Selecting Diversifying Heuristics for Cluster Ensembles

Stefan T. Hadjitodorov[1] and Ludmila I. Kuncheva[2]

[1] CLBME, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria
sthadj@argo.bas.bg
[2] School of Computer Science, University of Wales, Bangor, UK
l.i.kuncheva@bangor.ac.uk.

Abstract. Cluster ensembles are deemed to be better than single clustering algorithms for discovering complex or noisy structures in data. Various heuristics for constructing such ensembles have been examined in the literature, e.g., random feature selection, weak clusterers, random projections, etc. Typically, one heuristic is picked at a time to construct the ensemble. To increase diversity of the ensemble, several heuristics may be applied together. However, not any combination may be beneficial. Here we apply a standard genetic algorithm (GA) to select from 7 standard heuristics for k-means cluster ensembles. The ensemble size is also encoded in the chromosome. In this way the data is forced to guide the selection of heuristics as well as the ensemble size. Eighteen moderate-size datasets were used: 4 artificial and 14 real. The results resonate with our previous findings in that high diversity is not necessarily a prerequisite for high accuracy of the ensemble. No particular combination of heuristics appeared to be consistently chosen across all datasets, which justifies the existing variety of cluster ensembles. Among the most often selected heuristics were random feature extraction, random feature selection and random number of clusters assigned for each ensemble member. Based on the experiments, we recommend that the current practice of using one or two heuristics for building k-means cluster ensembles should be revised in favour of using 3-5 heuristics.[1]

Keywords: Pattern recognition; multiple classifier systems; cluster ensembles; genetic algorithms; diversifying heuristics.

1 Introduction

Selecting a good clustering algorithm is more difficult than selecting a good classifier. The difficulty comes from the fact that in clustering there is no supervision, i.e., data have no labels against which to match the partition obtained through the clustering algorithm. Therefore, instead of running the risk of picking an unsuitable clustering algorithm, a cluster ensemble can be used [13]. The

[1] This work was supported by research grant # 15035 under the European Joint Project scheme, Royal Society, UK.

M. Haindl, J. Kittler, and F. Roli (Eds.): MCS 2007, LNCS 4472, pp. 200–209, 2007.

presumption is that even a basic off-the-shelf cluster ensemble will outperform a randomly chosen clustering algorithm. The question then becomes whether we can guide the selection of a cluster ensemble.

Here we are interested in *cluster ensembles*. Various heuristics have been proposed for building diverse cluster ensembles. Usually these heuristics are applied one at a time or at most two. The large majority of the publications on cluster ensembles are devoted to finding a combination method (called sometimes a consensus function), for example [1, 3, 5, 11, 15, 6, 14, 10], while few papers look into comparisons between different diversifying heuristics e.g., [8]. In this study we propose to evaluate combinations of such heuristics by a standard genetic algorithm. Our hypothesis is that better cluster ensembles could be created using more than one diversifying heuristics at the same time. The objective is to find out which diversifying heuristics and combinations thereof are being selected more frequently by a data-guided GA.

Our application is focused on a class of datasets whose common characteristics are: (1) small number of true classes (often overlapping), 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. In the experiments reported in Section 4 we have used, among others, six such benchmark data sets from the UCI Machine Learning Repository [2].

The rest of the paper is organized as follows. Section 2 lists the heuristics for building diverse cluster ensembles and explains the main ensemble algorithm. Section 3 describes briefly the genetic algorithm. The choice of data sets and the experimental set-up are detailed in Section 4, where we also present and discuss the results. Section 5 concludes the study.

2 Cluster Ensembles

We investigate the effect of various design heuristics on the ensemble accuracy. These heuristics are necessary in order to make sure that the individual clusterers produce different, yet sensible, partitions of the data.

2.1 Cluster Ensembles

Let $P_1, \ldots, P_L$ be a set of partitions of a data set $\mathbf{Z}$, each one obtained from applying a clustering algorithm, or a 'clusterer'. The aim is to find a resultant partition P^* which best represents the structure of $\mathbf{Z}$. We implemented the pairwise approach [4] because it has been a popular choice despite its comparatively large computational complexity. The generic version of the pairwise cluster ensemble algorithm is outlined below.

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.

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 $m_{ij}^{(k)} = 1$, if $\mathbf{z}_i$ and $\mathbf{z}_j$ are in the same cluster in partition k, and $m_{ij}^{(k)} = 0$ otherwise.
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. A typical choice for $\mathbf{M}$ is the average of the individual matrices $M^{(k)}$.

The consensus matrix $\mathbf{M}$ can be regarded as a similarity matrix between the points of $\mathbf{Z}$. Therefore, it can be used with any clustering algorithm which operates directly upon a similarity matrix. 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. Extensive experimentation have singled out hypergraph methods (HGPA, CSPA and MCLA [13]) and average linkage as the best consensus functions. In a previous study we found that better results were obtained if we used $\mathbf{M}$ as a new feature space and ran k-means on it [9].

The randomisation heuristics come into play in Step 2 where the individual partitions are formed.

Cluster validation presents a difficult problem with no trivial solution. Here we assume that this problem has been solved and the "true" number of clusters is available. This assumption, restrictive as it is, is not unusual for studies like ours. The focus of this paper is the relative merit of heuristics and combinations of heuristics compared to one another. We may well pre-set the best possible scenario where the number of clusters is given as this setup will not disadvantage any of the heuristics.

The most widely used indices to estimate similarity between partitions are Rand, Jaccard, adjusted Rand, correlation, mutual information and entropy. When the number of obtained clusters is the same as the number of known groups in the data, the apparent accuracy of the cluster ensemble (classification accuracy) has been used as the most intuitive measure. To calculate classification accuracy, each cluster is labeled with the class most represented within and the proportion of correctly labeled objects from the whole of Z is evaluated. This re-labeling of the clusters guarantees the best classification accuracy.

3 The Genetic Algorithm for Selecting Diversifying Heuristics

Genetic algorithms (GA) are a popular optimization technique [7]. They provide a form of guided random search whereby the solution is evolved within a "population" through subsequent iterations called generations. Each population consists of "chromosomes" which describe the individuals. In our case, an individual will be a cluster ensemble encoded as a 12-bit binary string. The first seven bits encode the heuristics as explained in the next section. A value of 1 means that the respective heuristic is chosen for the ensemble. Bits 8 to 12 encode

the ensemble size in the following way. These bits are assigned "weights" as [5, 10, 20, 50, 100]. The ensemble size, L, is the sum of the weights of the selected bits (values 1). For example, [0,0,1,0,1] means $L = 120$. The fitness function used to evaluate the merit of a chromosome is the classification accuracy of the respective ensemble (the accuracy of the resultant partition P^*). Interestingly, in this implementation, the same chromosome may get different fitness values if evaluated twice. This is because only the structure of the ensemble is determined within the chromosome. The fitness depends upon various random parameters according to the heuristics in the chromosome. In other words, slightly different phenotypes may correspond to the same genotype. In order to eliminate part of this randomness, we take as the fitness of a chromosome, the average of five evaluation runs.

We use the standard GA with choices as shown below

1. Pick the parameters of the GA:
   (a) Population size m (even).
   (b) Maximum number of generations T_{max}.
   (c) Mutation probability P_m.
2. Generate a random population of m chromosomes and calculate their fitness values.
3. For $i = 1 : T_{max}$,
   (a) Assuming that the whole population is the mating set, select $m/2$ couples of parents from the current population (repetitions are allowed).
   (b) Perform (one-point) crossover to generate m offspring chromosomes.
   (c) Mutate the offspring according to the mutation probability.
   (d) Calculate the fitness values of the mutated offspring.
   (e) Pool the offspring and the current population and select as the next population the m chromosomes with the highest fitness.
4. End i.

The limit number of generations, population size and mutation probability are parameters of this GA model. We assume that the whole population is allowed to reproduce, the crossover probability is set to 1.0, and since elitist selection is used the generation gap is not fixed. This drives the model closer to a random search with main emphasis being on exploration.

4 The Experiment

4.1 Data Sets

Figure 1 shows four artificial data sets: difficult-doughnut, easy-doughnut, four-gauss and two-spirals. The first three datasets were generated in 2-D (as plotted) and then 10 more dimensions of uniform random noise were appended to each data set. A total of 100 points were generated from each distribution.[2]

[2] Matlab code for generating these data sets is available at http://www.informatics.bangor.ac.uk/∼kuncheva/activities/patrec1.html

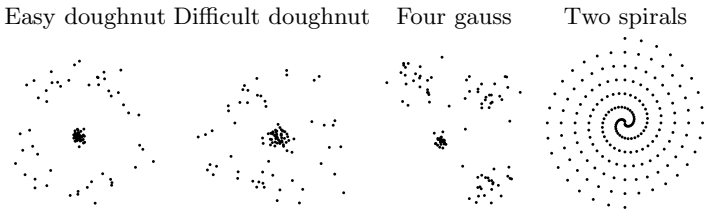


Fig. 1. The four artificial data sets used in this study. The first three data sets were generated with 10 additional noise features.

Three benchmark biological datasets were used: crabs [12], iris and soybean-small from UCI [2]. The parameters of all data sets are summarized in Table 1. The eleven medical data sets in this study come from two sources. The datasets breast, heart, liver, lymph, pima diabetes and thyroid are from UCI while the other five data sets are now available at .`http://www.informatics.bangor.ac.uk/.kuncheva/activities/patrec1.html`

These data sets are

contractions (98 objects, 9 features, 2 classses)
weaning (151 objects, 17 features, 2 classes)
respiratory (85 objects, 17 features, 2 classes)
laryngeal (213 objects, 16 features, 2 classes)
voice-3 (238objects, 10 features ,3 classes)

4.2 Experimental Protocol

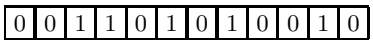
All real data sets except iris and soybean-small were standardized (all features were transformed to have mean 0 and standard deviation 1). The standardization was deemed necessary because the data contained mixed variables and variables measured in very different scales.

All ensembles used k-means, started from a random initialization, as the base clusterer. The following heuristics were encoded as the first 7 bits of the chromosome in the GA.

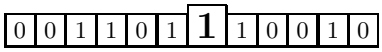
1. Different samples. We used subsampling of size randomly chosen between the number of clusters and the total number of objects.
2. Weak clustering algorithm. k-means is stopped after the second iteration.
3. Random projections (feature extraction). We form d random projections where d is the number of relevant principal components obtained from the correlation matrix of the data (eigenvalues greater than 1).
4. Feature selection. A non-empty random subset of the original feature set is picked. Each feature has a chance of 0.5 to be included in the set.
5. Label noise. Here we used 5% label noise.
6. Random number of clusters. If this heuristic is selected, the number of overproduced clusters, c, is picked from the range from 2 to 22.

7. Hybrid ensembles. This heuristic offers another possibility for incorporating diversity in a non-uniform way. The hybridization is not done over different clustering methods but consists in giving each clusterer in the ensemble the freedom to apply different heuristics. The example below illustrates this hybridization.

The seven heuristics are represented as the first 7 of the 12 bits of the chromosome while the last 5 bits encode the ensemble size. For example, an ensemble represented by chromosome



will consist of 55 (5+50) clusterers. Each of them will be built using k-means with random feature selection (heuristic 4) followed by random linear feature extractions (heuristic 3)[3] and a randomly chosen number of overproduced clusters between 2 and 22 (heuristic 6). Consider now the following chromosome, corresponding to a hybrid ensemble



In this case, each of the 55 clusterers will have a chance to select any combination of the three heuristics (3, 4 and 6) or none of them. This means that the hybridization opens up a second possibility for further "refined" selection of the already selected heuristics.

If none of the first 7 bits of the chromosome is switched to 1, only random initialization of k-means is applied. If none of the last 5 bits of the chromosome is switched to 1, a default value of $L = 5$ is assigned.

The GA parameters were chosen as follows: population size $m = 10$; maximum number of generations $T_{max} = 30$ and mutation probability $P_m = 0.15$.

4.3 Results

Table 1 displays the data characteristics, the end results from the GA and the corresponding accuracies. N denotes the number of objects in the data set, n is the number of features, c is the number of clusters, L is the ensemble size and *Acc* is the classification accuracy of the ensemble. Shown for each data set is the best chromosome in the last (30th) generation. The classification accuracy *Acc* is an average of 5 runs of the ensemble. In the last column we show the classification accuracy obtained by Greene et al. [8].

Our hypothesis is that the improved ensemble accuracy is owed to the selection of appropriate heuristics. Figure 2 shows the proportion of times each of the 7 heuristics has been selected. The large error bars give the means and the 95% confidence intervals of the respective proportions calculated within the last population of the GA. As there are 18 data sets, and each population contains 10

[3] The order in which we apply heuristics 3 and 4 is immaterial. We have chosen to apply 4 before 3 for computational convenience.

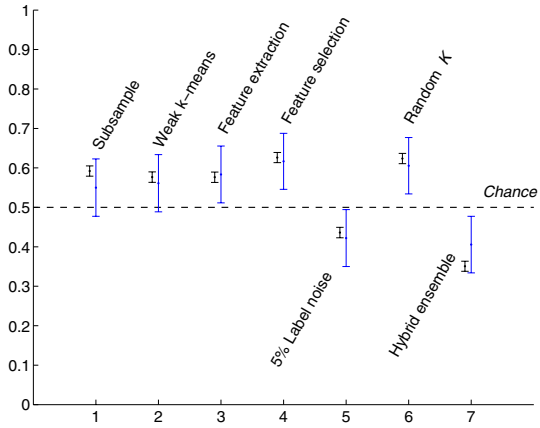


Fig. 2. Mean proportions of individual occurrences of the 7 heuristics with 95% confidence intervals for the mean. The large intervals are derived from the last population of the GA; the small intervals are derived from the whole run of the GA.

Table 1. Data characteristics and the end results from the GA

				Heuristics									
Dataset	N	n	c	1	2	3	4	5	6	7	L	Acc	[8]
difficult doughnut	100	12	2	0	0	0	1	0	1	0	65	0.982	
easy doughnut	100	12	2	0	0	0	1	0	1	1	60	1.000	
four gauss	100	12	4	1	1	1	1	1	0	0	130	0.982	
spirals-2	194	2	2	0	1	1	0	0	1	0	120	0.551	(1.000)
crabs	200	7	2	1	1	1	1	1	0	0	165	0.625	
iris	150	4	3	1	0	1	1	1	1	0	165	0.933	(0.893)
soybean-small	47	35	4	1	0	0	1	1	0	0	170	0.915	
breast	277	9	2	0	1	1	0	1	1	1	10	0.718	(0.762)
heart	270	13	2	1	1	0	1	1	0	0	150	0.829	(0.600)
liver	345	6	2	1	1	1	0	1	0	1	120	0.602	(0.585)
lymph	148	18	4	1	0	1	0	0	1	1	110	0.488	(0.615)
pima diabetes	768	8	2	1	1	0	1	0	1	0	185	0.698	(0.675)
thyroid	215	5	3	1	0	1	0	1	1	1	170	0.889	(0.793)
contractions	98	27	2	1	0	1	1	0	0	0	65	0.845	
intubation	302	17	2	0	0	1	0	0	0	1	170	0.772	
laryngeal	213	16	2	0	0	1	1	0	0	1	55	0.822	
respiratory	85	17	2	1	1	0	1	0	1	0	155	0.948	
voice-3	238	10	3	0	1	1	1	0	0	0	5	0.771	

chromosomes, the proportions are calculated from 180 chromosomes. For reference, we plot the probability of being selected by chance (0.5) with a dashed line. According to the confidence intervals, heuristics 3, 4 and 6 are selected more often than chance whereas label noise (heuristic 5) and hybrid ensembles (heuristic 7) are suppressed. The short error bars show the mean and the 95% confidence

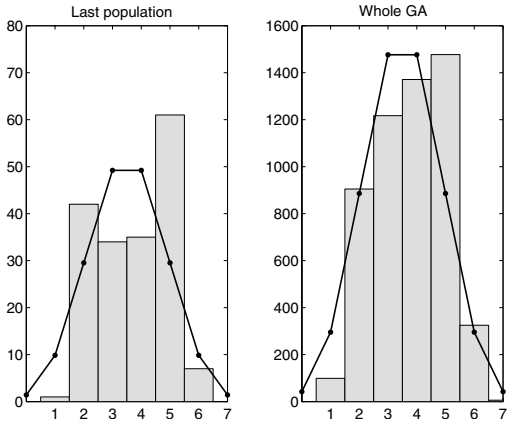


Fig. 3. Histograms of the number of selected heuristics. The overlaid polygon is the theoretical binomial distribution for $n = 7$ and $p = 0.5$.

intervals of the proportions calculated from the whole run of the GA (30 generations). Thus each proportion is evaluated on $30 \times 10 \times 18 = 5400$ chromosomes. The heuristics which are picked more often than chance are Subsample, Weak k-means, Feature extraction, Feature selection and Random c.

Shown in Figure 3 are the histograms of the number of selected heuristics for an ensemble. The left plot is obtained from the last populations for the 18 data sets, and the right plot is obtained from the whole run of the GA. If each heuristic was selected independently and completely by chance, the number of selected heuristics would follow a binomial distribution with parameters $n = 7$ and $p = 0.5$. The polygon for the binomial distribution is overlaid in the two plots. To check whether the obtained distribution differs from binomial, we carried out a χ^2 test. With significance $p < 10^{-9}$, both obtained distributions are different from binomial distribution.

Table 2. Combinations of heuristics with largest frequency of occurrence for a specified number of selected heuristics

	Heuristics	Frequency	Proportion	95% CI
# selected	1 2 3 4 5 6 7			
7	1 1 1 1 1 1 1	6	0.011	0.0020–0.0200
6	1 1 1 1 1 1 0	143	0.0265	0.0222–0.0308
5	1 1 1 1 0 1 0	302	0.0559	0.0498–0.0620
4	1 1 1 0 0 1 0	262	0.0485	0.0428–0.0542
3	0 0 1 1 0 0 1	131	0.0243	0.0202–0.0284
2	0 0 0 1 0 1 0	194	0.0359	0.0310–0.0409
1	0 0 0 1 0 0 0	56	0.0104	0.0077–0.0131

The most notable difference from the binomial probability is observed for 5 heuristics selected together. Five heuristics have been selected in 33.89% of the ensembles in the last populations of the GA and in 27.35% of the ensembles within the whole run. The next largest difference is for 2 selected heuristics in the last population (23.33%).

Table 2 shows which combinations of heuristics have been encountered most frequently when different number of heuristics have been selected. For example, for five selected heuristics, heuristics 1, 2, 3, 4, and 6 appeared in 302 out of 1477 chromosomes. The last column of the table shows the 95% confidence interval for the mean (averaged on 5400 cases). Knowing that the chance for selecting a particular combination is $\frac{1}{2^7} = 0.0078$, the chances of selecting the combinations shown in the table are significantly larger than chance ($\alpha = 0.05$) for 2 to 6 selected heuristics. The probability of selecting all 7 heuristics is significantly below chance while the probability for selecting only heuristic 4 is not significantly higher than chance.

No combination of heuristics appeared together consistently. A glance at the correlation matrix using the whole run of the GA reveals that correlations between pairs of heuristics are weak, varying between -0.2598 (feature extraction and feature selection) and 0.2456 (weak k-means and label noise). Therefore, we can think of the diversifying heuristics as fairly independent.

5 Conclusions

We restricted our study to datasets which one may acquire from pilot studies in biomedical domain, e.g., pilot clinical trials. Such data sets have small number of classes (we assume that they correspond to clusters), moderate number of observations (up to few hundred) and moderate number of features (typically 5 to 30). Our collection for this study consisted of 18 such data sets, among which artificial, real, benchmark and new medical data. Using a GA to select combination of heuristics as well as ensemble size we found that: (1) More than 1 heuristic is better. The collection of heuristics being chosen most often by the GA was {Subsample, Weak k-means, Feature extraction, Feature selection and Random c} and (2) Too many is not necessarily good. Ensembles with more than 5 heuristics appeared to be too random to be useful.

We also observed that ensemble sizes of 100+ fared better than smaller ensemble for the type of problems in this study. However, it seems that the accuracy for ensemble sizes beyond 100 starts to level off and ensembles of 2000 clusterers may only offer marginal improvement at the expense of a large increase of the computational cost.

Our experimental results indicated low dependency between heuristics. This was partly expected because the heuristics come from different ways of handling the data and setting the clustering procedures. The independence shows that each heuristic has a specific niche and should not be lightly ignored. This study was focused on selection of heuristics assuming that the "correct" number of clusters is known. Evaluating the number of clusters is a challenging problem of its own, worthy of a separate study.

References

1. H. Ayad, O. Basir, and M. Kamel. A probabilistic model using information theoretic measures for cluster ensembles. In *Proc. 5th International Workshop on Multiple Classifier Systems, MCS04*, pages 144–153, Cagliari, Italy, 2004.
2. C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.
3. X. Z. Fern and C. E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proc. 21th International Conference on Machine Learning, ICML*, Banff, Canada, 2004.
4. A. Fred. Finding consistent clusters in data partitions. In F. Roli and J. Kittler, editors, *Proc. 2nd International Workshop on Multiple Classifier Systems, MCS'01*, volume 2096 of *Lecture Notes in Computer Science*, pages 309–318, Cambridge, UK, 2001. Springer-Verlag.
5. A. N. L. Fred and A. K. Jain. Combining multiple clusterungs using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005.
6. J. Ghosh. Multiclassifier systems: Back to the future. In F. Roli and J. Kittler, editors, *Proc. 3d International Workshop on Multiple Classifier Systems, MCS'02*, volume 2364 of *Lecture Notes in Computer Science*, pages 1–15, Cagliari, Italy, 2002. Springer-Verlag.
7. D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, NY, 1989.
8. 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.
9. L. I. Kuncheva, S. T. Hadjitodorov, and L. P. Todorova. Experimental comparison of cluster ensemble methods. In *Proc. FUSION*, Florence, Italy, 2006.
10. B. Minaei, A. Topchy, and W. Punch. Ensembles of partitions via data resampling. In *Proceedings of the International Conference on Information Technology: Coding and Computing, ITCC04*, Las Vegas, 2004.
11. 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.
12. B. D. Ripley. *Pattern Recognition and Neural Networks.* University Press, Cambridge, 1996.
13. A. Strehl and J. Ghosh. Cluster ensembles - A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–618, 2002.
14. A. Topchy, B. Minaei, A. K. Jain, and W. Punch. Adaptive clustering ensembles. In *Proceedings of ICPR, 2004, Cambridge, UK*, 2004.
15. A. Weingessel, E. Dimitriadou, and K. Hornik. An ensemble method for clustering, 2003. Working paper, `http://www.ci.tuwien.ac.at/Conferences/DSC-2003/`.