Ensembles of Partitions via Data Resampling

Behrouz Minaei-Bidgoli, Alexander Topchy, William F. Punch
Computer Science Department, Michigan State University, East Lansing, MI, 48824, USA
{minaeibi, topchyal, punch}@cse.msu.edu

Abstract

The combination of multiple clusterings is a difficult problem in the practice of distributed data mining. Both the cluster generation mechanism and the partition integration process influence the quality of the combinations. In this paper we propose a data resampling approach for building cluster ensembles that are both robust and stable. In particular, we investigate the effectiveness of a bootstrapping technique in conjunction with several combination algorithms. The empirical study shows that a meaningful consensus partition for an entire set of objects emerges from multiple clusterings of bootstrap samples, given optimal combination algorithm parameters. Experimental results for ensembles with varying numbers of partitions and clusters are reported for simulated and real data sets. Experimental results show improved stability and accuracy for consensus partitions obtained via a bootstrapping technique.

1. Introduction

In order to optimally integrate clustering ensembles in a robust and stable manner, one needs a diversity of component partitions for combination. Generally, this diversity can be obtained from several sources:

1) Using different clustering algorithms to produce partitions for combination.
2) Changing initialization or other parameters of a clustering algorithm.
3) Using different features via feature extraction for subsequent clustering.
4) Partitioning different subsets of the original data.

The focus of this paper is the later method, namely the combination of clusterings using random samples of the original data. Our main motivation is two-fold. First, different data subsets should form different clusters. When these clusters are combined, the multiple partitions provide a more stable cluster structure than any single clustering. Second, one can provide a confidence estimate for the assignment of an object to a particular cluster in the consensus partition given that object's assignment in each of the bootstrap partitions. These assertions will be quantified in the empirical study.

A growing number of techniques have been applied to the combination of clusterings. A co-association consensus function was introduced for finding a combined partition in [1]. The authors further studied combining *k*-means partitions with random initializations and a random number of clusters. Topchy et al. proposed new consensus functions related to intra-class variance criteria as well as the use of weak clustering components [2], [3]. Strehl and Ghosh have made a number of important contributions, such as their detailed study of hypergraph-based algorithms for finding consensus partitions as well as considering object-distributed and feature-distributed formulations of the problem [4]. They also examined the combination of partitions with a deterministic overlap of points between data subsets (non-random).

Resampling methods have been traditionally used to obtain more accurate estimates of data statistics. Efron generalized the concept of so-called "pseudo-samples" to sampling *with* replacement – the *bootstrap* method [5]. Resampling methods such as bagging have been successfully applied in the context of supervised learning [6]. Jain and Moreau employed bootstrapping in cluster analysis to estimate the number of clusters in a multi-dimensional data set as well as for evaluating cluster tendency/validity [7]. A measure of consistency between two clusters is defined in [8]. Data resampling has been used as a tool for estimating the validity of clustering [9], [10] and its reliability [11], [12].

2. Consensus functions

In this paper we have employed four types of consensus functions:

Co-association based functions: The consensus function operates on the co-association matrix. Numerous hierarchical agglomerative algorithms (criteria) can be applied to the co-association matrix to obtain the final partition, including Single Link (SL), Average Link (AL) and Complete Link (CL). Note that the computational complexity of co-association based consensus algorithms is very high, $O(kN^2d^2)$ [15].

Quadratic Mutual Information Algorithm (QMI): Assuming that the partitions are independent, a consensus function based on *k*-means clustering in the space of standardized features can effectively maximize a generalized definition of mutual information [3]. The complexity of this consensus function is O(*kNB*), where *B* is the number of partitions. Though the QMI algorithm can be potentially trapped in a local optimum, its relatively low computational complexity allows using multiple restarts in order to choose a quality consensus solution with minimum intra-cluster variance.

Hypergraph partitioning: The clusters could be represented as hyperedges on a graph whose vertices correspond to the objects to be clustered. The problem of consensus clustering is then reduced to finding the minimum-cut of a hypergraph. The minimum *k*-cut of this hypergraph into *k* components gives the required consensus partition [4]. Efficient heuristics to solve the *k*-way min-cut partitioning problem are known, some with computational complexity on the order of O(|ε|), where ε is the number of hyperedges. Three hypergraph algorithms, CSPA, HGPA, and MCLA, are described in [4] and their corresponding source code are available at http://www.strehl.com.

Voting approach: In the previous algorithms there is no need to explicitly solve the correspondence problem between the labels of known and derived clusters. The voting approach attempts to solve the correspondence problem and then uses a majority vote to determine the final consensus partition [11]. The main idea is to permute the cluster labels such that best agreement between the labels of two partitions is obtained. All the partitions from the ensemble must be re-labeled according to a fixed reference partition. The complexity of this process is *k*!, which can be reduced to $O(k^3)$ if the Hungarian method is employed for the minimal weight bipartite matching problem.

The performance of all these consensus methods are empirically analyzed as a function of two important parameters: the type of sampling process (the redundancy of a sample) and the granularity of each partition (number of clusters).

This study seeks to answer the following questions:

1) What is the trade-off between the accuracy of the overall clustering combination and computational effort required for generating component partitions?
2) What is the optimal size and granularity of the component partitions?
3) What is the best possible consensus function to combine bootstrap partitions in a given data set?

3. Clustering ensemble algorithm

Bootstrap sampling and subsampling can discern various statistics from replicate subsets of data. Our goal is to obtain a reliable clustering with measurable uncertainty from a set of different *k*-means partitions. The key idea of this approach is to integrate multiple partitions produced by clustering of pseudo-samples of a data set. Two issues, specific to the clustering combination, must be addressed:

1) The generative mechanism for individual partitions in the combination.
2) The choice of consensus function to combine several partitions.

We have chosen the *k*-means algorithm as the partition generation mechanism, mostly for its low computational complexity. In addition, eight different consensus functions from two families of such algorithms (co-association, feature extraction) were examined.

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, and employs a group of hierarchical clustering algorithms to find the final target partition. A more complete discussion of the first family can be found in [1], [11], and [14].

The second family of algorithms for clustering combination is based on new features extracted through the partitioning process. In this approach, one can view consensus clustering as clustering in a space of new features induced by the set of partitions, *P*. Each partition P_i represents a feature vector with categorical values. The cluster labels of each object in different partitions are treated as a new feature vector, a *B*-tuple, given *B* different partitions in *P*. Therefore, instead of the original *d* attributes, which are shown in Table 1(a), the new feature vectors from a table with *N* rows and *B* columns (Table 1(b)) have been employed.

Table 1. (a) Data points and feature values, *N* rows and *d* columns. Every row shows a feature vector corresponding to *N* points. (b) Cluster labels for resampled data, *n* rows and *B* columns, each column is a new feature with categorical (nominal) values.

(a)

Data	*Features*					
$\boldsymbol{x_1}$	x_{11}	x_{12}	…	x_{1j}	…	x_{1d}
$\boldsymbol{x_2}$	x_{21}	x_{22}	…	x_{2j}	…	x_{2d}
…	…	…	…	…	…	…
$\boldsymbol{x_i}$	x_{i1}	x_{i2}	…	x_{ij}	…	x_{id}
…	…	…	…	…	…	…
$\boldsymbol{x_N}$	x_{N1}	x_{N2}	…	x_{Nj}	…	x_{Nd}

(b)

Data	*Cluster Labels*					
$\boldsymbol{x_1}$	$P_1(x_1)$	$P_2(x_1)$	…	$P_j(x_1)$	…	$P_B(x_1)$
$\boldsymbol{x_2}$	$P_1(x_2)$	$P_2(x_2)$	…	$P_j(x_2)$	…	$P_B(x_2)$
…	…	…	…	…	…	…
$\boldsymbol{x_i}$	$P_1(x_i)$	$P_2(x_i)$	…	$P_j(x_i)$	…	$P_B(x_i)$
…	…	…	…	…	…	…
$\boldsymbol{x_N}$	$P_1(x_N)$	$P_2(x_N)$	…	$P_j(x_N)$	…	$P_B(x_N)$

Here, $P_j(x_i)$ denotes the label of object x_i in the j-th partition of P. Hence the problem of combining partitions becomes a categorical clustering problem.

Input:
D – the input data set N d-dimensional data,
B – number of partitions to be combine
M – number of clusters in the final partition σ,
k – number of clusters in the components of the combination,
Γ – consensus function operating with categorical features
Reference Partition ← k-means(D)
for i=1 to B
  Draw a random pseudo-sample X_j
  Cluster the sample X_j: $P(i)$←k-means($\{X_j\}$)
  Store partition P_i
end
Re-label (if necessary)
Apply consensus function Γ on partition labels $\{P\}$ to find final partition σ
Validate final partition σ (optional)
return σ *// consensus partition*

Figure 1. Algorithms for clustering ensemble, based on categorical clustering

The parameter k in the algorithms is the number of clusters in every component partition. If the value of k is too large then the partitions will overfit the data set, and if k is too small then the number of clusters may not be large enough to capture the true structure of data set. In addition, if the total number of clusterings, B, in the combination is too small then the effective sample size for the estimates of distances between co-association values is also insufficient, resulting in a larger variance. The algorithm parameters will be discussed in the next section. In the rest of this paper "k" stands for number of clusters in every partition, "B" for number of partitions/pseudo-samples.

4. Experimental results and discussion

The experiments were performed on several data sets, including two challenging artificial problems ("2-Spirals" and "Halfrings"), a classical data set from the UCI repository ("Iris"), and two other real world data sets ("LON" and "Star/Galaxy"). A summary of data set characteristics is shown in table 2.

4.1. Data sets

The "Halfrings" data set, as shown in figure 2, consists of two unbalanced clusters with 100 and 300 patterns. The k-means algorithm by itself is not able to detect the two natural clusters since it implicitly assumes hyper-spherical clusters.

The "2-spirals" dataset, as shown in Figure 2, exhibits complex cluster shapes. Again, the simple k-means cannot identify true clusters in this data set.

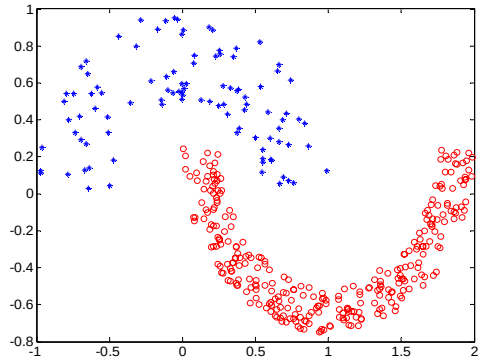

Figure 2. "Halfrings" data set with 400 patterns (100-300 per class)

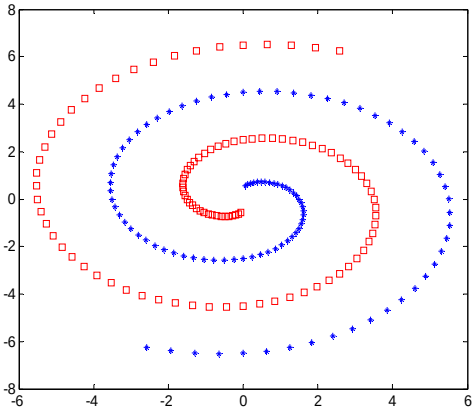

Figure 3. "2-Spirals" dataset with 200 patterns (100-100 per class)

Table 2. A summary of data sets characteristics

	No. of Classes	*No. of Features*	*No. of Patterns*	*Patterns per class*
Halfrings	2	2	400	100-300
2-Spirals	2	2	200	100-100
Star/Galaxy	2	14	4192	2082-2110
LON	2	6	227	64-163
Iris	3	4	150	50-50-50

The "*LON*" data set [12] is extracted from the activity log in a web-based course using an online educational system developed at Michigan State University (MSU): the Learning Online Network with Computer-Assisted Personalized Approach (LON-CAPA[1]). The data set includes the student and course information on an introductory physics course (PHY183), collected during the spring semester 2002. This course included 12 homework sets with a total of 184 problems, all of which were completed online using LON-CAPA. The data set consists of 227 student records from one of the two groups: "Passed" for the grades above 2.0, and "Failed"

[1] http://www.lon-capa.org

otherwise. Each sample contains 6 features. The *"Iris"* data set contains 150 samples in 3 classes of 50 samples each, where each class refers to a type of iris plant. One class is linearly separable from the other two. Each sample has four continuous-valued features. The "*Star/Galaxy*" data set described in [13] has a significantly larger number of samples (N=4192) and features (d=14). The task is to separate patterns of galaxies from stars. Domain experts manually provided true labels for these objects.

For all these data sets the number of clusters, and their 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.

4.2. The role of algorithm's parameters

It is important to note that the bootstrap experiments probe the accuracy of partition combination as a function of the resolution of partitions (value of k) and the number of partitions, B (number of partitions to be merged). One of our goals is to determine the minimum number of bootstrap samples, B, necessary to form high-quality combined cluster solutions. In addition, different values of k in the k-means algorithm provide different levels of resolution for the partitions in the combinations. We studied the dependence of overall performance on the number of clusters, k. In particular, clustering on the bootstrapped samples was performed for the values of B in the range [5, 1000] and the values of k in the interval [2, 20].

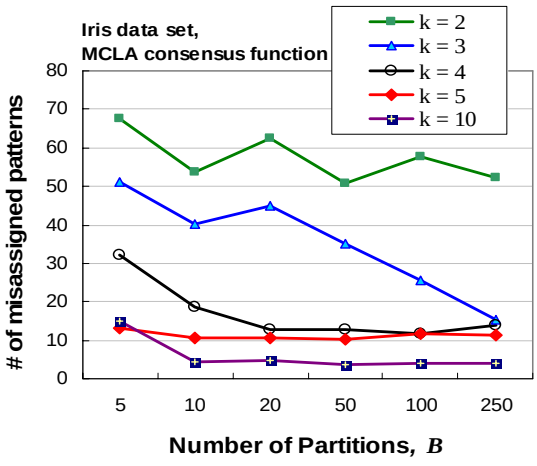

Figure 4. "Iris" data set. Bootstrapping for fixed consensus function MCLA, different partition numbers and different values of k.

The experiments employed eight different consensus functions: co-association based functions (single link, average link, and complete link), hypergraph algorithm (HGPA, CSPA, MCLA), QMI algorithm, as well as Voting-based function.

4.3. The Role of Consensus Functions

Perhaps the most important single design element of the combination algorithm is the choice of a consensus function. In the "*Halfrings*" and "*2-Spiral*" data sets the true structure of the data sets (100% accuracy) was obtained using co-association based consensus functions. (in the "Halfrings" data set with both AL and SL, and in the "2-Spiral" data set with SL, where B≥100, and k≥10).

For the "*LON*" data set the optimal accuracy of 79% was achieved only by co-association-based (using the AL algorithm) consensus function. This accuracy is comparable to the result of the k-NN classifier, multilayer perceptron, naïve Bayes classifier, and some other algorithms when the LON data set is classified in a supervised framework based on labeled patterns [12].

For the "*Iris*" data set, the hypergraph consensus function, HPGA algorithm led to the best results when k≥10. The AL and the QMI algorithms also gave acceptable results, while the single link and average link did not demonstrate a reasonable convergence. Figure 4 shows that the optimal solution could not be found for the "Iris" data set with k in [2..5], while optimum was reached for k ≥ 10 with only B=10 bootstrap partitions.

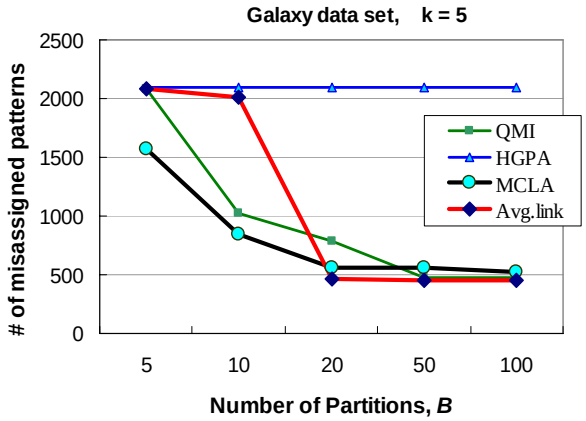

Figure 5. "Galaxy" data set. Bootstrapping for different consensus functions, different partition numbers B, and fixed value of k=5.

For the "*Star/Galaxy*" data set the CSPA function (similarity based hypergraph algorithm) could not be used due to its computational complexity. The HGPA function and SL did not converge at all. Voting and complete link also did not come up with an optimal solution. However, the MCLA, the QMI and the AL functions led to an error

rate of approximately 10%, which was much better than the performance of an individual *k*-means result (21%). (See Figure 5). Table 3 shows the error rate of classical clustering algorithms, which were used in this study. The error rates reported for the *k*-means algorithm were an average over 100 runs, with random initializations for the cluster centers, and where value of *k* was fixed to the true number of clusters. One can compare it to the error rate of ensemble algorithms in table 4.

Table 3. The average error rate (%) of classical clustering algorithms. An average over 100 independent runs is reported for the *k*-means algorithms

Data set	*k*-means	Single Link	Complete Link	Average Link
Halfrings	25%	24.3%	14%	5.3%
2 Spiral	43.5%	0%	48%	48%
Iris	15.1%	32%	16%	9.3%
LON	27%	27.3%	25.6%	27.3%
Star/Galaxy	21%	49.7%	44.1%	49.7%

Table 4. Summary of the best resluts of Bootstrap methods

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

The question of the best consensus function remains open for further study. Each consensus function explores the structure of data set in different ways, thus its efficiency greatly depends on different types of existing structure in the data set. One can suggest having several consensus functions and then combining the consensus function results through maximizing mutual information [2], but running different consensus functions on large data sets is computationally expensive.

To summarize, we proposed an approach to combine partitions through resampling of the original data. This study showed that meaningful consensus partitions for the entire set of objects emerges from clusterings of bootstrap samples. Empirical studies were conducted on five data sets for different consensus functions, number of partitions in the combination, and number of clusters in each component. The results demonstrate that there is a trade-off between the number of clusters per component and the number of partitions in the ensemble that can be optimized. Our work extends the previous research by providing a detailed comparative study of several consensus techniques in conjunction to different number of partitions. Future work can focus on alternative resampling methods such as subsampling (resampling without replacement), to determine whether subsamples of small size can reduce computational cost and measurement complexity for explorative, distributed-source data mining tasks.

5. References

[1] 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, pp. 276 – 280.

[2] A. Topchy, A.K. Jain, and W. Punch, "Combining Multiple Weak Clusterings", submitted to *IEEE Intl. Conf. on Data Mining*, 2003, Melbourne Florida, pp 331-338.

[3] A. Topchy, A.K. Jain, and W. Punch, "A Mixture Model of Clustering Ensembles", Proc. *SIAM Intl. Conf. on Data Mining* 2004, in press.

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

[5] B. Efron, "Bootstrap methods: Another Look at the Jackknife". *Annals of Statistics*, 1979, 7: pp. 1-26.

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

[7] 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.

[8] A. Ben-Hur, A. Elisseeff, and I. Guyon, "*A stability based method for discovering structure in clustered data*," in Pac. Symp. Biocomputing, 2002, vol. 7, pp. 6-17.

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

[10] E. Levine, and E. Domany, "Resampling method for unsupervised estimation of cluster validity". *Neural Computation*, 2001, 13, pp. 2573-2593.

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

[12] B. Minaei-Bidgoli, W.F. Punch, "Using Genetic Algorithms for Data Mining Optimization in an Educational Web-based System", In Proc. of *the Genetic and Evolutionary Computation Conference GECCO* 2003, pp. 2252-2263.

[13] S.C. Odewahn, E.B. Stockwell , R.L. Pennington, R.M. Humphreys, and W.A. Zumach. Automated Star/Galaxy Discrimination with Neural Networks, *Astronomical Journal*, 1992, 103, pp. 308-331.

[14] 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*, July 2003, Volume 52 Issue 1-2.

[15] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*. 2nd Edition, John Wiley & Sons Inc., New York NY, 2001.