Experimental Comparison of Cluster Ensemble Methods

L.I. Kuncheva
School of Informatics
University of Wales, Bangor
Bangor, Gwynedd, UK
l.i.kuncheva@bangor.ac.uk

S.T. Hadjitodorov, L.P. Todorova
Central Laboratory of Biomedical Engineering
(CLBME) Bulgarian Academy of Sciences
Sofia, BULGARIA
sthadj@argo.bas.bg, lpt@clbme.bas.bg

Abstract* – Cluster ensembles are deemed to be a robust and accurate alternative to single clustering runs. 24 methods for designing cluster ensembles are compared here using 24 data sets, both artificial and real. Adjusted Rand index and classification accuracy are used as accuracy criteria with respect to a known partition assumed to be the "true" one. The data sets are randomly chosen to represent medium-size problems arising within a variety of biomedical domains. Ensemble size of 10 was considered. It was found that there is a significant difference among the compared methods (Friedman's Two Way ANOVA). The best ensembles were based on k-means individual clusterers. Consensus functions interpreting the consensus matrix of the ensemble as data, rather than similarity, were found to be significantly better than the traditional alternatives, including CSPA and HGPA.

Keywords: Clustering, Cluster ensembles, Consensus functions, Experimental comparison, Biomedical data

1 Introduction

Cluster ensembles have been found to be more accurate than single clustering algorithms [8–12,14]. More importantly, they exempt the user from deciding on a particular clustering algorithm, thereby from running the risk of a poor choice. While in classification, the adequacy of the chosen algorithm is clear from the estimated accuracy of the classifier, in clustering, a bad choice of algorithm may compromise the whole study. Thus the common consensus seems to be that a random choice of an ensemble is not as hazardous as a random choice of a single clustering method [11,20]. The next question is how to reduce the uncertainty in the choice of an ensemble design strategy.

The paper reports the results from an experimental comparison of ensemble design methods on 24 data sets. The rest of the paper is organised as follows. Section 2 details the ensemble design methods. Section 3 describes the data sets focusing on those which have not been used as benchmarks hitherto. Section 4 gives the experimental protocol and the results. Section 5 concludes the paper.

2 Cluster ensembles

There are two main issues in designing cluster ensembles: (1) the design of the individual "clusterers" so that they form potentially an accurate ensemble, and (2) the way the outputs of the clusterers are combined to obtain the final partition, called the *consensus* function. In some ensemble design methods the two issues are merged into a single design procedure, e.g., when one clusterer is added at a time and the overall partition is updated accordingly (called the direct or greedy approach). In this study we consider the two tasks separately so that we can study possible matches between individual clusterer design and consensus functions.

2.1 The individual clusterers

Both diversity within the ensemble and accuracy of the individual clusterers are important factors [5,12,13], although not straightforwardly related to the ensemble accuracy. Many heuristics have been proposed in order to achieve diverse clusterers using the same clustering method [4,7,12,22]. Using a *random number of target clusters* for each ensemble member has been found to be one of the most successful heuristics, therefore we use it in all ensemble designs here. The number of target clusters was randomly chosen between 2 and 22. The choice of this interval was guided by our pilot experiments with a small number of data sets. All the ensembles studied here were homogeneous, i.e., all clusterers were created using the same clustering method, in one of the following ways:

(a) *k-means* with random initialisations
(b) *k-means* with random initialisations and using *random sub-samples* of the data
(c) *single linkage* using different random sub-samples
(d) *mean linkage* using different random sub-samples.

2.2 The consensus function

The consensus function aggregates the outputs of the individual clusterers into a single partition. Many consensus functions use the consensus matrix obtained from the adjacency matrices of the individual clusterers. Let *N* be the number of objects in the data set. The adjacency matrix for clusterer *k* is an *N* by *N* matrix with entry (*i*,*j*) = 1 if objects *i* and *j* are placed in the same cluster by clusterer *k*, and (*i*,*j*) = 0, otherwise. The overall consensus matrix, **M**, is the average of the adjacency matrices of the clusterers [8,17,20]. Its (*i*,*j*) entry gives the proportion of clusterers which put *i* and *j* in the same cluster.

Here we examined the following consensus functions

(i) *single linkage* on **M** *(similarity)*. The overall consensus matrix, **M**, is interpreted as similarity between the objects. Then 1 – **M** can be thought of as distance and used as input of a single linkage clustering. The result is taken to be the ensemble partition.

(ii) *single linkage* on **M** *(data)*. Here the overall consensus matrix, **M**, is interpreted as "data". Each object is represented by *N* features, i.e., the *j*-the feature for object *i* is the (*i*,*j*) entry of **M**. Using similarities as features has been demonstrated to work well [18], and is also related to the idea which underpins the SVM classifier.

(iii) *mean linkage* on **M** *(data)*.

(iv) *k-means* on **M** *(data)*.

(v) *CSPA* on **M** *(similarity graph)*. The Cluster-based Similarity Partition Algorithm [20] treats **M** as a graph with the *N* objects as the vertices and the similarities being the weights. CSPA uses a graph partitioning algorithm, METIS. It reduces the size of the graph by collapsing vertices and edges, partitions the smaller graph, and then un-coarsens it to construct a partition for the original graph, which represents the ensemble output.

(vi) *HGPA* on the set of individual adjacency matrices. The HyperGraph-Partitioning Algorithm [20] works by constructing a hypergraph from all individual adjacency matrices. Thus two vertices *i* and *j* are connected by as many edges as there are 1's as the (*i*,*j*) entry of the adjacency matrices. The hyperfraph is then partitioned using an algorithm called HMETIS.

Consensus function (i) is standard, straightforward and a common choice for cluster ensembles. Functions (v) and (vi) are less straightforward but have been demonstrated to be robust and accurate. (Provision of the Matlab code by Strehl and Ghosh has been great help in this respect.) On the other hand, consensus functions (iii), (iv) and (v) are unusual as **M** is treated, rather counter-intuitively, as data. Our reasons to include these functions here was their good performance in the pilot experiments.

As in many similar studies, we shall assume that the number of true clusters is given and is available to the consensus function. Specifically, consensus functions (iv), (v) and (vi) require this number as an input parameter. On the other hand, functions (i), (ii) and (iii) can provide an estimate of the number of clusters by examining the criterion function and finding the cluster structure before the "largest jump".

2.3 The cluster ensembles

Table 1 shows the ensembles examined here.

Table 1. Ensemble designs examined here

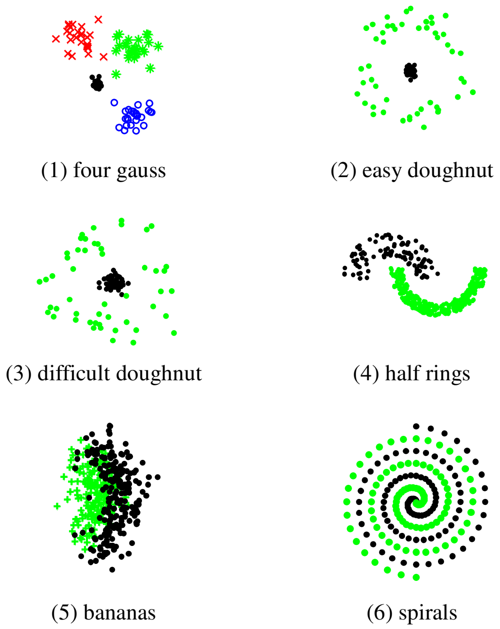
Ens	Individual clusterers	Consensus function
1	(a) k-means	(iii) mean linkage + data
2	(c) single linkage	(iii) mean linkage + data
3	(d) mean linkage	(iii) mean linkage + data
4	(b) k-means + subsample	(iii) mean linkage + data
5	(a) k-means	(iv) k-means + data
6	(c) single linkage	(iv) k-means + data
7	(d) mean linkage	(iv) k-means + data
8	(b) k-means + subsample	(iv) k-means + data
9	(a) k-means	(vi) HGPA
10	(c) single linkage	(vi) HGPA
11	(d) mean linkage	(vi) HGPA
12	(b) k-means + subsample	(vi) HGPA
13	(a) k-means	(v) CSPA
14	(c) single linkage	(v) CSPA
15	(d) mean linkage	(v) CSPA
16	(b) k-means + subsample	(v) CSPA
17	(a) k-means	(ii) single linkage + data
18	(c) single linkage	(ii) single linkage + data
19	(d) mean linkage	(ii) single linkage + data
20	(b) k-means + subsample	(ii) single linkage + data
21	(a) k-means	(i) single linkage + similarity
22	(c) single linkage	(i) single linkage + similarity
23	(d) mean linkage	(i) single linkage + similarity
24	(b) k-means + subsample	(i) single linkage + similarity

3 Data sets

The data sets in this study were chosen to represent a variety within a specific class of data sets characterised by: (1) small number of true classes, which may or may not correspond to coherent clusters; (2) moderate number of observations (up to few hundred); (3) moderate number of features (typically 5 to 30). Such data sets are collected, for example, in clinical medicine for pilot research studies. Thus we picked mostly biomedical data as the real data sets.

3.1 Artificial data

Figure 1 displays the 6 artificial data sets with the "ground truth" clusters to be discovered by the clustering algorithms. Data sets (4)-(6) were used in 2 dimensions as shown. Ten dimensions of uniform random noise with uniform distribution were added to each of data sets (1)-(3), so they were used in the experiments as 12-dimensional.



(1) four gauss

(2) easy doughnut

(3) difficult doughnut

(4) half rings

(5) bananas

(6) spirals

Figure 1. The six artificial data sets

Since all dimensions of all data sets were normalised to a mean of zero and standard deviation 1, the noise for sets (1)-(3) became the predominant component and the clustering results were relatively poor. Set (5) almost represents the "impossible problem" as there are no distinguishable clusters there. However, the points are generated from two aligned banana-shaped curves with Gaussian noise around the points on the curves. Thus the density distribution within the clouds is supposed to be the guide to the desired clustering result. Sets (4) and (6) have been used in comparative studies similar to ours.

3.2 Real data

A summary of the real data is given in Table 2. The notations in the tables are: N: number of data points, n: number of features, c: number of classes. We assume that classes correspond to clusters. Unsubstantiated as this assumption is, we chose to make it, as many authors have done elsewhere. The problem is that there is no reasonable way to establish the "true" number clusters, and the alternative would be to leave out all the experiments with real data.

Ten of the real data sets are taken from the UCI Machine Learning Repository[1] [2]. The crabs data set is used in [19]. The remaining 7 data sets are new. They have been used in recent medical studies as described below[2].

•*Contractions.* This data set comes from wireless capsule endoscopy [23]. The problem is to detect intestinal contractions in video images sent by a small capsule travelling along the intestinal tract. Twenty seven features were extracted using basic image descriptors. The 98 objects (49 in each class) were manually selected to represent the most clear examples of the classes.

Table 2. Details of the real data sets

Dataset	N	n	c	Source
breast	277	9	2	UCI
contractions	98	27	2	CVC, Barcelona
crabs	200	7	2	Ripley []
ecoli	336	7	8	UCI
heart	270	13	2	UCI
iris	150	4	3	UCI
laryngeal_2	213	16	2	CLBME, Sofia
laryngeal_3	353	16	3	CLBME, Sofia
leukaemia	38	7129	2	UCI
liver	345	6	2	UCI
lymph	148	18	4	UCI
pima	768	8	2	UCI
respiratory	85	17	2	CLBME, Sofia
soybean_large	266	35	15	UCI
thyroid	215	5	3	UCI
voice_3	238	10	3	CLBME, Sofia
voice_9	428	10	9	CLBME, Sofia
weaning	302	17	2	CLBME, Sofia

• *Laryngeal-2 and -3 and Voice-3 and -9.* These data sets consist of feature vectors representing voice signals of patients suffering from laryngeal diseases [3]. Different parameters are used in Laryngeal and Voice, respectively. The number following the data set name denotes the number of classes.

• *Respiratory*. The set consists of the clinical records (17 features) for 85 newborn children with two types of respiratory distress syndrome (RDS):- Hyaline Membrane Disease (HMD) and non-HMD. The two classes need urgent and completely different treatments, therefore an early and accurate RDS classification is crucial within the first few hours after delivery.

• *Weaning*. This set consists of data from a retrospective study of 151 patients suffering from acute respiratory insufficiency on a long-term (at least 7 days) mechanical ventilation [21]. Each case is described by 17 clinical and preclinical features. Two classes of patients have formed: not ready for weaning and ready for weaning from mechanical ventilation. Each patient is an instance in both classes as their parameters were measured once before weaning and once at the start of weaning.

4 Experiment

4.1 Protocol

As all ensemble methods rely on a random element, we built 100 ensembles for each data set and each ensemble method. Each ensemble consisted of 10 clusterers. Two random parameters were involved. The first was random choice of target number of clusters for each ensemble

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html

[2]http://www.informatics.bangor.ac.uk/~kuncheva/
activities/real_data.htm

member. The second random parameter is either random initialization (k-means) or random subsample (single- and mean- linkage methods), as explained in 2.1. The accuracy of an ensemble was measured as the agreement between the ensemble partition and the "true" partition. We used two measures: the adjusted Rand index (AR) [16] and the classification accuracy (CA). AR evaluates the dependence between two partitions. If they are formed completely independently of one another AR takes values close to 0 (small negative values are also possible). The maximum value of AR is 1, and is achieved for identical partitions. The classification accuracy, CA, is commonly used for evaluating clustering results. To guarantee the best re-labelling of the clusters, CA is computed in the following way. Consider the true class label of each object. Re-label each cluster produced by the ensemble with the class label most represented among the members of this cluster. Store the number of objects from the majority class, as they will receive correct labels. The proportion of correctly labelled objects in the data set is CA for this partition.

4.2 Results

Our first test was Friedman's Two-way ANOVA in order to determine whether the ensemble methods were significantly different. Denote by $AR(i,j)$ the averaged AR across the 100 runs for ensemble number j and data set number i. Thus the results were organised in a 24 by 24 matrix, $AR(i,j)$, where the data sets corresponded to the rows (blocks) and the ensemble methods corresponded to the columns (treatments). The ranks of the methods were calculated separately for each data set. The entries in the row for the data set were sorted in descending order, the first was assigned rank 1, the second – rank 2, etc. If the ensemble methods were equivalent, then their ranks would be close to random for the different data sets. To test this hypothesis, we calculate the test statistic

$$\chi_r^2=\frac{12}{nk(k+1)}\left[\sum_{j=1}^{k}\left(\sum_{i=1}^{n}R_{ij}\right)^2\right]-3n(k+1) \quad (1)$$

where n is the number of rows, k is the number of columns (both 24 in our case), and R_{ij} is the rank in cell (i,j). If the ranks were random, their sums would be approximately equal. If k and n are not too small, χ_r^2 has a chi-square distribution. If the computed statistic is greater than the tabulated value with k – 1 degrees of freedom, then we reject the hypothesis that the ensemble methods are equivalent.

We obtained $\chi_r^2 = 92.8$, hence significant difference at 23 degrees of freedom with $p < 0.0001$.

Let $CA(i,j)$ be the classification accuracy averaged across the 100 runs for ensemble number j and data set number i. Friedman's Two-way ANOVA was applied to both AR and CA. For CA we obtained $\chi_r^2 = 107.7$ which also allows us to reject the hypothesis that the ensemble methods are equivalent at $p < 0.0001$.

The confirmed differences require a further analysis to determine which ensemble method or methods are the best.

Let $s_{AR}(i,j)$ and $s_{CA}(i,j)$ be the respective standard deviations for AR and CA. The 95% confidence interval for the mean AR can then be calculated as

$$\left[AR(i,j)-1.96\frac{s_{AR}(i,j)}{\sqrt{100}},\, AR(i,j)+1.96\frac{s_{AR}(i,j)}{\sqrt{100}}\right]$$

Thus we can evaluate the statistical significance of the difference between any two ensemble methods on any data set. To compare a pair of ensemble methods, x and y, we count the number of data sets (out of 24) for which x has been significantly better than y. Denote this number by $b(x,y)$. Significant difference for a given data set, k, is observed if the respective confidence intervals have no intersection, i.e.,

$$AR(k,x)-0.196\times s_{AR}(k,x) > AR(k,y)+0.196\times s_{AR}(k,y)$$

Respectively, we get the number of data sets where x has been significantly worse than y, denoted $w(x,y)$, and then $s(x,y) = 24 - b(x,y) - w(x,y)$ is the number of data sets where the difference is not statistically significant. Comparisons for all pairs of ensemble methods were done for both criteria, AR and CA. Finally, an index of total performance was produced for each method as

$$t(i)=\sum_{j=1}^{24}b(i,j)-w(i,j), \qquad i=1,\dots,24$$

Tables 3 and 4 display the results for AR and CA, respectively. The ensemble methods are sorted by $t(i)$. The best method is the one at the top of the table. The sum of ranks for the methods across the 24 data sets are also provided (used in (1)).The total indices for the methods are visualised as bar graphs in Figure 2.

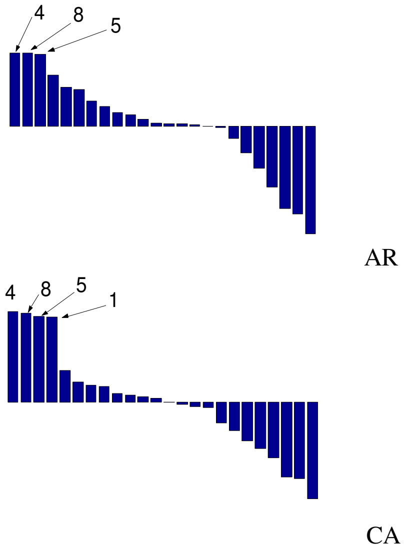


Figure 2.Bargraph of sorted $t(i)$ for the 24 ensemble methods

Table 3. Results using adjuster Rand index (AR) (times better, times worse, total index *t* and rank)

Ens	*b*	*w*	*t*	Rank
4	303	102	201	206.5
8	302	102	200	192.5
5	302	105	197	191.5
1	264	124	140	223.0
7	262	156	106	245.0
15	278	179	99	256.5
3	266	198	68	270.0
17	229	176	53	281.5
21	212	176	36	282.5
2	240	209	31	283.0
16	227	210	17	294.0
20	197	189	8	284.0
12	217	212	5	311.5
13	224	219	5	296.0
24	189	186	3	299.0
14	225	226	-1	298.5
9	218	224	-6	297.5
6	204	239	-35	320.5
19	186	261	-75	332.5
11	164	281	-117	343.5
18	148	316	-168	390.5
10	118	346	-228	413.0
23	114	357	-243	428.0
22	92	388	-296	459.5

Table 4. Results using classification accuracy (CA) (times better, times worse, total index *t* and rank)

Ens	*b*	*w*	*t*	Rank
4	332	88	244	163.0
8	326	85	241	177.0
5	328	96	232	184.0
1	329	99	230	182.5
3	262	176	86	259.5
17	228	174	54	257.5
21	228	182	46	280.5
15	233	193	40	274.5
16	218	195	23	291.5
20	212	195	17	288.0
7	198	184	14	296.0
13	213	204	9	296.5
12	210	212	-2	308.5
24	185	193	-8	294.5
9	208	222	-14	303.5
2	215	231	-16	298.0
19	190	247	-57	333.5
14	184	263	-79	342.5
11	158	263	-105	350.0
6	153	279	-126	361.5
18	151	303	-152	386.0
23	123	328	-205	414.0
10	110	319	-209	413.0
22	104	367	-263	444.5

To evaluate which ensembles form a group of undistinguishable accuracy, we applied Friedman's Two-Way ANOVA to a sequence of nested sets of ensemble methods. Using the ordering in Table 3, the first set contained ensembles 4 and 8. We calculated the χ_r^2 statistic and plotted it in Figure 3 against the degrees of freedom (set size minus one, so x = 1 because k = 2 (n=24) in eq. (1)). To evaluate the statistical significance of the difference, we plotted also the tabular values of chi-square at degrees of freedom k – 1 for p = 0.05 and p = 0.0001. The next set of methods, {4,8,5}, produced the point plotted at degrees of freedom 2 (k = 3, n = 24), and so on. The number next to each dot in the plot is the ensemble number entering the set. The Figure shows that statistically significant difference between methods, at significance p = 0.0001 appears only after including ensemble 10 in the group. Thus the difference between the ensembles shown in rows 1 to 21 in Table 3 is not significant at p = 0.0001. Significant difference will be observed at p = 0.05 when ensemble 2 enters the set. The analogous result for CA, using Table 4 is plotted in Figure 4.

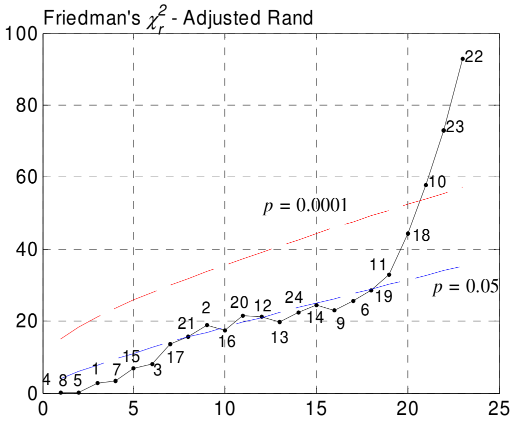


Figure 3. χ_r^2 (AR criterion) and tabulated chi-square values plotted against the degrees of freedom for nested sets of ensemble methods: {4,8}, {4,8,5}, {4,8,5,1}, etc.

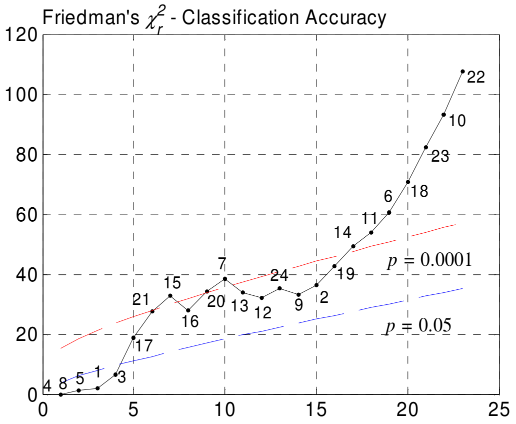


Figure 4. χ_r^2 (CA criterion) and tabulated chi-square values plotted against the degrees of freedom for nested sets of ensemble methods: {4,8}, {4,8,5}, {4,8,5,1}, etc

Table 5. Ensemble designs sorted by the total index of performance, $t(i)$, using AR

Ens	Individual clusterers	Consensus function
4	(b) k-means + subsample	(iii) mean linkage + data
8	(b) k-means + subsample	(iv) k-means + data
5	(a) k-means	(iv) k-means + data
1	(a) k-means	(iii) mean linkage + data
7	(d) mean linkage	(iv) k-means + data
15	(d) mean linkage	(v) CSPA
3	(d) mean linkage	(iii) mean linkage + data
17	(a) k-means	(ii) single linkage + data
21	(a) k-means	(i) single linkage + similarity
2	(c) single linkage	(iii) mean linkage + data
16	(b) k-means + subsample	(v) CSPA
20	(b) k-means + subsample	(ii) single linkage + data
12	(b) k-means + subsample	(vi) HGPA
13	(a) k-means	(v) CSPA
24	(b) k-means + subsample	(i) single linkage + similarity
14	(c) single linkage	(v) CSPA
9	(a) k-means	(vi) HGPA
6	(c) single linkage	(iv) k-means + data
19	(d) mean linkage	(ii) single linkage + data
11	(d) mean linkage	(vi) HGPA
18	(c) single linkage	(ii) single linkage + data
10	(c) single linkage	(vi) HGPA
23	(d) mean linkage	(i) single linkage + similarity
22	(c) single linkage	(i) single linkage + similarity

Table 6. Ensemble designs sorted by the total index of performance, $t(i)$, using CA

Ens	Individual clusterers	Consensus function
4	(b) k-means + subsample	(iii) mean linkage + data
8	(b) k-means + subsample	(iv) k-means + data
5	(a) k-means	(iv) k-means + data
1	(a) k-means	(iii) mean linkage + data
3	(d) mean linkage	(iii) mean linkage + data
17	(a) k-means	(ii) single linkage + data
21	(a) k-means	(i) single linkage + similarity
15	(d) mean linkage	(v) CSPA
16	(b) k-means + subsample	(v) CSPA
20	(b) k-means + subsample	(ii) single linkage + data
7	(d) mean linkage	(iv) k-means + data
13	(a) k-means	(v) CSPA
12	(b) k-means + subsample	(vi) HGPA
24	(b) k-means + subsample	(i) single linkage + similarity
9	(a) k-means	(vi) HGPA
2	(c) single linkage	(iii) mean linkage + data
19	(d) mean linkage	(ii) single linkage + data
14	(c) single linkage	(v) CSPA
11	(d) mean linkage	(vi) HGPA
6	(c) single linkage	(iv) k-means + data
18	(c) single linkage	(ii) single linkage + data
23	(d) mean linkage	(i) single linkage + similarity
10	(c) single linkage	(vi) HGPA
22	(c) single linkage	(i) single linkage + similarity

It appears that CA is more sensitive than AR in terms of finding differences between ensemble methods. It is interesting to observe that the ordering of the methods is similar but not identical for the two criteria.

To help evaluating the design options for the ensemble methods, Tables 5 and 6 reproduce the descriptions in Table 1 sorted by the total index of ensemble performance, $t(i)$, for AR and CA, respectively. The methods which fetch a significant difference at p = 0.0001 are separated with a double line. The highlighted cells contain the methods which are not distinguishable as the best group, at $p = 0.05$.

Tables 5 and 6 reveal some interesting results

(1) The standard single linkage method appears to be the worst choice for the individual clusterers. Apart from method 2, which apparently benefits from a successful consensus function, all ensembles based on single linkage fall in the bottom part of the tables.

(2) The best ensemble methods use k-means for the individual clusterers. The diversifying heuristic used for all ensembles here – random assignment of number of clusters for each clusterer – is probably sufficient together with random initialization. Drawing a sub-sample for k-means is slightly favourable with some consensus functions but the difference is not significant.

(3) Devising a good consensus function is a topic of continuing interest [1,6,15,20,22]. Many studies advocate CSPA and HGPA as a better alternative to single linkage on the consensus matrix **M**. All these studies assume that M taken to represent ***similarity***. They are generally right, as CSPA and HGPA are better than single linkage + similarity apart from ensembles designed by k-means (ensemble 21 is better than ensembles 13 and 9 on both criteria). However, it appears that the clear winner in the consensus function "competition" is using the ***consensus matrix as data***. Within the shaded blocks of rows there is one CSPA (ensemble 15) and one single linkage + similarity (ensemble 21) as the consensus functions. All the others use **M** as data. This confirms our previous results as well [13]. Based on these findings we recommend using **M** *as data* in ensemble designs 4, 8, 5 and 1.

5 Conclusions

In this paper we compare experimentally 24 ensemble design methods shown in Table 1. A collection of 24 data sets were used in order to evaluate the relative performance of the methods. The performance criteria were the Adjusted Rand index (AR) and the classification accuracy (CA).

We have limited the ensemble size to 10. The ensembles considered here are very small, compared to the ensembles studied elsewhere, e.g Greene et al. [12], who use sizes up to 3000. One positive side of this choice is the obvious gain in computational speed, which allowed us to run 100 experiments for each of the 24 data sets and each of the 24 ensemble methods. We expect that for large ensemble sizes the differences between the methods may be blurred because all ensembles will be expected to give reasonable performance for sizes beyond 1000. The negative side of the small ensemble size is that the results here are valid for small, and admittedly not very accurate ensembles. We cannot project the findings for the case of large ensembles which limits the impact of the results.

Because of the large number of experiments, we were able to carry out statistical comparisons using both confidence intervals for pairwise comparisons and Friedman's Two-Way ANOVA for group comparisons.

The consensus functions tested here included a new approach whereby the consensus matrix **M** is used as data (features) rather than as similarity. This appeared to be even more successful than we expected ourselves, although it does confirm our previous experiments of a much smaller scale. Interpreting **M** as data fared better than some standard and widely used consensus functions including CSPA and HGPA.

The fact that the number of clusters was "given" to the consensus function seems to be a serious limitation of this study. However, this has been a common practice, especially when consensus functions are being compared, because many consensus functions rely on this number being provided. Alternatively, the number of clusters can be estimated by using cluster validity indices at individual level and deriving a final "consensus" number. The single linkage combination method provides a means for estimating this number. Other options lie with using stability measures. In all these cases, the accuracy of this guess will make or break the ensemble. As our focus was on comparing consensus functions, we decided to place them all in the best starting position whereby the number of clusters is known in advance. Future research directions include expanding the experiment to examine various diversity measures in relationship to AR and CA, as well as checking the consistency of the results obtained here for other ensemble sizes.

Acknowledgements

This work was supported by research grant # 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 T. Windeatt and F. Roli, editors, *Proc. 4th International Workshop MCS'03*, LNCS 2709, pages 166–175, Guildford, UK, 2003.

[2] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[3] B. Boyanov and S. Hadjitodorov. Acoustic analysis of pathological voices and screening of laryngeal diseases. *IEEEEMB Magazine*, **16**:74–82, 1997.

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

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

[6] X.Z. Fern and C.E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In Proc. 21th International Conference on Machine Learning, ICML, Ban_, Canada, 2004.

[7] B. Fischer and J. M. Buhmann. Bagging for path-based clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**(11):1411–1415, 2003.

[8] A. Fred. Finding consistent clusters in data partitions. In F. Roli and J. Kittler, editors, *Proc. 2nd International Workshop MCS'01*, LNCS 2096, pages 309–318, Cambridge, UK, 2001. Springer-Verlag.

[9] A. N. L. Fred and A. K. Jain. Combining multiple clusterings using evidence accumulation. IEEE *Transactions on PAMI*, **27**(6):835–850, 2005.

[10] 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.

[11] J. Ghosh. Multiclassifier systems: Back to the future. In F. Roli and J. Kittler, editors, *Proc. 3d International Workshop MCS'02*, LNCS 2364, pages 1–15, Cagliari, Italy, 2002. Springer-Verlag.

[12] D. Greene, A. Tsymbal, N. Bolshakova, and P. Cunningham. Ensemble clustering in medical diagnostics. Technical Report TCD-CS-2004-12, Department of Computer Science, Trinity College, Dublin, Ireland, 2004.

[13] S. T. Hadjitodorov, L. I. Kuncheva, and L. P. Todorova. Moderate diversity for better cluster ensembles. *Information Fusion*, to appear.

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

[15] X. Hu and I. Yoo. Cluster ensemble and its applications in gene expression analysis. *In Proc. 2-nd Asia-Pacific Bioinformatics Conference (APB2004)*, Dunedin, New Zealand, 2004.

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

[17] 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.

[18] E. Pękalska. Dissimilarity Representations in Pattern Recognition. PhD thesis, *Delft University of Technology*, The Netherlands, 2005.

[19] B.D. Ripley. Pattern Recognition and Neural Networks. *University Press*, Cambridge, 1996.

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

[21] L. Todorova and A. Temelkov. Weaning from long-term mechanical ventilation: a nonpulmonary weaning index. *Journal of Clinical Monitoring and Computing*, **18**:275–281, 2004.

[22] A. Topchy, A. K. Jain, and W. Punch. Clustering ensembles: models of consensus and weak partitions, *IEEE Transactions on PAMI*, **27**(12):1866-1881, 2005

[23] F. Vilariño, L. I. Kuncheva, and P. I. Radeva. ROC curves in video analysis optimization in intestinal capsule endoscopy. *Pattern Recognition Letters*, 2005. to appear.