which two clusters are merged was found and this determined the final number of clusters. If this number appeared to be too large, we conjectured that there is no reasonable structure in the data and reassigned the final number of clusters to 1. We used a threshold of 80% of the total size of the data set, N. If the obtained number of clusters was greater than the threshold, we abstained from identifying a structure.

Figure 2 shows the results from applying the two clustering methods to three artificial data sets called four-gauss, easy-doughnut and difficult-doughnut, respectively. All three sets were generated in 2-D (as plotted) and then 10 more dimensions of uniform random noise were appended to each data set. The points which share a cluster label are joined. A total of 100 points were generated for each distribution.

Table 1 shows the values of the similarity/diversity measures introduced in Section 3 between the true and the guessed labels for the three examples. The ensemble method shows better performance on the two easier data sets, however the similarity measures disagree on the difficult-doughnut set.

To try to explain the inconsistent performance of the ensemble method we calculated diversity using the Jaccard index between every pair of partitions in the ensemble. By analogy with the kappa-error plots [4, 12]

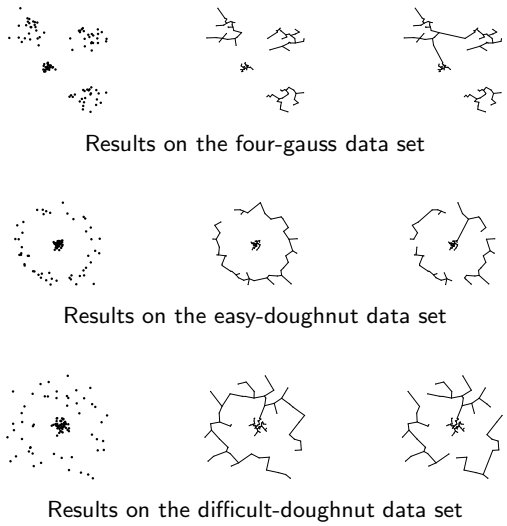

Figure 2: Data set (left plot), ensemble results (middle plot) and k-means results (right plot) for three data sets. 100 data points were generated for each distribution.

Table 1: Similarities between the original and the obtained labels for the example in Figure 2

Data set	Method	Rand	Jaccard	Adj Rand	NMI	final c
four-gauss	k-means	0.62	0.39	0.33	0.58	2
	ensemble	**1.00**	**1.00**	**1.00**	**1.00**	4
easy-doughnut	k-means	0.75	0.55	0.43	0.52	4
	ensemble	**1.00**	**1.00**	**1.00**	**1.00**	2
difficult-doughnut	k-means	**0.65**	0.49	**0.34**	**0.39**	3
	ensemble	0.49	**0.49**	0.00	0.00	1

we plotted each pair as a point in a 2-D space where the x-axis is the Jaccard index between the pair (the smaller the index, the more diverse the pair) and the y-axis is the "Accuracy" of the pair. The accuracy of partition C_i is the Jaccard index between the partition and the true labels. The y-coordinate of the pair is the averaged accuracy of the two partitions. Figure 3 shows the Diversity-Accuracy plots for the three data sets. We changed the ensemble parameter c (overproduced clusters) to take values 4, 8, and 20, hence there are three sets of points in each plot. The accuracy of the ensemble (Jaccard index between the ensemble output and the true labels) is shown in boldface above the respective number of overproduced clusters c.

Each plot is an illustration rather than a summary of an experiment because it is produced from three runs of the ensemble algorithm, one for each c. The figures for the two simple data sets show that the lower the agreement is (Jaccard index), the larger is the ensemble match of the true labels. The accuracies of the ensemble members vary significantly depending on the number of overproduced clusters. Note that, contrary to the similar intuition for classifier ensembles, in cluster ensembles the low individual accuracy does not seem to have a strong adverse effect on the ensemble performance. The difficult-doughnut plot (Figure 3, right) shows an inconsistent pattern. The best ensemble accuracy of 0.76 is achieved for $c = 8$, where the diversity is not as high as for $c = 20$ (the cloud for $c = 8$ is situated more to the right).

5 The proposed variant

In order to make use of both findings from the pilot example in Section 4, we propose a variant of the pairwise cluster ensemble algorithm. The difference from the generic algorithm in Figure 1 is that instead of selecting c in advance, we chose it at random for each ensemble member. The rationale for selecting c at random is to induce extra diversity in the ensemble.

Figure 4 shows the Diversity-Accuracy plot for the proposed variant and the three data set. The elongated clouds of points show that the ensembles contain variety of pairs, both with high diversity and low diversity. The pairs which have smaller Jaccard index (higher diversity) tend to be more accurate, which might contribute to the overall ensemble accuracy, as suggested by the results in Figure 3. The ensemble accuracy is shown in the plots in boldface. We note again that each plot gives the results of a single run, therefore a direct comparison between the accuracies in Figure 3 and Figure 4 is not justified.

We applied k-means, the standard pairwise cluster ensemble and the proposed method to 100 random generations from the distributions of the three data set. The averaged results are shown in Table 2. The maximum value for each measure is shown in boldface. A star next to it indicates that its difference with the second best measure is statistically significant ($\alpha = 0.05$).

Table 2 shows also the accuracies of the three clustering methods applied 100 times on three data sets from UCI

The results suggest that if there are distinguishable clusters in the data the proposed algorithm can identify them more accurately (in general) than k-means and the standard pairwise cluster ensemble algorithm. The results with the real data sets are rather disappointing. The reason for this might be that the class labels do not necessarily correspond to natural clusters. For the iris data, for example, there are two distinguishable clusters, one containing one of the classes and the other containing the remaining two classes. A successful clustering algorithm will report two clusters and will have low recognition rate if measured as the match with the true class labels. Therefore our view is that, in comparing clustering algorithms, more weight should be put on the results with data sets where test cluster structures are specifically designed or known.

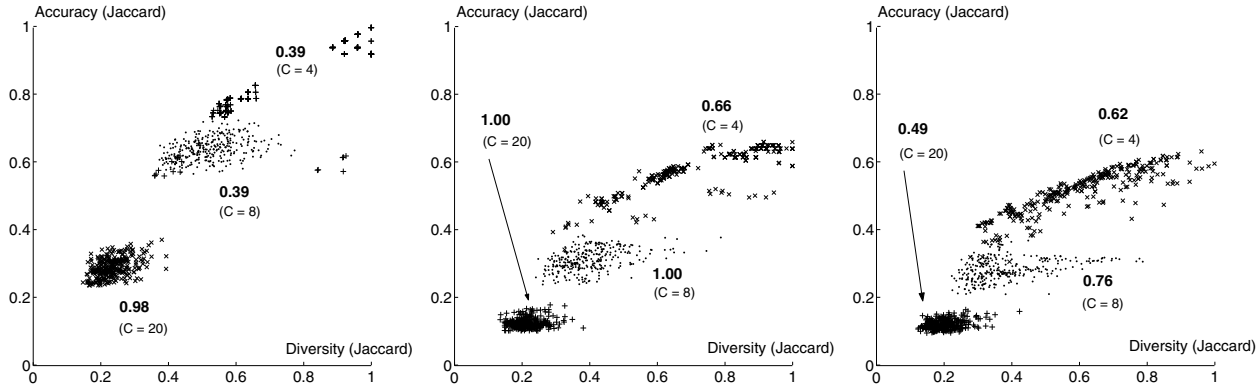

Figure 3: Diversity-Accuracy plot: four-gauss (left), easy-doughnut (middle) and difficult doughnut (right)

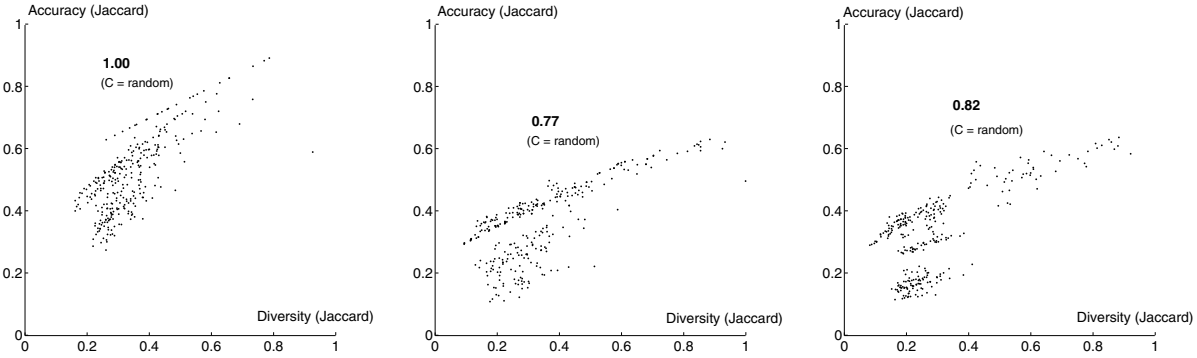

Figure 4: Diversity-Accuracy plot with the proposed pairwise cluster ensemble. four-gauss (left), easy-doughnut (middle) and difficult doughnut (right)

6 Conclusions

We propose a variant of the pairwise cluster ensemble paradigm where the number of overproduced clusters is chosen at random for each ensemble member. This modification is based on an anecdotal observation from an example with three artificial data sets: diverse ensembles tend to be more accurate than non-diverse ensembles even if the latter consist of more accurate individual members (Figure 3). The results with artificial and real data demonstrate the advantage of using diverse cluster ensembles (Table 2).

We note that in this study we were not aiming at finding the best possible clustering result but at showing how diversity can be exploited for improving on the clustering quality. Therefore we used the simplest possible clustering algorithm to design the ensemble members. More elaborate ensemble members may lead to a better overall cluster ensemble.

Another weak point of the proposed method, shared by the pairwise cluster ensemble paradigm in general, is that they do not scale well for large data sets. The co-association matrix $\mathbf{M}$ is of size $N \times N$ and running a single-link clustering on it may be too time-consuming. Using the threshold θ as in the majority vote algorithm may partly solve the problem. However, θ has to be specified and this requires additional analyses. We found that for $\theta = 0.5$ the results were inferior to those when the number of clusters was found by cutting the dendrogram at the largest distance jump.

The ensemble members can be built so as to take into account possible large dimensionality of the feature space. The solution proposed in [4] is to use random linear projections in lower-dimensional spaces and run the clustering in these spaces. Unlike data size scalability, feature size scalability can be incorporated in the proposed variant.

To alleviate the computational burden of the single link and improve the data size scalability of the pairwise cluster ensemble methods we are planning to investigate possible ways to select cluster prototypes. Thus the co-association matrix $\mathbf{M}$ will be much smaller. Another benefit from prototype selection would be that separate data sets may be used for building the ensemble members (object-distributed clustering).

An interesting further avenue is designing a cluster

Table 2: Similarities between the original and the obtained labels for the artificial and real data.

Data set	Features	Clusters/ classes, c	Method	Rand	Jaccard	Adjusted Rand	NMI	Guessed c
four-			k-means	0.68	0.48	0.43	0.64	2.30
gauss	12	2	ensemble	0.93	0.85	0.86	0.91	3.60
($N = 100$)			proposed	**0.97***	**0.94***	**0.94***	**0.96***	3.9
easy			k-means	0.74	0.55	0.49	0.56	4.55
doughnut	12	2	ensemble	**0.93**	**0.90**	**0.87**	**0.88**	2.29
($N = 100$)			proposed	**0.93**	0.87	0.86	**0.88**	2.83
difficult			k-means	0.70	0.52	0.40	0.47	3.94
doughnut	12	2	ensemble	0.63	0.57	0.26	0.29	2.84
($N = 100$)			proposed	**0.77***	**0.66***	**0.53***	**0.56***	3.61
			k-means	**0.60**	**0.34***	**0.24***	**0.38***	3.73
glass	9	6	ensemble	**0.60**	0.30	0.19	0.31	4.20
($N = 214$)			proposed	0.56	0.31	0.18	0.29	4.77
			k-means	0.76	0.57	0.54	0.66	2.00
iris	4	3	ensemble	**0.78**	0.59	0.56	**0.73**	2.41
($N = 150$)			proposed	**0.78**	**0.60**	**0.57**	**0.73**	2.00
			k-means	**0.67***	**0.47***	**0.37***	**0.43***	2.00
wine	13	3	ensemble	0.62	0.31	0.19	0.30	5.17
($N = 178$)			proposed	0.63	0.35	0.24	0.33	3.49

ensemble algorithm using the boosting philosophy, i.e., explicitly or constructively based on diversity.

Acknowledgment

This work was supported by a research grant number 15035 under the European Joint Project scheme, Royal Society, UK.

References

[1] H. Ayad and M. Kamel. Finding natural clusters using multi-clusterer combiner based on shared nearest neighbors. In *Proc. 4th International Workshop on Multiple Classifier Systems, MCS'03*, LNCS 2709, pages 166–175, Guildford, UK, 2003. Springer-Verlag.

[2] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, NY, second edition, 2001.

[3] A. Ben-Hur A. Elisseeff and I. Guyon. A stability based method for discovering structure in clustered data. In *Proc. Pacific Symposium on Biocomputing*, pages 6–17, 2002.

[4] X. Z. Fern and C. E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proc. 20th ICML*, pages 186–193, Washington, DC, 2003.

[5] A. Fred. Finding consistent clusters in data partitions. In *Proc. 2nd International Workshop on Multiple Classifier Systems, MCS'01*, LNCS 2096, pages 309–318, Cambridge, UK, 2001. Springer-Verlag.

[6] A. Fred and A.K. Jain. Data clustering using evidence accumulation. In *Proc. 16th International Conference on Pattern Recognition, ICPR*, pages 276–280, Canada, 2002.

[7] A.L.N. Fred and A.K. Jain. Robust data clustering. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, USA, 2003.

[8] J. Ghosh. Multiclassifier systems: Back to the future. In *Proc. 3d International Workshop on Multiple Classifier Systems, MCS'02*, LNCS 2364, pages 1–15, Cagliari, Italy, 2002. Springer-Verlag.

[9] K. Hornik. Clustrer ensembles. http://www.imbe.med.uni-erlangen.de/links/EnsembleWS/talks/Hornik.pdf

[10] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[11] L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles. *Machine Learning*, 51:181–207, 2003.

[12] D.D. Margineantu and T.G. Dietterich. Pruning adaptive boosting. In *Proc. 14th Int Conf on Machine Learning*, pages 378–387, San Francisco, 1997.

[13] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: A resampling based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52:91–118, 2003.

[14] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[15] A. Strehl and J. Ghosh. Cluster ensembles - A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–618, 2002.

[16] A. Strehl and J. Ghosh. Cluster ensembles - A knowledge reuse framework for combining partitionings. In *Proceedings of AAAI*. AAAI/MIT Press, 2002.

[17] A. Weingessel, E. Dimitriadou, and K. Hornik. An ensemble method for clustering, 2003. Working paper, http://www.ci.tuwien.ac.at/Conferences/DSC-2003/.