Weighted Cluster Ensemble Using a Kernel Consensus Function

Sandro Vega-Pons, Jyrko Correa-Morris, and José Ruiz-Shulcloper

Advanced Technologies Application Center, Havana, Cuba
{svega, jmorris, jshulcloper}@cenatav.co.cu

Abstract. Cluster ensemble is a good alternative to face the problem of data clustering. Some studies based on mathematical models have shown that cluster ensemble methods lead to an effective improvement of the results of the standard clustering algorithms. In this paper, we focus on this problem, proposing a new approach to solve it, by adding a new step into the usual cluster ensemble methodology. Representing partitions by graphs and a new kernel function to measure the similarity between partitions are other proposals for this work. Experiments with synthetic and real databases show the suitability and effectiveness of our method.

Keywords: cluster ensemble, graph kernel, consensus function.

1 Introduction

Data clustering is an essential technique on any field of investigation which involves analysis or processing of multivariate data, such as, data mining, document retrieval, image segmentation, pattern classification, etc. Its goal is to find underlying structuring of a data set [1]. A large variety of clustering algorithms that have been proposed; the c-Means algorithm, EM, those based on graphs theory and Mean Shift are some examples (see a summary in [2]). However, as is known, there is no clustering method capable of working correctly for all data structure. These methods impose an organization to the data, but in some occasions this structure can be far from the best one, due to the different cluster sizes, densities and shapes that can be present in a set of patterns. With the goal of improving the clustering results, and based on the successes of the combination of supervised classifiers, the idea of clustering ensemble [3] emerges, which is the combination of the results of the clustering algorithms. This approach combines the information of a partition set through a consensus function, to obtain a more representative partition.

A common difficulty present in the clustering ensemble process is that some partitions are very different from the rest of partitions and they represent noise in the consensus step. For this reason we propose in the section 4.2 a heuristic to estimate the importance of each partition, which allows a better use of information in the clustering ensemble. Also in section 4.3 is introduced a new consensus function based on the definition of a new similarity measure for partitions using a kernel function. Finally, in section 4.4 is given a brief overview of our method.

J. Ruiz-Shulcloper and W.G. Kropatsch (Eds.): CIARP 2008, LNCS 5197, pp. 195–202, 2008.

2 Related Works

Different clustering ensemble methods have been proposed in the literature [1]. The consensus by co-association [3], is based on the computation of the co-association matrix C, where C_{ij} represents how many times the objects x_i and x_j are in the same cluster in the cluster ensemble, the final partition is obtained by the single-link algorithm using C as a new similarity measure of the data set.

When solving the cluster ensemble problem as an optimization problem, some heuristics have been proposed in the literature: CSPA, HGPA and MCLA based on graph partitioning [4]. These methods represent the cluster ensemble by a hyper-graph, where each cluster of each partition is represented by a hyper-edge.

Another method is to find the partition $\mathcal{P}^*$ which maximizes $\sum_{k=1}^{s} U(\mathcal{P}^*, \mathcal{P}_k)/s$ [5], where $U(\mathcal{P}^*, \mathcal{P}_k)$ is a function based on mutual information [6], maximizing the quadratic mutual information (QMI) between the partitions on the clustering ensemble and the consensus partition. Also [5] proposes a probabilistic model of consensus using a finite mixture of multinomial distributions in a space of clustering and the consensus partition is found using the EM algorithm.

In order to make a rigorous analysis of the cluster ensemble methods, several theoretical results based on a probabilistic model have been introduced in [7]. Recently, some comparative studies about different methods of clustering ensemble have been realized. A study of 24 algorithms, combining the generation and the consensus process, over 24 data bases was realized [8]. The implementation of some heuristics to solve this problem and the comparison of the performance taking into account efficiency and accuracy was presented on [9].

3 Our Problem Formulation

So far, the cluster ensemble methods are based on two main steps: ensemble generation and consensus function [3]. During the ensemble generation process, noisy partitions may appear, and they will be treated as the others by the consensus process. For this reason, the consensus partition may not be as good as possible, and it can even be worst than some partitions in the cluster ensemble. Therefore, we consider that an intermediate step should exist between the generation and consensus process, which estimates the importance of every partition in the ensemble and assigns a proportional weight to each one. We consider three fundamental steps in our method: ensemble generation, qualitative analysis of the cluster ensemble and the consensus process.

4 Proposed Method

4.1 Generation of the Clustering Ensemble

Given a set of objects $\chi = \{x_1, x_2, \ldots, x_n\}$, a clustering ensemble is a set $\mathbb{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_m\}$, where $\mathcal{P}_i$ is a partition of χ, for $i = 1, 2, \ldots, m$. The partitions of $\mathbb{P}$ can be obtained using different representations of the elements of χ, different

clustering algorithms or the same algorithm with different initialization values for its parameters. The number of clusters in each partition $\mathcal{P}_i$ does not have to be necessarily the same.

4.2 Qualitative Analysis of the Clustering Ensemble

It is the evaluation of the partitions of $\mathbb{P}$, taking into account a set of relevant properties and the way that these properties are satisfied by each partition. The properties selection can be conditioned by the problem. Some of them could be compactness, separability, shape of clusters, etc.

Properties Validation Indexes

In order to measure the selected properties, we propose to use a set of indexes $\mathbb{I} = \{I_1, I_2, \dots, I_k\}$, where each index represents a different property. Formally we can define an index as a function $I : \mathbb{P}_\chi \to [0,1]$ where $\mathbb{P}_\chi$ is the set of all possible partitions with the elements of χ. For each $\mathcal{P} \in \mathbb{P}_\chi$, $I(\mathcal{P})$ is defined as the degree of accomplishment of the property represented by I evaluated on $\mathcal{P}$ and for all $\mathcal{P}, \mathcal{P}' \in \mathbb{P}_\chi$, if $I(\mathcal{P}) > I(\mathcal{P}')$ implies that the partition $\mathcal{P}$ satisfies the property more than $\mathcal{P}'$. Using this set of indexes, we can create the set $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$, where λ_i is the weight associated to $\mathcal{P}_i$.

Determining the Set of Weights

For each index $I_j \in \mathbb{I}$, we compute $S_j = \sum_{i=1}^{m} I_j(\mathcal{P}_i)$, and we define the function $\varphi_j : \mathbb{P}_\chi \to [0,1]$ such that $\varphi_j(\mathcal{P}) = \frac{I_j}{S_j}$ so $\sum_{i=1}^{m} \varphi_j(\mathcal{P}_i) = 1,\ \forall j = 1, \dots, k$. Therefore, each φ_j can be related to the distribution function of certain discrete random variable Y_j. Then we define:

$$H(I_j) = H(Y_j) = -\sum \varphi_j(\mathcal{P}_i) log\left(\varphi_j(\mathcal{P}_i)\right),$$

where $H(Y_j)$ is the entropy of Y_j [6]. Using the entropy properties, we have $H(I_j) \geq 0$, $H(I_j) \leq log|\mathbb{P}|$ and $H(I_j)$ reaches the maximum value when $\varphi_j(\mathcal{P}_1) = \varphi_j(\mathcal{P}_2) = \dots = \varphi_j(\mathcal{P}_m)$. Using the continuity property of $H(Y_j)$, we can conclude that the higher values of $H(Y_j)$ the stronger likeness between the $I_j(\mathcal{P}_i)$ values. Therefore, $H(I_j)$ is a good measure of the representativeness of the property related to the index I_j.

Finally, to each partition $\mathcal{P}_i$ we assign the weight λ_i, which is given by:

$$\lambda_i = \sum_{j=1}^{k} H(I_j)\left(1 - \left|I_j(\mathcal{P}_i) - \frac{1}{m} S_j\right|\right).$$

The second factor of this expression is an evaluation of $I_j(\mathcal{P}_i)$, based on the distance of this value to the mean value $\frac{1}{m} S_j$.

4.3 Consensus Function

It is a function, which maps the given set of partitions $\mathbb{P}$ to a single consensus partition $\mathcal{P}^*$. In the consensus process we are looking for a partition, which consolidates the partitions in the clustering ensemble. Using the weight λ_i associated to each partition $\mathcal{P}_i$, we define $\mathcal{P}^*$ as:

$$\mathcal{P}^* = arg \max_{\mathcal{P} \in \mathbb{P}_x} \sum_{i=1}^{m} \lambda_i \Gamma(\mathcal{P}, \mathcal{P}_i) ,$$

where $\mathcal{P}$ goes through all possible partitions of the data χ, $\Gamma(\mathcal{P}_i, \mathcal{P})$ is a similarity measure between two partitions. Our consensus function is based on the definition of a new kernel between partitions, representing each partition by a graph with a specific structure.

Representing Partitions by Graphs

For each partition $\mathcal{P}$, we define a graph $G_{\mathcal{P}} = (V, E)$, where $V = \{v_{x_1}, v_{x_2}, \dots, v_{x_n}\}$ and v_{x_i} is the node associated to x_i. On this kind of graphs there is an edge between the nodes v_{x_i}, v_{x_j} if and only if the objects x_i and x_j are in the same cluster on $\mathcal{P}$. This way, all graphs obtained from partitions of $\mathbb{P}_\chi$ are graphs in which each connected component is a complete graph.

Let $\mathcal{G}: \mathbb{P}_\chi \rightarrow \mathbb{G}_\chi$ be a map from $\mathbb{P}_\chi$ into the space $\mathbb{G}_\chi$ (the space of all graphs obtained by a partition of the objects of χ), where to each partition corresponds its graph representation. It is not difficult to prove that $\mathcal{G}$ is a bijective function. For this reason, to work with graphs in $\mathbb{G}_\chi$ is equivalent to work with partitions in $\mathbb{P}_\chi$.

Similarity Measure

In this section we propose a new similarity measure between partitions by the introduction of a new kernel for graphs. Our kernel is based in the comparison of all path existent on the graphs (path kernel). Due to the structure of our graphs, it is possible to compute very fast all paths in a graph (a not computable in polynomial time problem for the general case).

Let $\Sigma(G)$ denotes the set of all possible paths on a graph $G = (V, E)$, $h \in V^l$ is a sequence of nodes $h_1 h_2 \dots h_l$, where there is an edge between every consecutive pair of nodes (h_i, h_{i+1}) and $\forall i \neq j, h_i \neq h_j$; l is the path length.

Based on the Marginalized Kernel [10] we define k as:

$$k(G, G') = \sum_{h \in \Sigma(G)} \sum_{h' \in \Sigma(G')} k_\delta(h = h') p(h/G) p(h'/G') ,$$

where $k_\delta(h = h')$ is the matching kernel, taking value 1 if $h = h'$ and 0 otherwise and $p(h/G)$ is given by:

$$p(h/G) = p_s(h_1) \prod_{i=2}^{l} p_t(h_i/h_{i-1}) p_e(h_l) ,$$

being $p_s(h_1)$ the probability to start at the vertex h_1 as well as the probability of moving to h_i being on h_{i-1} is $p_t(h_i/h_{i-1})$ and $p_e(h_l)$ is the probability of ending on h_l. In our case $p(h/G) = \frac{1}{n|C_h|^{l-1}}$, where C_h is the connected component, which contains h, $p_s(h_1) = \frac{1}{n}$, $p_t(h_i/h_{i-1}) = \frac{1}{|C_h|}$, and $p_e(h_l) = \frac{1}{|C_h|}$.

Proposition 1: *The function k is a positive semi-definite kernel.*

Proof: Let $\Omega(\mathbb{G}_\chi)$ be the set of all possible paths that can exist in the graphs of $\mathbb{G}_\chi$. Let us assign to each graph $G \in \mathbb{G}_\chi$ the vector $X_G \in \mathbb{R}^{|\Omega(\mathbb{G}_\chi)|}$, whose coordinates are $\delta(h/G)p(h/G)$ for each $h \in \Omega(\mathbb{G}_\chi)$, where $\delta(h/G)$ is 1 if $h \in \Sigma(G)$, and 0 otherwise. It is not difficult to verify that, for two graphs G and G', $k(G, G') = \langle X_G, X_{G'} \rangle$ and therefore the quadratic form $\sum_{i,j} \alpha_i \alpha_j k(G_i, G_j)$ in $\alpha_1, \alpha_2, \ldots, \alpha_s \in \mathbb{R}$ is positive semi-definite for all $G_1, G_2, \ldots, G_s \in \mathbb{G}_\chi$ and every positive integer number s.

This kernel is expressive enough as a similarity measure between partitions because it takes into account the subsets belonging to the same cluster in both partitions, and measures the relationship between these subsets and the cluster to which it belongs in each partition respectively. We can implement this kernel function very efficiently, because we do not need to search every path in the graph, since all paths with the same length in the same connected component have equal values of $p(h/G)$, and the number of paths with length l in the connected component C_h is the variations of $|C_h|$ in l.

Determining the Consensus Partition

For the kernel k defined in the previous section exists a Hilbert space $\mathcal{H}$ and a function ϕ: $\mathbb{G}_\chi \to \mathcal{H}$ such that $\forall\, G, G' \in \mathbb{G}_\chi$, $k(G, G') = \langle \phi(G), \phi(G') \rangle_{\mathcal{H}}$. Here we can consider $\mathcal{H}$ as the Reproducing Kernel Hilbert Space associated with k.

We are looking for the partition $\mathcal{P}^*$, represented by the graph G^* such that:

$$G^* = arg \max_G \sum_{i=1}^{m} w_i k(G_i, G)\,. \tag{1}$$

Theorem 1: *The optimization problem (1) in the Reproducing Kernel Hilbert Space associated with k, has a unique solution ψ, which admits a representation of the form:* $\psi = \sum_{i=1}^{m} \beta_i \phi(G_i)$ *with* $\beta_i = \frac{\lambda_i}{\sum_{j=1}^{m} \lambda_j}$.

The previous theorem assures that $G^* = \phi^{-1}(\psi)$, which implies to solve the pre-image problem. This problem consists in finding the graph in the input space given the solution in the feature space. To get an exact solution for this problem is a very difficult task. That is why we choose as an approximate solution:

$$G^* = arg \min_{G \in \mathbb{G}_\chi} \|\psi - \phi(G)\|^2\,,$$

with $\|\psi - \phi(G)\|^2 = k(G, G) - 2\sum_{i=1}^{m} \beta_i k(G, G_i) + \sum_{i,j=1}^{m} \beta_i \beta_j k(G_i, G_j)$.

To solve this problem we follow the efficient sample strategy introduced in [11], which in our case is simplified due to the structure of our graphs. This strategy is based on a stochastic search approach to determine the adjacency matrix of the solution graph, which represents the consensus partition.

4.4 Steps of the Proposed Method

- Apply different clustering algorithms (or the same with different parameters initializations) on a given set of objects and generate the ensemble with their outcomes.
- Define a set of properties indexes to evaluate the way that these properties are satisfied by each partition in the cluster ensemble.
- Compute an associated weight to each partition using the indexes defined previously.
- Represent each partition by a graph using the proposed graph kernel as a similarity measure.
- Obtain the representation of the consensus partition on the feature space.
- Solve the pre-image problem to find the consensus partition.

5 Experimental Results

The experiments were performed using several databases. First we apply our algorithm over a synthetic data base: *Half-Rings* conformed by two classes, each one with 150 elements Fig 1 (a). We also use four real databases: *Iris Data* (three clusters) available at the UCI Machine Learning Repository, and three WEKA's ARFF databases: *Drug1n* (five clusters), *Weather* (two clusters) and *Labor* (two clusters). In all experiments T represents the number of partition on the clustering ensemble and c is the number of classes used for the clustering algorithms to generate the ensemble.

For the experiments we define four simple property indexes that measure the properties: Inter-cluster distance, Intra-cluster distance, mean size of clusters, difference between the clusters size. We use only these four indexes, but in a general way, this set of indexes could be defined using any type of indexes, which evaluate some properties over the dataset.

Some results of the experiment are showed in Fig 1. The first image is the original data set and the remainders are 5 different partitions of this data set. In Table 1, λ is the weight associated with each partition and the value $\|\phi - \psi\|^2$ is the distance between each partition to the solution of the problem in the feature space. This experiment shows the good performance and the robustness of the algorithm, because partitions like (f), which represents noise in comparison with the rest of cluster ensemble partitions, take lower values of λ and higher values of $\|\phi - \psi\|^2$. It implies that this partition will have the minimum influence over the result.

Table 2 is a comparison of our method with other methods of cluster ensemble EMC, QMI [5] and CSPA, HGPA, MCLA [4]; our method is denoted by WKF. On this experiment we use c-Means with random centroids initialization to generate the cluster ensemble. In [3] was realized a similar experiment using $T = 50$ and the

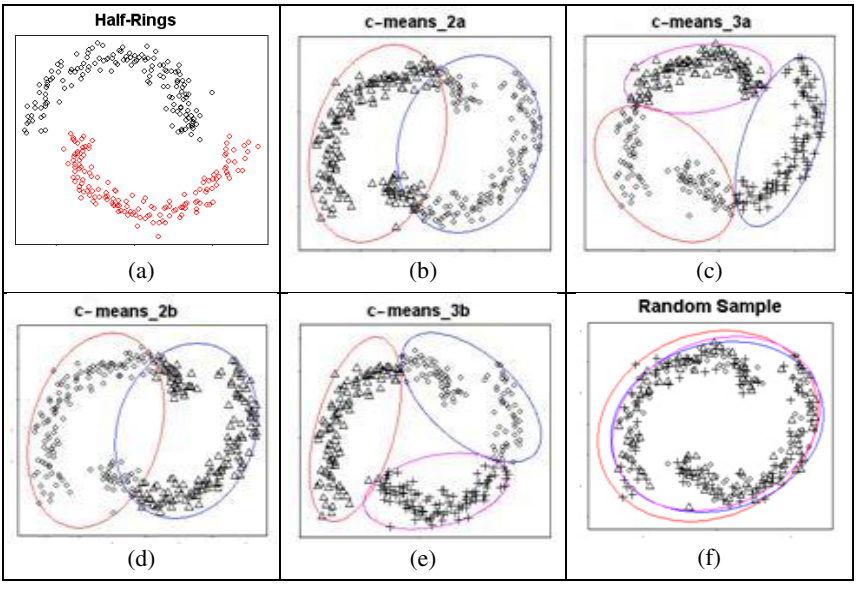

Fig. 1. (a) Original Database, (b)(d) c-Means with $c = 2$, (c)(e) c-Means with $c = 3$, (f) partition obtained by random distribution of the elements in 3 clusters

Table 1. Results of the experiment showed in Fig 1

	1(a)	**1(b)**	**1(c)**	**1(d)**	**1(e)**
$\boldsymbol{\lambda}$	1.0	0.9469	1.0	0.9737	0.7886
$\|\boldsymbol{\phi} - \boldsymbol{\psi}\|^2$	0.1579	0.4031	0.1579	0.3227	0.7181

Table 2. Clustering error rate (%) for the Iris dataset. The last column shows the distance from the best partition obtained by our method to the solution in the feature space.

T	c	*EMC*	*QMI*	*CSPA*	*HGPA*	*MCLA*	*WKF*	$\|\boldsymbol{\phi} - \boldsymbol{\psi}\|^2$
5	3	11.0	14.7	11.2	41.4	10.9	10.6	0.04172
10	3	10.8	10.8	11.3	38.2	10.9	10.0	0.03980
20	3	10.9	14.5	9.8	39.1	10.9	10.0	0.03980

Table 3. Clustering error rate (%) of c-Means, EM and our algorithm (WKF)

Dataset	*Attributes*	*Instances*	*T*	*c-Means*	*EM*	*WKF*
Drug1n	7	200	10	64.25	64.14	51.5
Iris	4	150	10	25.66	18.19	10.0
Weather	4	14	10	45.99	49.55	35.71
Labor	17	57	10	38.94	29.81	21.05

obtained error rate was 11.1%. Table 3 is a comparison of our method with standards clustering algorithms (c-Means and EM with different parameters initialization).

6 Conclusions

This paper extends previous works on clustering ensembles in several aspects. First, we propose the weighted cluster ensemble approach, which allows obtaining a better consensus partition by estimating the relevance of each partition. Second, we define a new consensus function based on the definition of a new similarity measure between partitions using a graph kernel function. Experimental results show the good performance of our method for several databases and favorable comparison with other consensus function methods.

References

1. Jain, A.K., Law, M.H.C.: Data Clustering: A User's Dilemma. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 1–10. Springer, Heidelberg (2005)
2. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Computing Surveys (CSUR) 31(3), 264–323 (1999)
3. Fred, A., Jain, A.K.: Data Clustering Using Evidence Accumulation. In: Proc. of the 16th Intel Conference on Pattern Recognition, ICPR 2002, pp. 276–280. Quebec City (2002)
4. Strehl, A., Ghosh, J.: Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions. Journal of Machine Learning Research 3, 583–617 (2002)
5. Topchy, A., Jain, A.K., Punch, W.: Clustering Ensembles: Models of Consensus and Weak Partitions. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(12), 1866–1881 (2005)
6. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons, Inc., New York (2006)
7. Topchy, A.P., Law, M.H.C., Jain, A.K., Fred, A.L.: Analysis of Consensus Partition in Cluster Ensemble. In: International Conference on Data Mining, pp. 225–232 (2006)
8. Kuncheva, L.I., Hadjitodorov, S.T., Todorova, L.P.: Experimental Comparison of Cluster Ensemble Methods. In: 9th International Conference on Information Fusion, pp. 1–7 (2006)
9. Gorder, A., Filkov, V.: Consensus Clustering: Comparison and Refinement. In: Proceedings of ALENEX, pp. 109–117 (2008)
10. Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized Kernels Between Labeled Graphs. In: Proc. of the Twentieth Int. Conf. on Machine Learning (2003)
11. Bakir, G.H., Zien, A., Tsuda, K.: Learning to Find Graph Pre-Images. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 253–261. Springer, Heidelberg (2004)