X Congresso Brasileiro de Inteligência Computacional (CBIC'2011), 8 a 11 de Novembro de 2011, Fortaleza, Ceará

CONSENSUS CLUSTERING USING WEIGHTED ASSOCIATION

Aparajita Nanda
Sambalpur University Institute Of Information Technology, India
aparajita1.nanda@gmail.com

Arun K Pujari
Vice Chancellor, Sambalpur University, India
arun.k.pujari@gmail.com

Abstract – Consensus clustering has emerged as a method of improving quality and robustness in clustering by optimally combining the results of different clustering process. In last few years, several approaches are proposed. In this paper, we propose a new method of arriving at a consensus clustering. We assign confidence score to each partition in the ensemble and compute weighted co-association for all pairs of data objects. In order to derive the consensus clustering from the co-association matrix, we use cross-association technique to group the rows and columns simultaneously. The objective is to derive as many clusters of homogenous blocks as possible. The set of non-zero blocks are taken as the resulting partition. The use of cross-association technique captures the transitive relationship. We show empirically that for the benchmark datasets, our technique yields better consensus clustering than any other known algorithms.

Keywords – Clustering ensemble, Co-association, Cross-association.

1. INTRODUCTION

The problem of consensus clustering is concerned with arriving at a clustering by combining the outcomes of different runs of several clustering algorithms. The need for consensus clustering arises due to the fact that none of the existing clustering techniques can yield satisfactory partition for all instances of input data. Some of these algorithms also give different clustering based on different values of initial parameters and there is no way to determine the most appropriate values for a given situation. In such cases, consensus clustering (also reffered to as clustering ensembles) attempts to combine the results of different clustering obtained in different ways with an aim to get a better clustering. Many algorithms are proposed in recent years and one can broadly categorize these an hypergraph based,information theory based, mixture model based, voting based and co-association based.

In this paper, we propose a new way of arriving at a consensus clustering from a set of clustering with fixed number of clusters. Our approach is a combination of weighted co-association and cross-association. For a given clustering ensemble, we first determine the confidence score of each partition and use this as weights to compute weighted co-association for all pairs of data points. We derive a binary matrix by thresholding the weighted co-association matrix and then use a cross-association based method to partition the rows and columns. This simultaneous partitioning yields a set of nearly homogenous submatrices (almost all entries of the submatrix is either 1s or 0s). The submatrices corresponding non-zero entries yield a consensus partition. We use *k*-means clustering algorithm with different initial parameters to generate clustering ensemble. We have experimented with several benchmark datasets with known classification. We show that the proposed method yields better consensus clustering compared to many well-known consensus clustering algorithms.

The main contribution of present work is the use of cross-association along with co-association with confidence measure for consensus clustering. While computing consensus clustering of an ensemble, it is worthwhile to assign importance to each partition based on its confidence score. The motivation for assigning priorities comes from the realization that most of the existing algorithm determine a sort of median of ensemble of partition as the consensus and hence, the presence of noisy partition affects the quality of consensus drastically. By assigning priorities to each partition, influences of partitions on the resulting consensus can be controlled. We use the difference of intra-cluster and inter-cluster distance as the confidence score of a partition. The other novelty in the proposed method is the use of cross-association for deriving consensus from the weighted co-association. By using cross-association method, we make use of transitive qualitative co-association between the data points to arrive at a consensus. It is qualitative in the sense that we use a binary matrix to indicate whether a pair of objects shares the same cluster in a minimum number of partitions contrary to the quantitative approach of counting the number of partitions as a measure of co-association, It captures the transitive properties in the sense that, if there is co-association between objects O_1 and O_2, as well as between objects O_2 and O_3, then we tend to establish a similarity between objects O_1 and O_3 also.

The rest of the paper is organized as follows. Section 2 describes the notations and some existing methods, In section 3, we build the necessary concepts of co-association consensus clustering. In section 4 ,we give the cross-association based clustering to determine the consensus. Section 5, 6 describes the evaluation criteria and experimental results respectively, followed by conclusions in section 7.

2 NOTATIONS

consensus clustering has been proposed as a useful approach to strengthen the performance of simple clustering algorithms [1-3, 6-9, 10-14, 16-18]. The goal is to combine multiple, diverse, and independent clustering arrangements to obtain a single, comprehensive clustering. The problem of combining multiple clusterings can be describes as follows.
Let $\boldsymbol{O}$ be the set of data points, $\boldsymbol{O}$=$\{O_1, O_2, \ldots, O_n\}$ and a set of T partitions be $\boldsymbol{P} = \{P^1, P^2, \ldots, P^T\}$ of $\boldsymbol{O}$, where a partition P on $\boldsymbol{O}$ is defined as P=$\{C_1, C_2, \ldots, C_k\}$ such that $C_i \subseteq \boldsymbol{O}$, $\forall i$, $C_i \cap C_j = \emptyset$ and $\cup C_i = \boldsymbol{O}$. The objective is to find consensus partition $P^* = \{C_1^*, C_2^*, \ldots, C_k^*\}$ that optimizes a criterion (consensus) function. A consensus function maps a given set of partitions $P = \{P^1, P^2, \ldots, P^T\}$ to a final partition P. The P^* is a sort of median of $P^1, P^2, \ldots, P^T$ in the space of partitions.

In last few years, consensus clustering (also known as clustering ensembles) is one of the major topics of data mining research. There have been very large number of research papers published recently covering several aspects such as new algorithms, theoretical investigations and many novel applications.

Fred et al [7-8] combine clustering produced by multiple runs of k-means algorithm with different initializations into co-association matrix. Co-association between any pair of objects is the number of partitions in which two objects share the same cluster. The co-association values of all pairs of objects are used in a hierarchical single-link clustering algorithm to partition the datset into final consensus clusters. Topchy et al [19] formulate the consensus clustering problem into a maximum likelihood problem which is solved by the EM algorithm. Caruana et al [4] discuss ensemble selection from a library of trained models. Gionis et al [9] provide a formal definition to the problem of cluster aggregation and discuss a few consensus algorithm with theoretical guarantees. Topchy et al [20] present two approaches to prove the effectiveness of a cluster ensemble, using plurality voting and using a metric on the space of partitions. Strehl and Ghosh [18] define the cluster ensemble problem as an optimization problem and maximize the normalized mutual information of the consensus clustering.They introduce three different algorithms to obtained good consensus clustering, namely Cluster-based Similarity Partitioning (CSPA), HyperGraph partitioning (HGPA), Meta-clustering (MCLA) algorithms. Nguyen and Caruana [15] find that an iterative EM-like method is remarkably effective for consensus clustering problem. They introduce Iterative Voting Consensus (IVC) and its two variations, Iterative Probabilistic consensus (IPVC) and Iterative Pairwise Consensus (IPC) for finding clustering consensus. The mechanism of these iterative method is as follows: For each data points O_i in $\boldsymbol{O}$ a T dimensional feature vector is constructed, where the j^{th} feature is the cluster label of O_i in P_j. An iteration is essentially the reassignment of data points to different clusters based on the distance function.
We reiterate at this stage that the co-association information is used by many authors. In present work, we investigate the co-association based approach in a different perspective. We use weighted co-association and cluster data points based on cross-association. The cross-association based approach works on a very novel idea of partitioning a binary matrix into set of submatrices by maximizing the homogeneity of submatrices. This is accomplished by simultaneously partitioning the rows and columns of the binary matrix into row-groups and column-groups. We get the binary matrix from the weighted co-association by thresholding and use a practical algorithm of simultaneously partitioning the rows and columns.

3 WEIGHTED CO-ASSOCIATION

Let $\boldsymbol{O}$ be the set of data points, $\boldsymbol{O}$=$\{O_1, O_2, \ldots, O_n\}$ and a set of T partitions be $\boldsymbol{P} = \{P^1, P^2, \ldots, P^T\}$ of $\boldsymbol{O}$, where a partition P on $\boldsymbol{O}$ is defined as P=$\{C_1, C_2, \ldots, C_k\}$ such that $C_i \subseteq \boldsymbol{O}$, $\forall i$, $C_i \cap C_j = \emptyset$ and $\cup C_i = \boldsymbol{O}$. We denote m_j as the centroid (or mean) of the cluster C_j. Let us define $\lambda_i^{(q)}$ as the cluster to which O_i belongs in partition P^q.

Co-association

The co-association of a pair of objects O_i and O_j is defined as

$$CA_{ij} = \frac{1}{T} \sum_{q=1}^{T} I(\lambda_i^{(q)} = \lambda_j^{(q)}),$$

I(prop) is 1 if the proposition *prop* is true and is 0 if it is false.

Weighted Co-association

When we assign weights to each partition in P, we define weighted co-association as

$$CA_{ij} = \frac{1}{T} \sum_{q=1}^{T} W_q I(\lambda_i^{(q)} = \lambda_j^{(q)}),$$

Where W_q represents the weight of a partition.

Weights

The confidence scores are normalized over the ensemble of partitions to determine the weights of partitions.

$$W_q = \frac{confi(P^q)}{\sum_q confi(P^q)}$$

Here $confi(p^q)$ represents the confidence score of a partition.

Confidence score of a partition

We calculate a confidence score from the intra-cluster similarity and inter-cluster dissimilarity of the partitions. We define *confidence score* of a partition P^q as:

$$confi(P^q) = \left(\frac{1}{k} \sum_{j=1}^{k} \frac{1}{|C_j|} \sum_{O_i \epsilon C_j} \parallel O_i - m_j \parallel^2 \right) - \left(\frac{2}{k(k-1)} \sum_{j=1}^{k} \sum_{i=1}^{k} \parallel m_i - m_j \parallel^2 \right)$$

The first term above quantifies the average intra-cluster similarity within the partition as it takes the sum of squared distance between the centroid and all points in the cluster. The second term quantifies the average inter-cluster similarity between the adjacent clusters for the given partition as it computes the distance between their centroids. By negating the second term, the confidence score thus combines each cluster's average similarity and the dissimilarity between adjacent clusters.

From weighted co-association for all the pairs of objects, we get a binary matrix *M* by thresholding the values with a threshold θ. Thus we define elements of $n \times n$ matrix *M* as follows.

$$M_{ij} = \begin{cases} 1 & \text{if } CA_{ij} \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

4. CROSS ASSOCIATION

Cross-association of a binary matrix is to partition its rows and columns into disjoint row-groups and column-groups such that the submatrices obtained by taking individual row-groups and column-groups are homogenous. Since it is not possible to get all submatrices to be totally homogenous, we aim at maximizing some measure of homogeneity across all submatrices. When the input matrix is symmetric, the row-groups and column-groups correspond to the same partition. Chakrabarti et al [5] present a method of finding cross-association based on Minimum Description Language (MDL) principle which seeks to minimize the compression cost.

Let us assume that the rows of *M* are partitioned into row-groups $R_1, R_2, \ldots, R_k$ and since *M* is symmetric, the columns also are partitioned into same groups. The objective is to determine partitions such that the submatrices *M'(i, j)* defined by row-groups R_i and column-groups $R_j, \forall i, j$ are as homogenous as possible. We measure the homogeneity in the following way.

For any submatrix *M'* of *M*, let n_0, n_1 denote the number of 0s and 1s, respectively in *M'*. The measure of non-homogeneity, *cost(M')*, is defined as

$$cost(M') = -\left(n_0 \ln \frac{n_0}{n_0 + n_1} + n_1 \ln \frac{n_1}{n_0 + n_1} \right)$$

The aggregated cost of the partition $R = (R_1, R_2, \ldots, R_k)$ is given by $\sum_{ij} cost(M'(i, j))$. The aim is to determine a partition of rows *M* such that this cost is minimized. The minimization problem is hard to solve and we give here a practical algorithm, due to [5], to get a local minimum.

We can start with any arbitrary partition, the same partition for both rows and columns. In each iteration, for each individual row, the least cost row-group is determined. If assignment of all rows to the respective row-groups yields a lower overall cost then the algorithm makes this assignment and proceeds to the next iteration. The iterative process terminates when it is not possible to decrease the overall cost by reassignment. The computational details are given as follows.

Let us assume that the current row partition (as well as column-partition) is $R = \{R_1, R_2, \ldots, R_k\}$. R_{jm} denotes the submatrix corresponding row-group R_j and column-group R_m. Let $n_0(j, m)$ and $n_1(j, m)$ be the number of 0's and 1's, respectively in R_{jm}, $1 \leq j \leq k$ and $1 \leq m \leq k$. Every row *r* of *M* is partitioned into *k* parts as $r_1, r_2, \ldots, r_k$ and $n_0(r_i)$, $n_1(r_i)$ are number of 0's and 1's, respectively in r_i. We compute the cost of assigning r_i to row-group R_j using the following formula.

$$cost(r, R_j) = -\left(\sum_{m=1}^{k} n_0(r_m) \ln \left(\frac{n_0(j, m)}{n_0(j, m) + n_1(j, m)} \right) + n_1(r_m) \ln \left(\frac{n_1(j, m)}{n_0(j, m) + n_1(j, m)} \right) \right)$$

The row-group corresponding to the smallest cost is the candidate row-group to which *r* should be included. Thus row *r* is assigned to row-group R_p if $P = argmin_j cost(r, R_j)$. In this manner, we determine the appropriate row-group for each row. The new assignment yields a different overall cost $\sum_{ij} cost(M'(i, j))$. If the cost decreases due to the new assignment then we carry out the assignment and proceed to the next iteration. It is shown by Chakrabarti et al [5], that the overall cost never increases in this process. The algorithm yields *k* diagonal blocks with all most all 1 entries. The *k* row-groups correspond to *k*-cluster as consensus clustering.

5 EVALUATION CRITERIA

In general, there are two approaches to evaluate consensus clustering, one is external evaluation criteria and other is relative consensus criteria. We compute the external consensus criteria, that compares the resulting consensus clustering with the external true label of data points.

External Consensus Criteria

External evaluation criteria measure the similarity/diversity between consensus clustering and the ground truth clustering. There are several similarity measures that compare a pair of partitions. These include Rand Index [16], Jaccard Index [3], Adjusted Rand Index (ARI) [11, 17], Wallace Index [21], and Normalized Mutual Information [8, 18]. In our approach, in line with other existing methods, we use adjusted Rand Index and Normalized mutual Information as the similarity measure between two partitions.

The Adjusted Rand Index[11, 17]

Adjusted Rand Index, an important variant of Rand Index, corrects the lack of invariance when partitions are selected at random. let us first define the following quantities.

$$t_0 = \sum_{r,s} \binom{N_{rs}}{2}, t_1 = \sum_{r=1}^{ki} \binom{n_{r(i)}}{2}$$

$$t_2 = \sum_{s=1}^{kj} \binom{n'_{s(j)}}{2}$$

$$t3 = \frac{t_1 \times t_2}{\binom{n}{2}}$$

Where N_{rs} is the number of common data items in r^{th} cluster of partition P^i and s^{th} cluster of partition P^j, that is in $C_r \bigcap C'_s$. Let $n_{r(i)}$ and $n'_{s(j)}$ be number of items in C_r of P^i and C'_s of P^j, respectively. The Adjusted Rand Index is defined as [11]

$$ARI(P^i, P^j) = \frac{t_0 - t_3}{\frac{1}{2}(t_1 + t_2) - t_3}$$

For two identical clusterings, the value of the index is 1 and if two clusterings are in total disagreement then the value is 0.

Normalized Mutual Information[18]

The Mutual Information computes the mutual information or transformation between ground truth label and the resulting consensus cluster label normalized by geometric mean of entropies. Mutual information, which is a symmetric measure to quantify the statistical information shared between two distributions (Cover and Thomas, 1991), provides a sound indication of the shared information between a pair of clusterings.

Let *X* and *Y* be the random variables described by the cluster labeling P^a and P^b, with k^a and k^b groups respectively. Let *I(X, Y)* denote the mutual information between *X* and *Y* , and *H(X)* denote the entropy of *X*. One can show that *I(X, Y)* is a metric. There is no upper bound for *I(X, Y)*,so for easier interpretation and comparisons, a normalized version of *I(X, Y)* that ranges from 0 to 1 is desirable. Several normalizations are possible based on the observation that *I(X, Y)*$\leq$ min(*H(X)*, *H(Y)*). These include normalizing using the arithmetic or geometric mean of *H(X)* and *H(Y)*. Since $H(X) = I(X, X)$, we prefer the geometric mean because of the analogy with a normalized inner product in Hilbert space. Thus the normalized mutual information (NMI) used is:

$$NMI(X, Y) = \frac{I(X, Y)}{\sqrt{H(X)H(Y)}}$$

6 EXPERIMENTAL RESULTS

We use both artificial and real-world benchmark datasets from UCI ML repository such as Iris, Glass, Wine and Ecoli [22]. It should be noted that all the datasets are labeled and contain supervised class information. Table 1 gives the basic information about these datasets.

Ensemble Generation

To build our ensemble, we use *k*-means algorithm as our basic clustering techniques because it is one of the most widely used clustering algorithm and has been used in many previous cluster ensemble studies. The ensemble of partitions is obtained by applying *k*-means to the same data with different initializations. The value of *k* is considered as specified in the classlabel of each dataset in Table 1. We generate ensembles of sizes 100 and also consider first 30 and 50 ensembles individually for all datasets. If any duplicate clusterings are found from each of ensemble, then those duplicates are removed. This means we only consider

Tabela 1: Description of the Datasets, where 'n' denotes number of samples, 'd' denotes number of features/dimension and 'k' denotes the number of class label

Datasets	Samples (n)	Features (d)	Class (K)
Iris	150	4	3
Zoo	101	18	7
Soyabean	47	35	4
Glass	214	9	6
Wine	178	13	3
E-coli	336	7	8
Satlogimage	2310	19	7
Chart	600	60	6

the distinct partitions in an ensemble.

Value of θ

We construct the binary matrix M using the weighted co-association and thresholding with θ. The choice of the value of θ is very crucial and it is observe that the quality of consensus clustering is influenced seriously by the choice of θ. We experimented with several values of θ, and found that θ in the range of [0.5, 0.7] yields best results for all datasets. It is also observe that in this range the quality of the consensus clustering remains constant. Infact an interesting observation of our experiment with different values of θ is that the external measure of the consensus clustering gradually increases with the increase the value of θ from 0 and remains constant with in a range, mostly within [0.5, 0.7] and than decreases when θ approaches 1.

We compare the average performance of existing consensus clustering methods, namely CSPA, HGPA and MCLA with our methods for all three types of ensembles for different datasets. We report the experimental results in following Tables. In each experiment, for a dataset we run all the algorithms to get the consensus clustering and compute the ARI and NMI between the resulting clustering and the ground truth partition. For both the measures the best value is 1 and hence, closer the value to 1 better is the consensus.

Tabela 2: Comparison Results for IRIS

partitions	Threshold		CSPA	HGPA	MCLA	proposed
76	0.5	ARI	0.70	0.46	0.72	0.72
		NMI	0.7139	0.5341	0.7419	0.7481
31	0.5	ARI	0.70	0.43	0.73	0.72
		NMI	0.7139	0.5031	0.7582	0.7481
15	0.5	ARI	0.70	0.44	0.72	0.72
		NMI	0.7139	0.5117	0.7419	0.7481

IRIS dataset (Table 2) is one of well-behaved dataset and most of the algorithms give a consensus with Adjusted Rand Index 0.72. So does the proposed method. If we view the consensus clustering with respect to Mutual Information, our method gives better result for all the three ensemble. Table 3 shows the comparison for ZOO dataset, our method outperforms both in terms of Adjusted Rand Index and Mutual Information measure for 30 cluster ensemble. In case of 50 cluster ensemble our method outperforms only in terms of Adjusted Rand Index. For 100 cluster ensemble MCLA performs better. It is interesting to note that

Tabela 3: Comparison Results for ZOO

partitions	Threshold		CSPA	HGPA	MCLA	proposed
100	0.6	ARI	0.38	0.43	0.69	0.68
		NMI	0.6263	0.6412	0.7749	0.7655
50	0.6	ARI	0.43	0.39	0.54	0.63
		NMI	0.6606	0.6211	0.7555	0.7198
30	0.6	ARI	0.40	0.47	0.54	0.59
		NMI	0.6350	0.6824	0.6779	0.7408

for GLASS dataset (Table 4), the proposed algorithm outperforms all other algorithms in term of ARI for 30 cluster ensemble. Mutual Information measure also outperforms for both the 50 and 30 cluster ensemble. But for 100 cluster ensemble MCLA performs better in both Adjusted Rand Index and Mutual Information.

For WINE dataset (Table 5), we observe another interesting feature of our method. For 100 cluster ensemble it performs better than other algorithms. In case of 50 cluster ensemble CSPA performs much better in terms of Adjusted Rand Index and

Tabela 4: Comparison Results for GLASS

partitions	Threshold		CSPA	HGPA	MCLA	proposed
100	0.7	ARI	0.18	0.16	0.22	0.20
		NMI	0.3255	0.3275	0.3595	0.3578
50	0.7	ARI	0.12	0.15	0.22	0.22
		NMI	0.2448	0.2353	0.3540	0.3577
30	0.7	ARI	0.16	0.16	0.16	0.37
		NMI	0.2912	0.2290	0.3221	0.3715

Tabela 5: Comparison Results for WINE

partitions	Threshold		CSPA	HGPA	MCLA	proposed
96	0.5	ARI	0.41	0.33	0.40	0.42
		NMI	0.4006	0.3296	0.4375	0.4309
47	0.5	ARI	0.81	0.67	0.72	0.37
		NMI	0.7961	0.6794	0.6915	0.4288
29	0.5	ARI	0.41	0.31	0.39	0.37
		NMI	0.4049	0.3516	0.4372	0.4288

Mutual Information. For 30 cluster ensemble, the Mutual information of MCLA is high than other. We could not find any specific reason for this deviation. For SOYABEAN (Table 6) dataset performance of our algorithm is not good in comparison to all three algorithms.

Tabela 6: Comparison Results for SOYABEAN

partitions	Threshold		CSPA	HGPA	MCLA	proposed
97	0.5	ARI	0.51	0.55	0.56	0.44
		NMI	0.6381	0.7158	0.7225	0.6527
48	0.5	ARI	0.52	0.55	0.55	0.52
		NMI	0.6629	0.7158	0.7180	0.6960
29	0.5	ARI	0.53	0.55	0.60	0.50
		NMI	0.6427	0.7158	0.7481	0.7093

Tabela 7: Comparison Results for ECOLI

partitions	Threshold		CSPA	HGPA	MCLA	proposed
100	0.6	ARI	0.31	0.30	0.38	0.46
		NMI	0.5353	0.5068	0.5379	0.5976
50	0.6	ARI	0.31	0.30	0.48	0.29
		NMI	0.5209	0.5102	0.5917	0.5335
30	0.6	ARI	0.30	0.26	0.37	0.44
		NMI	0.5250	0.4498	0.5528	0.5776

In case of ECOLI (Table 7) dataset for 100 and 30 cluster ensembles our algorithm outperforms in terms of Adjusted Rand Index and Mutual Information measure than other algorithms. For STLOGIMAGE (Table 8) dataset our method performs better in all respect of performance measure for all the three cluster ensembles. The performance measure of CSPA is better than the HGPA and MCLA performance measure.

Tabela 8: Comparison Results for SATLOGIMAGE

partitions	Threshold		CSPA	HGPA	MCLA	proposed
100	0.6	ARI	0.52	0.33	0.43	0.55
		NMI	0.6281	0.4712	0.5402	0.6738
50	0.6	ARI	0.42	0.34	0.38	0.45
		NMI	0.5257	0.4061	0.4522	0.5573
30	0.6	ARI	0.35	0.20	0.27	0.37
		NMI	0.4856	0.3873	0.4341	0.5007

For CHART dataset (Table 9) our algorithm perform best for 100 and 30 ensembles for both the ARI and MI. But for 50 ensembles MCLA performs better. For most of the instances the proposed algorithm is better than other methods.

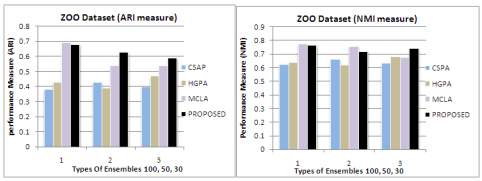

Figura 1: Performance Comparison of ZOO Dataset

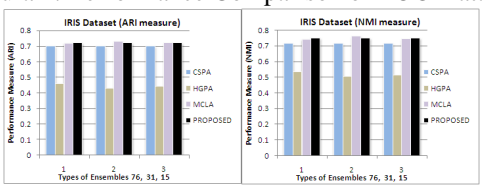

Figura 2: Performance Comparison of IRIS Dataset

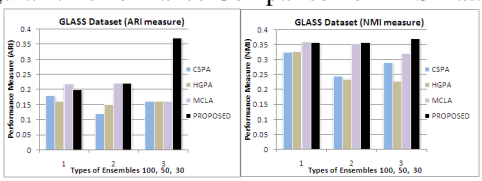

Figura 3: Performance Comparison of GLASS Dataset

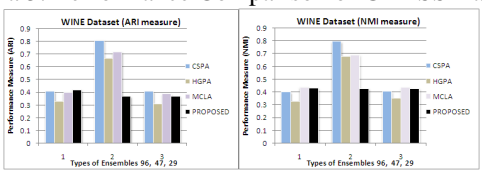

Figura 4: Performance Comparison of WINE Dataset

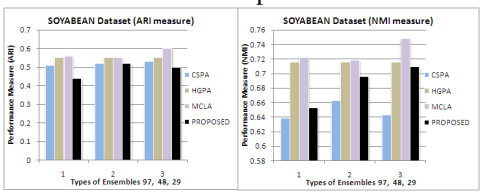

Figura 5: Performance Comparison of SOYABEAN Dataset

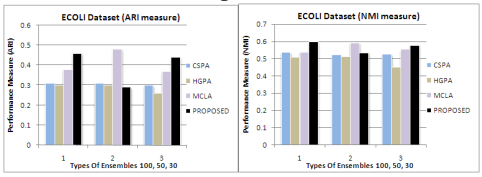

Figura 6: Performance Comparison of ECOLI dataset

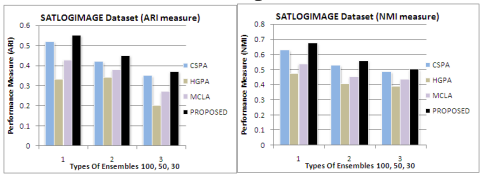

Figura 7: Performance Comparison of SATLOGIMAGE Dataset

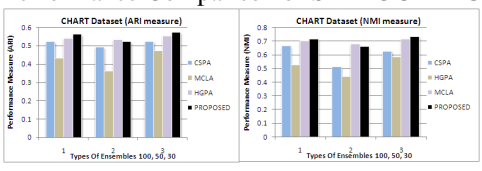

Figura 8: Performance Comparison of CHART Dataset

Tabela 9: Comparison Results for CHART

partitions	Threshold		CSPA	HGPA	MCLA	proposed
100	0.6	ARI	0.52	0.43	0.54	0.56
		NMI	0.6633	0.5232	0.7035	0.7143
50	0.6	ARI	0.49	0.36	0.53	0.52
		NMI	0.5123	0.4397	0.6742	0.6598
30	0.6	ARI	0.52	0.47	0.55	0.57
		NMI	0.6234	0.5824	0.7139	0.7298

7. CONCLUSION

The consensus partition generated by the proposed algorithm is likely to be appeared as the optimal solution for the ensemble of partitions. By assigning the priorities to each partition, influences of partitions on the resulting consensus can be controlled, it means presence of noisy partitions will not affect the quality of consensus drastically. The transitive qualitative co-association between the data items is obtained for any large dataset by using the MDL based simultaneous grouping (cross-association) method. The computational experiments clearly shows that the algorithm proposed above gives better results than the heuristics [18] on the benchmark data for consensus clustering problem.

REFERENCES

[1] M. Al-Razgan, and C. Domeniconi, Weighted Clustering Ensembles, Proceedings of the Sixth SIAM International Conference on Data Mining, Bethesda, Maryland April 20-22, 2006.

[2] H. G. Ayad, M. S. Kamel. Cumulative voting consensus method for partitions with varaiables number of clusters. IEEE trans on Patteren Analysis and Machine Intelligence, Vol.30, No.1, pp. 160-173, 2008.

[3] A. Ben-Hur, A. Elisseeff and A. Guyon. A stability based method for discovering structure in clustered data. In Proc. Pacific Symposium on Biocomputing, pages 6-17,2002.

[4] R. Caruana, A. Niculescu-Mizil, G. Crew, A. Ksikes, Ensemble selection from libraries of models, Proceedings of the twenty-first international conference on Machine Learning, Banff, Alberta, Canada, pp.18, July 04-08, 2004.

[5] D. Chakrabarti, S. Papadimitriou, D. S. Modha and C. Faloutsos, Fully Automatic Cross Associations, The Tenth ACM SIGKDD. Int. conf on Knowledge Discovery and Data Mining (KDD 04), Seattle, Washington, August 22-25,2004.

[6] R. O. Duda, P. E. Hart, D.G. Stork, Patteren Classification, Wiley, New York, 2002.

[7] A. Fred and A. K. Jain. Data clustering using evidence accumulation. In Proc. ICPR (4), pp. 276-280, 2002.

[8] A. Fred and A. K. Jain. Robust data clustering. In Proc. IEEE Computer Society Conference on Computer Vision and Patteren Recognition, CVPR, USA, 2003.

[9] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. 21st International Conference on Data Engineering (ICDE'05), pages 341-352, 2005.

[10] A. Goder and V. Filkov. Consensus clustering algorithms: Comparison and Refinement. Proceedings of ALENEX, pp. 109-117, 2008.

[11] L. J. Hubert and P. Arabie. Comparing partitions. Journal of Classification, 2, 193-218. 1985

[12] L. I. Kuncheva, S. T. Hadjitodorov and L. P. Todorova. Experimental comparison of cluster ensemble methods. Proc FUSION 2006, pp. 231-255, Florence, Italy, 2006.

[13] T. Li, M. Ogihara and S. Ma. On combining multiple clusterings. In proc. CIKM'04, pp. 123-146, 2004.

[14] T. Li, C. Ding and M. I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In Proceedings of the 2007 IEEE International Conference on Data Mining (ICDM 2007), pages 577-582, 2007.

[15] N. Nguyen and R. Caruana. Consensus Clusterings. Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), pages 607-612.

[16] W. M. Rand, Objective Criteria for the evaluation of clustering methods, Journal of the American Statistical Association 66, pp. 846-850, 1971.

[17] D. Steinley. Properties of the Hubert-Arabie adjusted Rand index. Psychol Methods, 9: 386-396, 2004.

[18] A. Strehl and J. Ghosh, Cluster ensembles - A knowledge reuse framework for combining multiple partitions. In proceedings of AAAI, pages 93-98, 2002.

[19] A. Topchy, A. K. Jain and W.Punch. A mixture model for clustering ensembles. In proc. SIAM Conference on Data Mining, pages 379-390, 2004.

[20] A. Topchy, M. Law, A. K. Jain and A. Fred. Ananlysis of consensus partition in cluster ensemble. IEEE International Conference on Data Mining, ICDM, pp. 225-232, 2004.

[21] D. L. Wallace. Comment. Journal of American Statistical Association. 78: 569-576, 1983.

[22] UCI Machine Learning Repository (Website: http:// archive. ics. uci. edu/ ml).