Adaptive Clustering Ensembles[*]

Alexander Topchy, Behrouz Minaei-Bidgoli, Anil K. Jain, and William F. Punch
Department of Computer Science and Engineering
Michigan State University, E. Lansing, MI 48824, USA
{topchyal, minaeibi, jain ,punch}@cse.msu.edu

Abstract

Clustering ensembles combine multiple partitions of the given data into a single clustering solution of better quality. Inspired by the success of supervised boosting algorithms, we devise an adaptive scheme for integration of multiple non-independent clusterings. Individual partitions in the ensemble are sequentially generated by clustering specially selected subsamples of the given data set. The sampling probability for each data point dynamically depends on the consistency of its previous assignments in the ensemble. New subsamples are drawn to increasingly focus on the problematic regions of the input feature space. A measure of a data point's clustering consistency is defined to guide this adaptation. An empirical study compares the performance of adaptive and regular clustering ensembles using different consensus functions on a number of data sets. Experimental results demonstrate improved accuracy for some clustering structures.

1. Introduction

Exploratory data analysis and, in particularly, data clustering can significantly benefit from combining multiple data partitions. Clustering ensembles can offer better solutions in terms of robustness, novelty and stability [1, 2, 3]. Moreover, their parallelization capabilities can be naturally used in distributed data mining.

Combination of clusterings is a more challenging task than combination of supervised classifications. In the absence of labeled training data, we face a difficult correspondence problem between cluster labels in different partitions of an ensemble. Recent studies [4] have demonstrated that consensus clustering can be found using graph-based, statistical or information-theoretic methods without explicitly solving the label correspondence problem. Other empirical consensus functions were also considered in [5, 6, 7]. However, the problem of consensus clustering is known to be NP complete [8].

Beside the consensus function, clustering ensembles need a partition generation procedure. Several methods are known to create partitions for clustering ensembles. For example, one can use: (i) different regular clustering algorithms [2], (ii) different initializations, parameter values or built-in randomness of a specific clustering algorithm [1], (iii) weak clustering algorithms [3], (iv) data resampling [5, 6, 9]. All these methods generate ensemble partitions independently, in a sense that the probability to obtain the ensemble consisting of H partitions $\{\pi_1, \pi_2,\ldots,\pi_H\}$ of the given data D can be factorized:

$$p(\{\pi_1,\pi_2,...,\pi_H\} \mid D) = \prod_{t=1}^{H} p(\pi_t \mid D)\,. \qquad (1)$$

Hence, the increased efficacy of an ensemble is mostly attributed to the number of identically distributed and independent partitions, assuming that a partition of data is treated as a random variable π. Even when the clusterings are generated sequentially, it is traditionally done without considering previously produced clusterings:

$$p(\pi_t \mid \pi_{t-1},\pi_{t-2},...,\pi_1; D) = p(\pi_t \mid D)\,. \qquad (2)$$

However, similar to the ensembles of supervised classifiers using boosting algorithms [10], a more accurate consensus clustering can be obtained if contributing partitions take into account the solutions found so far. Unfortunately, it is not possible to mechanically apply the decision fusion algorithms from supervised (classification) to unsupervised (clustering) domain. New objective functions for guiding partition generation and the subsequent decision integration process are necessary.

We propose an adaptive approach to partition generation that makes use of clustering history. In clustering, ground truth in the form of class labels is not available. Therefore, we need an alternative measure of performance for an ensemble of partitions. We determine clustering consistency for data points by evaluating a history of cluster assignments for each data point within the generated sequence of partitions. Clustering consistency serves for adapting the data sampling to the current state of an ensemble during partition generation. The goal of adaptation is to improve confidence in cluster assignments by concentrating sampling distribution on problematic regions of the feature space. In other words, by focusing attention on the data points with the least consistent clustering assignments, one can better approximate (indirectly) the inter-cluster boundaries. To achieve this goal, we address the problems related to estimation of clustering consistency (Section 2) and of finding a consensus

[*] This research was supported in part by ONR grant # N00014-01-1-0266 and by National Science Foundation ITR 0085921.

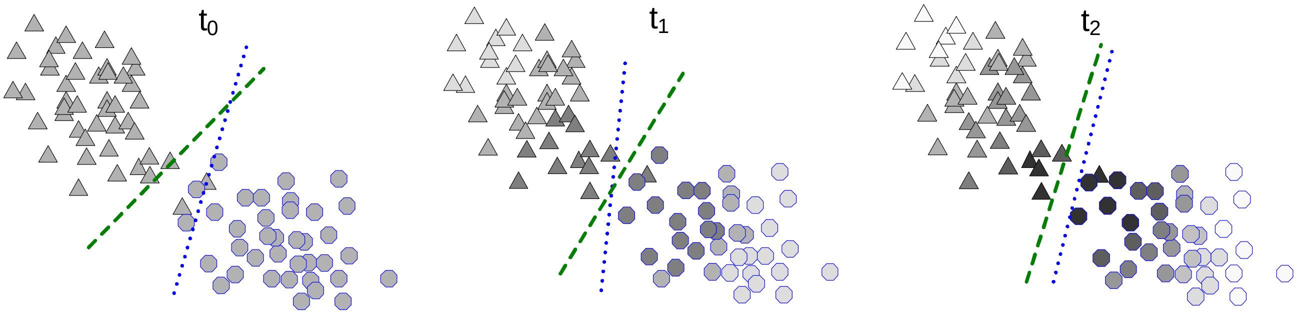

Figure 1. Two possible decision boundaries for a 2-cluster data set. Sampling probabilities of data points are indicated by gray level intensity at different iterations ($t_0 < t_1 < t_2$) of the adaptive sampling. True components in the 2-class mixture are shown as circles and triangles.

clustering. Finally, we evaluate the performance of adaptive clustering ensembles (Section 3) on a number of real-world and artificial data sets in comparison with more conventional clustering ensembles of bootstrap partitions [5,6,9].

2. Adaptive sampling and clustering

While there are many ways to construct diverse data partitions for an ensemble, not all of them easily generalize to adaptive clustering. Our approach extends the studies of ensembles whose partitions are generated via data resampling [5, 6]. Though, intuitively, clustering ensembles generated by other methods can be also boosted. It was shown [9] that small subsamples generally suffice to represent the structure of the entire data set in the framework of clustering ensembles. Subsamples of small size can reduce computational cost for many exploratory data mining tasks with distributed sources of data [11].

We begin with a formalization of clustering combination problem. Let D be a data set of N points that is available either as $N\times d$ pattern matrix in d-dimensional space or $N\times N$ dissimilarity matrix. Suppose that $X = \{X_1,\ldots,X_H\}$ is a set of H subsamples of size N drawn with replacement from the given data D. A chosen clustering algorithm is run on each of the samples in X that results in H partitions $\Pi=\{\pi_1, \pi_2,\ldots, \pi_H\}$. Each component partition in Π is a set of non-overlapping and exhaustive clusters with $\pi_i = \{C_1^i, C_2^i,\ldots,C_{K(i)}^i\}$, $X_i = C_1^i \cup\ldots\cup C_{K(i)}^i$, $\forall\pi_i$ and $K(i)$ is the number of clusters in the i-th partition.

The problem of combining partitions is to find a new partition $\sigma=\{C_1,\ldots,C_M\}$ of the entire data set D given the partitions in Π, such that the data points in a cluster of σ are more similar to each other than to points in different clusters of σ. We assume that the number of clusters M in the consensus clustering is predefined and can be different from the number of clusters k in the ensemble partitions. In order to find this target partition σ, one needs a consensus function utilizing information from the partitions $\{\pi_i\}$. Several known consensus functions [1, 2, 3] can be employed to map a given set of partitions $\Pi=\{\pi_1, \pi_2,\ldots, \pi_H\}$ to a target partition σ in our study.

The adaptive partition generation mechanism is aimed at reducing the variance of inter-class decision boundaries. Unlike the regular bootstrap method that draws subsamples uniformly from a given data set, adaptive sampling favors points from regions close to the decision boundaries. At the same time, the points located far from the boundary regions will be sampled less frequently. It is instructive to consider a simple example that shows the difference between ensembles of bootstrap partitions with and without the weighted sampling. Figure 1 shows how different decision boundaries can separate two natural classes depending on the sampling probabilities. Here we assume that the k-means clustering algorithm is applied to the subsamples. Initially, all the data points have the same weight, namely, the sampling probability $p_i = \frac{1}{N}$, $i\in[1,\ldots,N]$. Clearly, the main contribution to the clustering error is due to the sampling variation that causes inaccurate inter-cluster boundary. Solution variance can be significantly reduced if sampling is increasingly concentrated only on the subset of objects at iterations $t_2 > t_1 > t_0$, as demonstrated in Figure 1.

The key issue in the design of the adaptation mechanism is the updating of probabilities. We have to decide how and which data points should be sampled as we collect more and more clusterings in the ensemble. A consensus function based on the co-association values [1] provides the necessary guidelines for adjustments of sampling probabilities. Remember that the co-association similarity between two data points x and y is defined as the number of clusters shared by these points in the partitions of an ensemble Π:

$$sim(x,y)=\frac{1}{H}\sum_{i=1}^{H}\delta(\pi_i(x),\pi_i(y)),\quad \delta(a,b)=\begin{cases}1, & \text{if } a=b\\ 0, & \text{if } a\neq b\end{cases} \qquad (3)$$

A consensus clustering can be found by using an agglomerative clustering algorithm (e.g., single linkage) applied to such a co-association matrix constructed from all the points. The quality of the consensus solution depends on the accuracy of similarity values as estimated by the co-association values. The least reliable co-association values come from the points located in the problematic areas of the feature space. Therefore, our adaptive strategy is to increase the sampling probability for such points as we proceed with the generation of different partitions in the ensemble.

The sampling probability can be adjusted not only by analyzing the co-association matrix, which is of quadratic complexity $O(N^2)$, but also by applying the less expensive $O(N+K^3)$ estimation of clustering consistency for the data points. Again, the motivation is that the points with the least stable cluster assignments, namely those that frequently change the cluster they are assigned to, require increased presence in the data subsamples. In this case, a label correspondence problem must be approximately solved to obtain the same labeling of clusters throughout the ensemble's partitions. By default, the cluster labels in different partitions are arbitrary. To make a correspondence problem more tractable, one needs to re-label each partition in the ensemble using some fixed reference partition. Table 1 illustrates how 4 different partitions of twelve points can be re-labeled using the first partition as a reference.

Table 1: Consistent re-labeling of 4 partitions of 12 objects.

	π_1	π_2	π_3	π_4	π_1'	π_2'	π_3'	π_4'	Consistency
$\mathbf{x}_1$	2	B	X	α	2	1	2	1	0.5
$\mathbf{x}_2$	2	A	X	α	2	2	2	1	0.75
$\mathbf{x}_3$	2	A	Y	β	2	2	1	2	0.75
$\mathbf{x}_4$	2	B	X	β	2	1	2	2	0.75
$\mathbf{x}_5$	1	A	X	β	1	2	2	2	0.75
$\mathbf{x}_6$	2	A	Y	β	2	2	1	2	0.75
$\mathbf{x}_7$	2	B	X	α	2	1	2	1	0.5
$\mathbf{x}_8$	1	B	Y	α	1	1	1	1	1
$\mathbf{x}_9$	1	B	Y	β	1	1	1	2	0.75
$\mathbf{x}_{10}$	1	A	Y	α	1	2	1	1	0.75
$\mathbf{x}_{11}$	2	B	Y	α	2	1	1	1	0.75
$\mathbf{x}_{12}$	1	B	Y	α	1	1	1	1	1

At the (t+1)-st iteration, when some t different clusterings are already included in the ensemble, we use the Hungarian algorithm for minimal weight bipartite matching problem in order to re-label the $(t+1)$st partition.

As an outcome of the re-labeling procedure, we can compute the consistency index of clustering for each data point. Clustering consistency index CI at iteration t for a point x is defined as the ratio of the maximal number of times the object is assigned in a certain cluster to the total number of partitions:

$$CI(x) = \frac{1}{H} \max \left\{ \sum_{i=1}^{H} \delta(\pi_i(x), L) \right\}_{L \in \text{cluster labels}} \qquad (4)$$

The values of consistency indices are shown in Table 1 after four partitions were generated and re-labeled. We should note that clustering of subsamples of the data set D does not provide the labels for the objects missing (not drawn) in some subsamples. In this situation, the summation in Eq. (4) skips the terms containing the missing labels.

The clustering consistency index of a point can be directly used to compute its sampling probability. In particular, the probability value is adjusted at each iteration as follows:

$$p_{t+1}(x) = Z(\alpha p_t(x) + 1 - CI(x)), \qquad (5)$$

where α is a discount constant for the current sampling probability and Z is a normalization factor. The discount constant was set to α=0.3 in our experiments. The proposed clustering ensemble algorithm is summarized in pseudocode below:

```
Input: $D$ – data set of $N$ points,
$H$ – number of partitions to be combined
$M$ – number of clusters in the consensus partition $\sigma$,
$K$ – number of clusters in the partitions of the ensemble,
$\Gamma$ – chosen consensus function operating on cluster labels
$\mathbf{p}$ – sampling probabilities (initialized to $1/N$ for all the points)
Reference Partition $\leftarrow$ $k$-means($D$)
for $i$=1 to $H$
  Draw a subsample $X_i$ from $D$ using sampling probabilities $\mathbf{p}$
  Cluster the sample $X_i$: $\pi(i) \leftarrow k$-means($X_i$)
  Re-label partition $\pi(i)$ using the reference partition
  Compute the consistency indices for the data points in $D$
  Adjust the sampling probabilities $\mathbf{p}$
end
Apply consensus function $\Gamma$ to ensemble $\Pi$ to find the partition $\sigma$
Validate the target partition $\sigma$ (optional)
return $\sigma$ // consensus partition
```

3. Empirical study and discussion

The experiments were conducted on artificial and real-world data sets ("Galaxy", "half-rings", "wine", "3-gaussian", "Iris", "LON"), with known cluster labels, to validate the accuracy of consensus partition. A comparison of the proposed adaptive and previous non-adaptive [9] ensemble is the primary goal of the experiments. We evaluated the performance of the clustering ensemble algorithms by matching the detected and the known partitions of the datasets. The best possible matching of clusters provides a measure of performance expressed as the misassignment rate. To determine the clustering error, one needs to solve the correspondence problem between the labels of known and derived clusters. Again, the Hungarian algorithm was used for this purpose. The *k*-means algorithm was used to generate the partitions of samples of size N drawn with replacement, similar to bootstrap, albeit with dynamic sampling probability. Each experiment was repeated 20 times and average values of error (misassignment) rate are shown in Figure 2.

Consensus clustering was obtained by four different consensus functions: hypergraph-based MCLA and CSPA algorithms [2], quadratic mutual information [3] and EM algorithm based on mixture model [4]. However, due to space limitations, we report only the key findings here. The main observation is that adaptive ensembles slightly

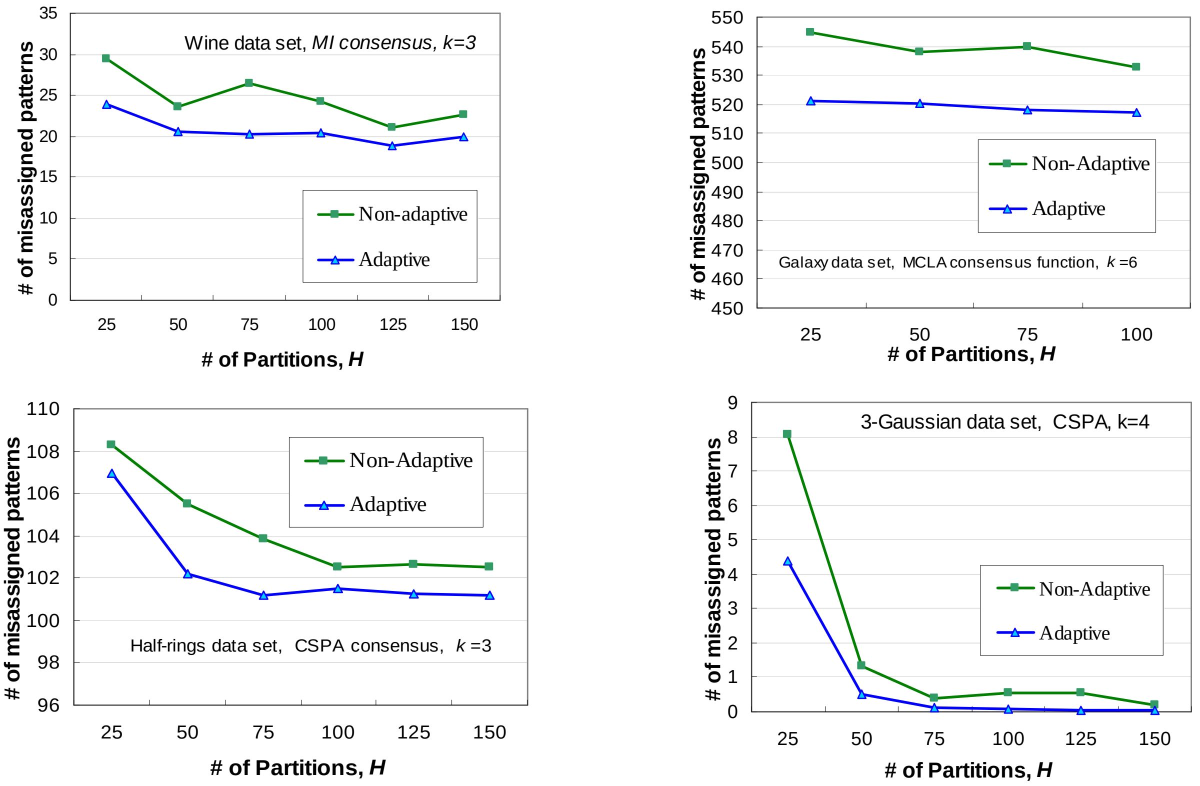

Figure 2. Clustering accuracy for ensembles with adaptive and non-adaptive sampling mechanisms as a function of ensemble size for some data sets and selected consensus functions.

outperform the regular sampling schemes on most benchmarks. Typically, the clustering error decreased by 1-5%. Accuracy improvement depends on the number of clusters in the ensemble partitions (K). Generally, the adaptive ensembles were superior for values of K larger than the target number of clusters, M, by 1or 2. With either too small or too large a value of K, the performance of adaptive ensembles was less robust and occasionally worse than corresponding non-adaptive algorithms. A simple inspection of probability values always confirmed the expectation that points with large clustering uncertainty are drawn more frequently.

Most significant progress was detected when combination consisted of 25-75 partitions. Large numbers of partitions (H>75) almost never lead to further improvement in clustering accuracy. Moreover, for H>125 we often observed increased error rates (except for the hypergraph-based consensus functions), due to the increase in complexity of the consensus model and in the number of model parameters requiring estimation.

To summarize, we have extended clustering ensemble framework by adaptive data sampling mechanism for generation of partitions. We dynamically update sampling probability to focus on more uncertain and problematic points by on-the-fly computation of clustering consistency. Empirical results demonstrate improved clustering accuracy and faster convergence as a function of the number of partitions in the ensemble.

4. References

[1] A.L.N. Fred and A.K. Jain, “Data Clustering Using Evidence Accumulation”, Proc. of the *16th Intl. Conf. on Pattern Recognition*, *ICPR* 2002, Quebec City, pp. 276 – 280, 2002.

[2] A. Strehl and J. Ghosh, “Cluster ensembles - a knowledge reuse framework for combining multiple partitions”. *Journal on Machine Learning Research*, 3, pp. 583-617, 2002

[3] A. Topchy, A.K. Jain, and W. Punch, “Combining Multiple Weak Clusterings”, Proc. 3d *IEEE Intl. Conf. on Data Mining*, 331-338, 2003

[4] A. Topchy, A.K. Jain, and W. Punch “A Mixture Model for Clustering Ensembles”, in Proc. *SIAM Intl. Conf. on Data Mining*, SDM 04, 379-390, 2004.

[5] S. Dudoit and J. Fridlyand, “Bagging to improve the accuracy of a clustering procedure”, *Bioinformatics*, 19 (9), pp. 1090-1099, 2003

[6] B. Fischer, J.M. Buhmann, “Bagging for Path-Based Clustering”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25 (11), 1411-1415, 2003.

[7] X. Fern, and C. E. Brodley, “Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach”, In Proc. *20th Int. conf. on Machine Learning*, *ICML* 2003.

[8] J.-P. Barthelemy and B. Leclerc, The median procedure for partition, In *Partitioning Data Sets, AMS DIMACS Series in Discrete Mathematics*, I.J. Cox et al eds., 19, pp. 3-34, 1995.

[9] B Minaei, A. Topchy and W. F. Punch, “Ensembles of Partitions via Data Resampling”, in Proc. Intl. Conf. on Information Technology, ITCC 04, Las Vegas, 2004.

[10] L. Breiman, “Arcing classifiers”, *The Annals of Statistics*, 26(3), 801-849,1998.

[11] A.K. Jain and J.V. Moreau, “The Bootstrap Approach to Clustering”, in *Pattern Recognition Theory and Applications*, P.A. Devijver and J. Kittler (eds.), Springer-Verlag, 1987, pp. 63-71.