Cascade evaluation of clustering algorithms

Laurent Candillier[1,2], Isabelle Tellier[1], Fabien Torre[1], Olivier Bousquet[2]

[1] GRAppA - Charles de Gaulle University - Lille 3
candillier@grappa.univ-lille3.fr
[2] Pertinence - 32 rue des Jeûneurs -75002 Paris
olivier.bousquet@pertinence.com

Abstract. This paper is about the evaluation of the results of clustering algorithms, and the comparison of such algorithms. We propose a new method based on the enrichment of a set of independent labeled datasets by the results of clustering, and the use of a supervised method to evaluate the interest of adding such new information to the datasets. We thus adapt the *cascade generalization* [1] paradigm in the case where we combine an unsupervised and a supervised learner. We also consider the case where independent supervised learnings are performed on the different groups of data objects created by the clustering [2].
We then conduct experiments using different supervised algorithms to compare various clustering algorithms. And we thus show that our proposed method exhibits a coherent behavior, pointing out, for example, that the algorithms based on the use of *complex* probabilistic models outperform algorithms based on the use of *simpler* models.

1 Introduction

In both supervised and unsupervised learning, the evaluation of the results of a given method, as well as the comparison of various methods, is an important issue. But if cross-validation is a widely accepted method to evaluate supervised learning algorithms, the problem of evaluating unsupervised learning algorithms remains an open issue. The main problem is that the evaluation of clustering results is subjective by nature. Indeed, there are often many different and relevant ways of grouping together some given data objects.

In practice, four main techniques are used to measure the quality of clustering algorithms. But each of these techniques has its own limitations.

1. Use artificial datasets where the desired grouping is known. But the given algorithms are thus evaluated only on the corresponding generated distribution, and results on artificial data can not be generalized to real data.
2. Use labeled datasets and check if the clustering algorithm retrieves the initial classes. But the classes of a supervised problem are not necessarily the classes that have to be found by a clustering algorithm because other groupings can also be meaningful.
3. Work with an expert who evaluates the meaning of the clustering in a particular field. However, if it is possible for an expert to tell if a given clustering

has some meaning, it is much harder to quantify its interest, or to tell if a given result is better than another one. Besides, the relevance of the method can not be generalized to various types of data.
4. Or use some internal criterion, like the intra-cluster inertia and/or the inter-clusters separation. But such pre-defined criteria are also subjective by nature because they use some pre-defined notion of what is a good clustering. For example, inter-clusters separation is not always the best criterion to use : clusters that overlap may sometimes be more relevant.

The main risk in evaluating a clustering method is to consider it as a goal in itself. In fact, what we want to evaluate is how well a given clustering method is able to capture new meaningful and useful information, that is some new knowledge interesting to use for some purpose. We also expect the method to be able to capture such interesting information on various types of problems.

So the main idea of our approach is to consider the clustering as a preprocessing step for another task that we are able to evaluate : supervised learning for instance. Thus the new evaluation method we propose in this paper consists in comparing the results of a supervised algorithm when it is (or not) provided with information coming from a clustering algorithm. If the results of the supervised learning algorithm are improved when some extra-knowledge coming from a clustering process is added, then we conjecture that it means that the clustering process managed to capture some new meaningful and useful information.

This method thus allows us to objectively evaluate the interest of the information captured by a given clustering algorithm. Moreover, the decrease of the error rate of the supervised algorithm when it is helped with the information coming from the clustering algorithm also allows us to quantify this interest. Our evaluation method thus depends on the chosen task, but it allows us to evaluate the contribution of the clustering in the achievement of this objective and real task. Besides, such a bias is less important than when a direct mapping between the clusters and the classes is evaluated.

So our method lies in the framework of classifier combination, in our case the combination of an unsupervised and a supervised method. Many ways of combining classifiers by votes can be found in [3], the two mostly used methods being *bagging* [4] and *boosting* [5]. Some theoretical generalization of these techniques have also been studied, leading to *arcing classifiers* [6], *ensemble methods* [7] and *leveraging methods* [8].

We focus here on techniques that use different learners in a sequential way. In such methods, the output of a learner is an enrichment of the example description, that is then used by the next learner. In that field, *stacked generalization* [9] is a very general framework in which different treatments are stacked : each treatment modifies the example description, and this new dataset is used by the next level. *Cascade generalization* [1] is a special case of stacked generalization. At each level, a classifier is applied on each example $\boldsymbol{x}$ providing probabilities $p(c|\boldsymbol{x})$ that $\boldsymbol{x}$ belongs to class c. These probabilities are then added to the example description and used by the next level classifier. Cascade generalization allows to combine several classifiers but in practice, only two learners are used.

Finally, we also consider the case where we combine an unsupervised and a supervised learner as is done in [2]. In that case, many clusterings are run with different input parameters, leading to different partitions of the set of data objects. For each partition, many independent supervised learnings are executed on the different created groups of data objects and the global error rate is computed. Finally, the partition that leads to the lower error rate is kept.

Based on this principle of sequentially combining an unsupervised and a supervised learner, and then computing the decrease of the error rate of the supervised learner when it is helped by the unsupervised learner, the new evaluation method of clustering algorithms we propose is called *cascade evaluation*. We first describe this new method in section 2. Then section 3 presents some experiments conducted with this new method. Finally, section 4 concludes the paper and suggests topics for future research.

2 Cascade evaluation methodology

Being given an initial dataset with classes information, the general steps of our proposed methodology are as follows :

1. learning 1 :
   - perform a supervised learning on the initial dataset;
2. learning 2 :
   - perform a clustering on the dataset without using the classes information;
   - enrich the dataset from the clustering results;
   - and perform a supervised learning on the enriched dataset;
3. compare the results of both learned classifiers.

As we already stated, we consider two different ways of enriching datasets from the results of a given clustering. The first one consists in creating new attributes that represent the information captured by the clustering process, and then adding these new attributes to the initial dataset before running the supervised learning on the enriched dataset. The second way is to consider the new sub-datasets created by the clustering and to run many supervised learnings independently on each sub-dataset.

Concerning the new attributes created from the clustering results in the case of the first combination method, different types of information can be added.

1. As many clustering algorithms provide as output a partition of the initial dataset, we can use the membership of the data objects to the clusters to create new attributes. This information would be represented by a new categorical attribute, each data object being associated with an identifier of the cluster it belongs to.
2. We can also associate to each data object a set of attributes that represent the center of the cluster it belongs to. We would thus double the number of attributes in the dataset.

3. Recently, many *subspace clustering* algorithms [10] emerged that are able to associate to each dimension of each cluster a weight specifying its relevance in determining the membership of the data objects to the cluster. So in such cases, we could add to each data object one new continuous attribute per initial dimension corresponding to the weight, on that dimension, of the cluster it belongs to. Such new attribute would thus allow to differentiate data objects for which a given dimension is relevant from those for which it is not relevant (according to the subspace clustering results).

Besides, as most clustering algorithms need some parameters to tune, we can run these algorithms many times with different input parameter values and enrich the dataset for each clustering results. For example, many clustering methods need as input the number of clusters to be found. In such cases, we could run them many times, varying this parameter from 2 to 10 for example. The supervised algorithm used afterwards would then be able to choose which attribute(s) to use among them.

In the case of the second combination method proposed, we first generate many partitions with different input parameters. We then compute the cross-validation error of independent supervised learners executed on the different groups of data objects created by the clustering. And finally, we select the partition that led to the lowest error rate.

To evaluate the improvement in the results of the supervised learning algorithm with or without the new information coming from the clustering process, we test both methods on various independent datasets. On each dataset, we perform five 2-fold cross-validations, as proposed in [11]. For each 2-fold cross-validation, we compute the balanced error rates of both methods. And we then use four measures to compare them :

- *nb wins*: the number of wins of each method;
- *sign wins*: the number of significant wins, using the *5×2cv F-test* [12] to check if the results are significantly different;
- *wilcoxon*: the wilcoxon signed rank test, that indicates if a method is significantly better than another one on a set of independent problems (if its value is higher than 1.96);
- and *av perf*: the mean balanced error rate.

3 Experiments

We present in this section the results of the comparisons of various clustering algorithms :

- Rand, an algorithm that generates random partitions, being given the number of expected clusters (used as a reference);
- K-means, the well-known full-space clustering algorithm based on the evolution of K centroids that represent the K clusters to be found;

- LAC [13], a subspace clustering algorithm based on K-means that associates with each centroid a vector of weights on each dimension, inversely proportional to the dispersion of the members of the clusters on the dimension;
- SSC [14], that is based on the use of a probabilistic model and the EM algorithm [15] under the assumption that the data follow independent gaussian distributions on each dimension;
- and SuSE [16], an adaptation of SSC that performs *hard feature selection* during the learning process, by selecting for each cluster a subset of the dimensions on which the standard deviation is minimized.

So the algorithms compared here use different models with different complexity levels. K-means uses only one centroid to represent a cluster. LAC adds to each centroid a vector of weights on each dimension. SSC defines a membership probability of each data object to each cluster, in addition to use a gaussian model. And SuSE also considers a subset of relevant dimensions associated to each cluster. All these algorithms need as input parameter the number K of clusters to be found. So as we discussed earlier, we will run them many times with K varying from 2 to 10.

In order to check if the results depend on the supervised algorithm used, we conduct these experiments with various supervised learning algorithms :

- C4.5 [17], the well-known supervised method based on the iterative construction of a decision tree;
- C5 [18] boosted 10 times, that uses the boosting of decision trees, in order to observe if the information added by clustering algorithms also help supervised methods that already combine many classifiers;
- DLG [19], a supervised method that uses *least general generalizations* instead of decision trees, so that many attributes are considered at a time to construct decision surfaces;
- and multi-class Support Vector Machines (SVM) [20], that construct large margin classifiers, in order to check if the information added by clustering algorithms also help supervised methods that use linear combinations of the initial features.

Finally, the datasets used are those of the UCI Machine Learning Repository [21] that contain only numerical attributes.

Table 1 presents the balanced error rates of C4.5 run on the initial dataset, and then run on datasets enriched by the results of the corresponding clustering algorithms. Each measure corresponds to an average over five 2-fold cross-validations. At each time, all the methods are run on the same training set and evaluated on the same test set.

From this table, we can observe that most of the time, the results of C4.5 are improved when some information coming from *real* clustering algorithms are added, whereas adding information from a random clustering degrades the results. Besides, we can note that the results of SSC and SuSE are often better than those of K-means and LAC. Then table 2 presents a summary of the comparison between C4.5 and C4.5 enriched by the clustering algorithms.

	C4.5	C4.5	C4.5	C4.5	C4.5	C4.5
	alone	+ Rand	+ K-means	+ LAC	+ SSC	+ SuSE
ecoli	48.5	48.3	42.8	**40.3**	42	43.1
glass	**32.6**	40.8	35.7	37	40.4	34.9
image	4.8	6	4.8	**4.6**	**4.6**	**4.6**
iono	14.1	15.8	14.2	13.1	**9.8**	11.2
iris	7.3	7.9	6.7	**3.7**	5.1	4.7
pima	31	35	32.1	32.1	30.8	**30**
sonar	31	35.2	30	28.8	28.8	**27.2**
vowel	29.5	38.5	25	26.4	24.1	**22.2**
wdbc	5.9	6.8	4.6	3.9	5.1	**3.1**
wine	8.7	8.8	10.4	9.6	**2.7**	3.6

Table 1. Balanced error rates (in %) of C4.5 enriched by clustering algorithms. The bold values correspond to the minimum error rates obtained on each dataset.

	C4.5	C4.5	C4.5	C4.5	C4.5	C4.5
	alone	+ Rand	+ K-means	+ LAC	+ SSC	+ SuSE
nb wins	-	1/9	5/4	7/3	**9/1**	**9/1**
sign wins	-	0/1	0/0	1/0	2/0	**3/0**
wilcoxon	-	-2.67	-0.05	1.31	1.83	**2.56**
av perf	21.3	24.3	20.6	20	19.3	**18.5**

Table 2. Comparison of C4.5 alone with C4.5 enriched by clustering algorithms.

SuSE is the only clustering algorithm that significantly helps C4.5 improve its results, according to the wilcoxon signed rank test. It is significantly better on 3 datasets according to the *5×2cv F-test*. But as SuSE, SSC improves the results of C4.5 nine times over ten, contrary to K-means and LAC. All algorithms improve the results of C4.5 on average, except the random clustering. And when C4.5 is combined with clustering algorithms based on more complex models, then the error rate is lower and the improvements are more significant than when it is combined with clustering algorithms based on simpler models.

Such experiments were also conducted using different supervised algorithms, namely C5 boosted, DLG and SVM, and using the second method for combining unsupervised and supervised algorithms. It is then very interesting to note that, in spite of the use of different supervised and combination methods, the clustering algorithms that best help supervised learners to minimize the cross-validation error rate on the different datasets remain mostly the same. In particular, SSC and SuSE still outperform K-means and LAC in many cases. Moreover, the order in which the clustering methods are ranked remains the same no matter which supervised and combination methods are used.

Finally, as a comparison, we computed the *F-measure* and the *Entropy* between the clusters obtained by the various clustering methods and the initial classes of the various problems in order to measure the mapping between them. We thus first observed that the two measures do not agree on which clustering method leads to the best mapping between the clusters and the classes on each dataset. Then we noted that there is no direct relation between the methods that optimize these values and the methods that better help the supervised learners to improve their results. Besides, such measures do not provide objective information about the interest of the clustering methods, contrary to our proposed evaluation method that shows if the results are significantly better with the help of the given clustering methods.

4 Conclusion

We have presented in this paper a new objective and quantitative evaluation method of clustering algorithms that consists in comparing the results of a supervised algorithm when it is (or not) provided with information coming from a clustering algorithm. We have considered different supervised algorithms to be used in our evaluation method. We have also considered two different ways of combining unsupervised and supervised learning algorithms.

The experiments pointed out that the order in which the clustering methods are ranked remains the same no matter which supervised algorithm and which combination method are used. So it shows the robustness of our proposed evaluation method. The experiments also pointed out that clustering methods based on the use of more complex models outperform methods based on the use of simpler models. This result is not surprising, but rather exhibits coherent results of our new evaluation method.

Although it was not the aim of our investigations, we have also shown that the results of supervised learning algorithms are improved when they use some extra-knowledge coming from non random clustering algorithms. We conjecture supervised learners can benefit from the information added by clustering methods because these new information are of very different nature. In particular, clustering algorithms can help supervised learners to specialize their treatments according to different specific areas in the input space. They can also help supervised learners fit more complex decision surfaces. It thus seems interesting to continue our investigations in the more general framework of classifier combination when one learner is unsupervised.

Our experiments seem to show that using the clustering to partition the object space, and then executing independent supervised learnings on each created group of data objects gives better results than enriching the datasets with new attributes and then executing a supervised learning on the enriched dataset, since the improvements are more important when the second method for combining unsupervised and supervised algorithms is used. But this may be a consequence of the method we have used to create new attributes, that significantly increases the size of the dataset. It would thus be interesting to examine this point in detail in future research.

Finally, in future works, it would also be interesting to find other tasks as objective as supervised learning, and for which clustering would be an interesting pre-processing, in order to conduct other experiments with our proposed evaluation method in such another framework. One possible way would be for example to compute the reduction in the execution time of various requests on OLAP databases that use (or not) a clustering algorithm to create their index.

References

1. Gama, J., Brazdil, P.: Cascade generalization. Machine Learning **41** (2000) 315–343
2. Apte, C.V., Natarajan, R., Pednault, E.P.D., Tipu, F.A.: A probabilistic estimaton framework for predictive model analytics. IBM Systems Journal **41** (2002)
3. Ali, K.M., Pazzani, M.J.: Error reduction through learning multiple descriptions. Machine Learning **24** (1996) 173–202
4. Breiman, L.: Bagging predictors. Machine Learning **24** (1996) 123–140
5. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Int. Conf. on Machine Learning. (1996) 148–156
6. Breiman, L.: Bias, variance, and arcing classifiers (1996) Technical Report 460, Statistics Department, University of California.
7. Dietterich, T.G.: Ensemble methods in machine learning. In Kittler, J., Roli, F., eds.: 1st Int. Workshop on Multiple Classifier Systems. Volume 1857 of LNCS., Springer-Verlag (2000) 1–15
8. Meir, R., Rätsch, G.: An introduction to boosting and leveraging. In Mendelson, S., Smola, A., eds.: Advanced Lectures on Machine Learning. Number 2600 in LNAI. Springer-Verlag (2003) 119–184
9. Wolpert, D.H.: Stacked generalization. Neural Networks **5** (1992) 241–259
10. Parsons, L., Haque, E., Liu, H.: Evaluating subspace clustering algorithms. In: Workshop on Clustering High Dimensional Data and its Applications, SIAM Int. Conf. on Data Mining. (2004) 48–56
11. Dietterich, T.G.: Approximate statistical test for comparing supervised classification learning algorithms. Neural Computation **10** (1998) 1895–1923
12. Alpaydin, E.: Combined 5x2cv F-test for comparing supervised classification learning algorithms. Neural Computation **11** (1999) 1885–1892
13. Domeniconi, C., Papadopoulos, D., Gunopolos, D., Ma, S.: Subspace clustering of high dimensional data. In: SIAM Int. Conf. on Data Mining. (2004)
14. Candillier, L., Tellier, I., Torre, F., Bousquet, O.: SSC : Statistical Subspace Clustering. In Perner, P., ed.: Machine Learning and Data Mining in Pattern Recognition (MLDM). LNCS, Leipzig, Germany, Springer Verlag (2005) 100–109
15. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B **39** (1977) 1–38
16. Candillier, L., Tellier, I., Torre, F., Bousquet, O.: SuSE : Subspace Selection embedded in an EM algorithm. In Miclet, L., ed.: Actes de la huitième Conférence d'Apprentissage (CAp). (2006)
17. Quinlan, J.R.: C4.5: Programs for Machine Learning. KAUFM (1993)
18. Quinlan, R.: Data mining tools see5 and c5.0 (2004)
19. Webb, G.I., Agar, J.W.M.: Inducing diagnostic rules for glomerular disease with the DLG machine learning algorithm. Artificial Intelligence in Medicine **4** (1992) 419–430
20. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research (JMLR) **6** (2005) 1453–1484
21. Blake, C., Merz, C.: UCI repository of machine learning databases [http://www.ics.uci.edu/∼mlearn/MLRepository.html] (1998)