A Comparison of Resampling Methods for Clustering Ensembles

Behrouz Minaei-Bidgoli
Computer Science Department
Michigan State University
East Lansing, MI, 48824, USA

Alexander Topchy
Computer Science Department
Michigan State University
East Lansing, MI, 48824, USA

William F. Punch
Computer Science Department
Michigan State University
East Lansing, MI, 48824, USA

Abstract — *Combination of multiple clusterings is an important task in the area of unsupervised learning. Inspired by the success of supervised bagging algorithms, we propose a resampling scheme for integration of multiple independent clusterings. Individual partitions in the ensemble are sequentially generated by clustering specially selected subsamples of the given data set. In this paper, we compare the efficacy of both subsampling (sampling without replacement) and bootstrap (with replacement) techniques in conjunction with several fusion algorithms. The empirical study shows that a meaningful consensus partition for an entire set of data points emerges from multiple clusterings of subsamples of small size. The purpose of this paper is to show 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 and measurement complexity for many unsupervised data mining tasks with distributed sources of data.*

Keywords — Subsampling, Bootstrap, consensus functions, clustering combination, distributed data mining.

1. Introduction

One of the major challenges for current clustering algorithms is the robustness of the derived solutions. Both supervised and unsupervised learning can be significantly improved by utilization of multiple solutions [1], [2]. However, the problem of finding a combination of clustering results is fundamentally different from combining multiple classifications in a supervised framework [3]. In the absence of training data, clustering algorithms face a difficult problem, namely the correspondence between labels in different partitions. Recent research [2], [3] on the combination of clusterings has addressed this issue by formulating consensus functions that avoid an explicit solution to the correspondence problem. Clustering ensembles require a partition generation process.

One of the main goals of clustering research is to design scalable and efficient algorithms for large datasets [4]. One solution to the scaling problem is the parallelization of clustering by sharing processing among different processors [5], [6]. Recent research in data mining has considered a fusion of the results from multiple sources of data or from data features obtained in a distributed environment [7]. Distributed data clustering deals with the combination of partitions from many data subsets (usually disjoint). The combined final clustering can be constructed centrally either by combining explicit cluster labels of data points or, implicitly, through the fusion of cluster prototypes (e.g. centroid-based). We analyze the first approach, namely, the clustering combination via consensus functions operating on multiple labelings of a given dataset's different subsamples. This study seeks to answer the question of the optimal size and granularity of the component partitions.

Several methods are known to create partitions for clustering ensembles. Taxonomy of solutions for the generative procedure as well as different consensus functions for clustering combination is shown in Figure 1. Many approaches proposed to generate different multiple clustering from a give data set; applying various clustering algorithms [2], using one algorithm with different built-in initialization and parameters [3], [8], projecting data onto different subspaces [3], [13], choosing different subset of features [3], and selecting different subsets of data points [10], [14] are instances of these generative mechanism. Several consensus functions, Co-association-based methods [8], [12], Voting approach [9], [10], Information-theoretic methods (e.g. Quadratic Mutual Information) [3], and mixture model [11] are developed to discover the final clustering solution from many multiple partitions. Details of the algorithms can be found in the listed references.

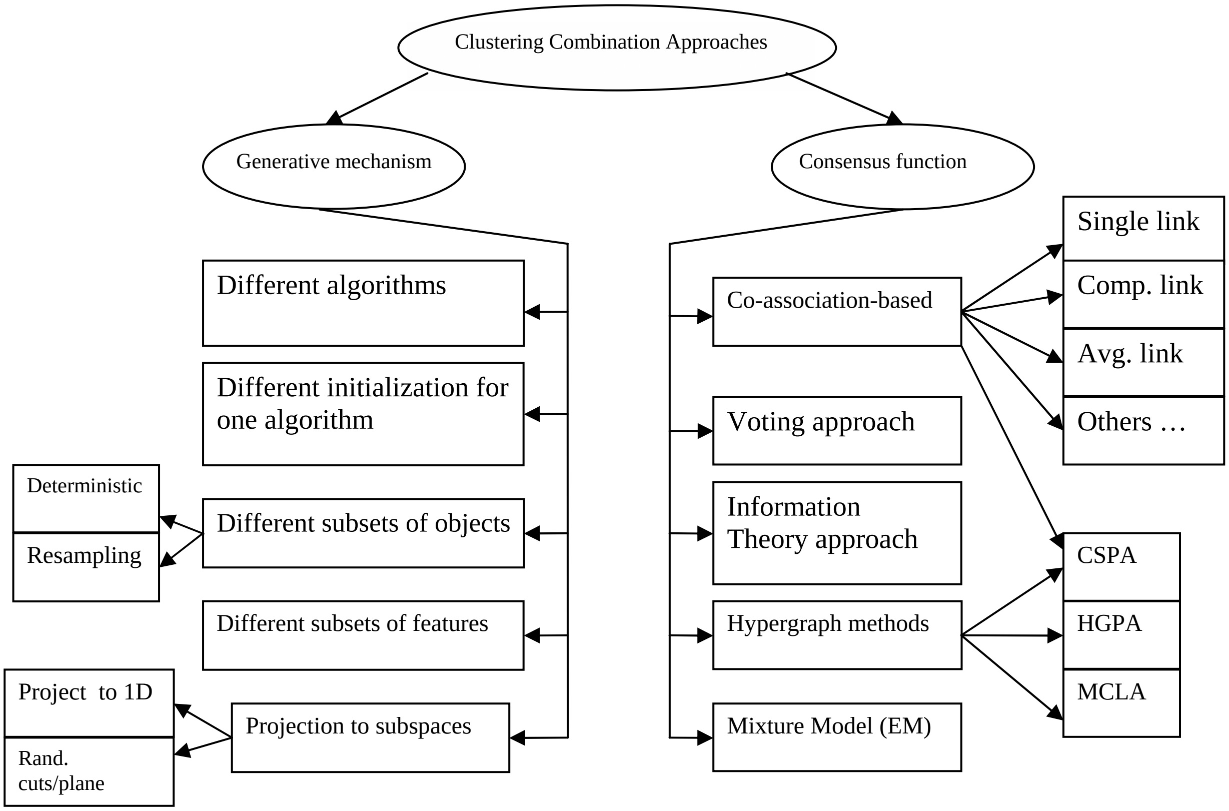

Figure 1. Taxonomy of different approaches to clustering combination; right side: different approaches how to obtain the diversity in clustering; left side: different consensus function to find the clustering ensemble.

2. Clustering ensemble algorithm

The problem of clustering combination can be formalized as follows. Let D be a data set of N data points in d-dimensional space. The input data can be represented as an $N \times d$ pattern matrix or $N \times N$ dissimilarity matrix, potentially in a non-metric space. Suppose that $X = \{X_1,\ldots,X_B\}$ is a set of bootstrap samples or subsamples of input data D. A chosen clustering algorithm is run on each of the samples in X that results in B partitions $P = \{P_1,\ldots, P_B\}$. Each component partition in P is a set of clusters $P_i=\{C_1^i, C_2^i, \ldots, C_{K(i)}^i\}$, $X_i = C_1^i \cup \ldots \cup C_{k(i)}^i$, $\forall P_i$, 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 P, such that the objects in a cluster of σ are more similar to each other than to objects in different clusters of σ. In order to find this target partition, σ, one needs to design a consensus function utilizing information from the partitions in $\{P_1,\ldots, P_B\}$.

A consensus function maps a given set of partitions $P = \{P_1, \ldots, P_B\}$ to a target partition σ using similarity values. The similarity between two objects x and y is defined as follows:

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

Similarity between a pair of objects simply counts the number of clusters shared by these objects in the partitions $\{P_1,\ldots, P_{\mathbf{B}}\}$. We have chosen the k-means algorithm as the partition generation mechanism, mostly for its low computational complexity.

```
Input:
$D$ – the input data set $N$ d-dimensional data,
$B$ - number of partitions to be combined
$M$ – number of clusters in the final partition $\sigma$,
$k$ – number of clusters in the components of the combination,
$\Gamma$ - a similarity-based clustering algorithm
for  j=1 to B
    Draw a random pseudosample $X_j$
    Cluster the sample $X_j$: $P(i) \leftarrow k$-means($\{X_j\}$)
    Update similarity values (co-association matrix) for all patterns in $X_j$
end
Combine partitions via chosen $\Gamma$: $\sigma \leftarrow \Gamma(P)$
Validate final partition $\sigma$ (optional)
return $\sigma$   // consensus partition
```

Figure 2. The algorithms for clustering ensemble, based on co-association matrix and using different similarity-based consensus functions

Under the assumption that diversity comes from resampling, two families of algorithms can be proposed for integrating clustering components. The first family is based on the co-association matrix, which employs a group of hierarchical clustering algorithms to find the final target partition. The pseudocode of these algorithms is shown in Figure 2. In the algorithms of this type, similarity-based clustering algorithms are used as the consensus function, Γ. Hierarchical clustering consensus functions with single-link (SL), complete-link (CL), and average-link (AL) criteria were used to obtain a target consensus clustering, σ. The second family of algorithms deals with the consensus functions, which are not based on co-association matrix. Details of the algorithms can be found in [8] and [9].

In the case of the subsampling algorithm (without replacement), the right choice of sample size S is closely related to the value of k and the value of B and proper setting of S is required to reach convergence to the true structure of the data set. The algorithm parameters will be discussed in section 5. In the rest of this paper k stands for number of clusters in every partition, B for number of partitions/pseudosamples (in both the bootstrap and the subsampling algorithms), and S, for the sample size.

3. Experimental results and discussion

The experiments were performed on several data sets, including a challenging artificial problem, the “Halfrings” data set, two data sets from the UCI repository, “Iris” and “Wine,” and two other real world data sets, the “LON” [14] and “Star/Galaxy” data sets. A summary of data set characteristics is shown in Table 1.

Table 1. A summary of data sets characteristics

	No. of Classes	*No. of Features*	*No. of Patterns*	*Patterns per class*
Halfrings	2	2	400	100-300
Star/Galaxy	2	14	4192	2082-2110
Wine	3	13	178	59-71-48
LON	2	6	227	64-163
Iris	3	4	150	50-50-50

For all data sets the number of clusters, and their label assignments, are known. Therefore, one can use the misassignment (error) rate of the final combined partition as a measure of performance of clustering combination quality. One can determine the error rate after solving the correspondence problem between the labels of derived and known clusters. The Hungarian method for minimal weight bipartite matching problem can efficiently solve the correspondence problem with. $O(k^3)$ complexity for k clusters. Consensus clustering was obtained by eight different consensus functions: hypergraph-based MCLA, HPGA and CSPA algorithms [2], quadratic mutual information (QMI) [3], and different co-association-based consensus functions including single-link (SL), average-link (AL), and complete-link (CL).

3.1. Algorithm’s parameters

The accuracy of consensus partition is a function of the resolution of partitions (value of k) and the number of partitions, B, to be merged. We studied the dependence of overall performance on the number of clusters, k. In particular, clustering on the subsamples and bootstrap samples was performed for the values of B in the range [5, 1000] and the values of k in the interval [2, 20]. Analogously, the size of the pseudosample, S in subsampling experiments, is another important parameter. The experiments were performed with different subsample sizes in the interval [N/20, 3N/4], where N is the size of original data sample. Thus, in the case of the “Halfrings”, S was taken in the range [20, 300] where the original sample size is N=400 while in the case of the “Galaxy” data set, S was varied in the range [200, 3000] where N=4192. Therefore, in resampling without replacement, we analyzed how the clustering accuracy was influenced by three parameters: number of clusters, k, in every clustering, number of drawn samples, B, and the sample size, S. It is worthwhile to note that all the experiments were repeated 20 times and the average error rate for 20 independent runs is reported, except for the "Star/Galaxy" set, where 10 runs were performed.

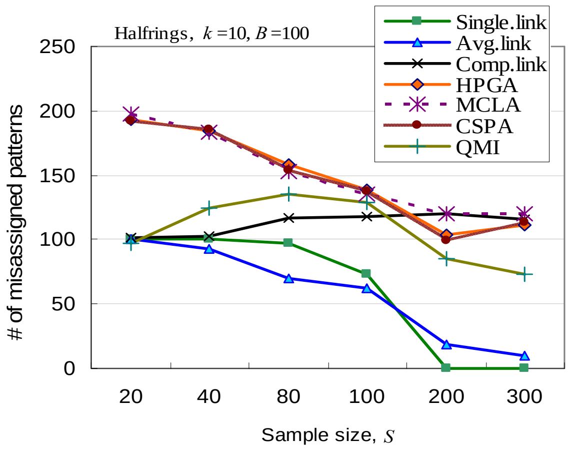

Figure 3. “Halfrings” data set. Experiments using subsampling with k=10 and B=100, different consensus functions and sample sizes.

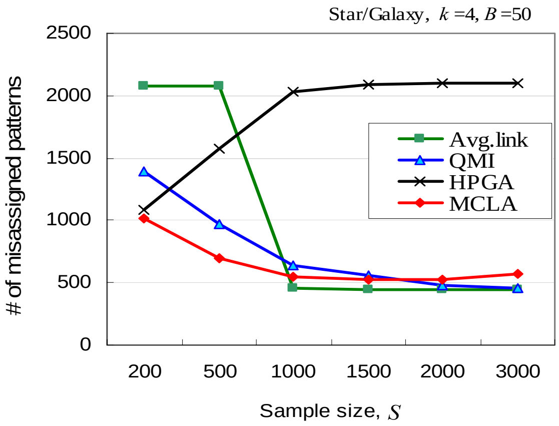

Figure 4. “Star/Galaxy” data set. Experiments using subsampling, with k = 4 and B = 50 and different consensus functions and sample sizes.

Note that in both the bootstrap and the subsampling algorithms all of the samples are drawn independently, and thus the resampling process could be performed in parallel. Therefore, using the B-parallel process, the computational process becomes B times faster.

Table 2. Summary of the best results of Bootstrap methods

Data set	***Best Consensus function(s)***	***Lowest Error rate obtained***	***Parameters***
Halfrings	Co-association, SL Co-association, AL	0% 0%	$k \geq 10, B. \geq 100$ $k \geq 15, B \geq 100$
Iris	Hypergraph-HGPA	2.7%	$k \geq 10, B \geq 20$
Wine	Hypergraph-CSPA Co-association, AL	26.8% 27.9%	$k \geq 10, B \geq 20$ $k \geq 4, B \geq 100$
LON	Co-association, CL	21.1%	$k \geq 4, B \geq 100$
Galaxy/ Star	Hypergraph-MCLA Co-association, AL Mutual Information	9.5% 10% 11%	$k \geq 20, B \geq 10$ $k \geq 10, B \geq 100$ $k \geq 3, B \geq 20$

Table 3. Subsampling methods: trade-off among the values of *k*, the number of partitions *B*, and the sample size, *S*. Last column denote the percentage of sample size regarding the entire data set. (Bold represents most optimal)

Data set	***Best Consensus function(s)***	***Lowest Error rate***	***K***	***B***	***S***	**% of entire data**
Halfrings	SL	0%	10	100	200	50%
	SL	0%	10	500	80	**20%**
	AL	0%	15	1000	80	20%
	AL	0%	20	500	100	25%
Iris	HGPA	2.3%	10	100	50	33%
	HGPA	2.1%	15	50	50	**33%**
Wine	AL	27.5%	4	50	100	56%
	HPGA	28%	4	50	20	**11%**
	CSPA	27.5%	10	20	50	28%
LON	CL	21.5%	4	500	100	44%
	CSPA	21.3%	4	100	100	**44%**
Galaxy/ Star	MCLA	10.5%	10	50	1500	36%
	MCLA	11.7%	10	100	200	**5%**
	AL	11%	10	100	500	12%

In subsampling, the smaller the *S*-value, the lower the complexity of the *k*-means clustering; therefore, the result is a much smaller complexity in the co-association based consensus functions, which are super-linear, *N*.

3.2. Subsampling vs. Bootstrapping

Comparing the results of the bootstrap and the subsampling methods shows that when the bootstrap technique converges to an optimal solution, that optimal result could be obtained by the subsampling as well, but with data points of a critical size. For example, in the "Halfrings" data set the perfect clustering can be obtained using a single-link consensus function with *k*=10, *B*=100 and *S*=200 (1/2 total set size) as shown in Figure 3 (compare to the bootstrap results in Table 2) while perfect results can be achieved with *k*=15, *B* = 50, and *S* = 80 (1/5 total set size). Thus, there is a trade off between the number of partitions, *B*, and the sample size, *S*. This comparison shows that the subsampling method can be much faster than the bootstrap (*N*=400) relative to computational complexity. The results of subsampling for the "Star/Galaxy" data set in Figure 4 show that in resolution *k*=3 and number of partitions *B*=100, with only sample size *S* = 500 (1/8 total set size), one can reach 89% accuracy – the same results required the entire data set in the bootstrap method. This implies that in a large data set, a small fraction of data can be representative of the entire data set, a result that holds great computational promise for distributed data mining.

The optimal sample size, *S*, and granularity of the component partitions derived by subsampling are reported in Table 3. We see that the accuracy of the resampling method is very similar to that of the bootstrap algorithm, as reported in Table 2. This equivalent level of accuracy was reached with remarkably smaller sample sizes and much lower computational complexity! The trade-off between the accuracy of the overall clustering combination and computational effort for generating component partitions is shown in Table 3, where we compare accuracy of consensus partitions. The most promising result is that only a small fraction of data (i.e., 12% or 5% for the "Star/Galaxy" data set) is required to obtain the optimal solution of clustering, both in terms of accuracy and computational time.

4. Conclusion

This study shows that meaningful consensus partitions for the entire data set of objects emerge from clusterings of bootstrap and subsamples of small size. The results demonstrate that there is a trade-off between the number of clusters per component and the number of partitions, and that the sample size of each partition needed in order to perform the combination process converges to an optimal value with respect to error rate. The bootstrap technique employed herein was recently applied in [9], [10], [12], and [14] with similar results and aims – namely, to create diversity in clustering ensembles. However, our work extends that of previous research by using a more flexible subsampling algorithm for ensemble generation; subsamples of small size can reduce computational cost and measurement complexity for many explorative data mining tasks with distributed sources of data. We also provide a detailed comparative study of several consensus techniques. The challenging aspects of using resampling techniques for maintaining diversity of partitions are also discussed in this paper. We show that there exists a critical fraction of data such that the structure of an entire data set is perfectly detectable. Further study of alternative resampling methods, such as the balanced (stratified) and recentered bootstrap methods are critical in generalizing these results.

5. References

[1] L. Breiman, “Bagging Predictors”, *Journal of Machine Learning*, Vol 24, no. 2, 1996, pp 123-140.

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

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

[4] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: an efficient data clustering method for very large databases”, *ACM SIGMOD Record*, 1996, 25 (2): 103-114.

[5] B. Zhang, M. Hsu, G. Forman, "Accurate Recasting of Parameter Estimation Algorithms using Sufficient Statistics for Efficient Parallel Speed-up Demonstrated for Center-Based Data Clustering Algorithms", *Proc. 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, in Principles of Data Mining and Knowledge Discovery*, D. A. Zighed, J. Komorowski and J. Zytkow (Eds.), 2000.

[6] I. S. Dhillon and D. S. Modha, “A Data-clustering Algorithm on Distributed Memory Multiprocessors”, In Proceedings of *Large-scale Parallel KDD Systems Workshop, ACM SIGKDD, in Large-Scale Parallel Data Mining, Lecture Notes in Artificial Intelligence*, 2000, 1759: 245-260.

[7] B.H. Park and H. Kargupta, “Distributed Data Mining”. In The *Handbook of Data Mining*. Ed. Nong Ye, Lawrence Erlbaum Associates, 2003.

[8] A.L.N. Fred and A.K. Jain, “Data Clustering Using Evidence Accumulation”, In Proc. of the *16th International Conference on Pattern Recognition*, *ICPR* 2002 ,Quebec City: 276 – 280.

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

[10] B. Fischer, J.M. Buhmann, “Path-Based Clustering for Grouping of Smooth Curves and Texture Segmentation”, *IEEE Trans. on PAMI*, 25 (4), pp. 513-518, 2003

[11] A. Topchy, A.K. Jain, and W. Punch, “A Mixture Model of Clustering Ensembles”, *SIAM Intl. Conf. on Data Mining* 2003.

[12] S. Monti, P. Tamayo, J. Mesirov, T. Golub, “Consensus Clustering: A reamlping-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data”, *Journal on Machine Learning*, Volume 52 Issue 1-2, July 2003.

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

[14] B. Minaei-Bidgoli, A. Topchy and W. F. Punch, “Ensembles of Partitions via Data Resampling”, *Proc. Intl. Conf. on Information Technology, ITCC 2004*, pp. 188-192, Las Vegas, 2004.