Dissimilarity Increments Distribution in the Evidence Accumulation Clustering Framework

Helena Aidos and Ana Fred

Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa, Portugal
{haidos,afred}@lx.it.pt

Abstract. In this paper, we combine two concepts. The first is the Evidence Accumulation Clustering framework, which uses a voting scheme to combine clustering ensembles and produce a co-association matrix. The second concept are Dissimilarity Increments, which are a high order dissimilarity measure which can identify sparse clusters, since it uses three data points at a time instead of two points, as in Euclidean distance. These two concepts are combined to form a new family of clustering algorithms, where the co-association matrix is used to form a distance which is then used to compute dissimilarity increments. These clustering algorithms are shown to improve the clustering results when compared to the usual Evidence Accumulation Clustering framework.

Keywords: dissimilarity increments distribution, hierarchical clustering, clustering ensembles, co-association matrix.

1 Introduction

Clustering consists of grouping objects into clusters, such that objects within a cluster are similar, and objects in different clusters are dissimilar. This process leads to a data partition, assuming that clusters are disjoint. Clustering algorithms have several applications, such as exploratory data analysis and data mining [9], and there are hundreds of them in the literature, handling different issues such as cluster shape, density and noise.

Clustering algorithms can be classified as partitional or hierarchical. Partitional methods assign each data object to exactly one cluster, and the number of clusters, k, is typically small and set by the user as a parameter. k-means is the most widespread partitional algorithm [12]; algorithms which estimate probability density functions from the data, such as Gaussian mixture decomposition algorithms [4,13,2], can also be used as partitional clustering techniques. Hierarchical methods yield a set of nested partitions which is graphically shown as a dendrogram [8] and a data partition is obtained by cutting the dendrogram at an appropriate level. Single-link and average-link are the most used hierarchical clustering algorithms [9,12].

Most clustering algorithms, *e.g.* k-means, have a diversity of solutions over the same dataset due to different initializations or parameters values. Recently, taking advantage of that diversity, an approach called *Clustering Ensemble methods*, has been proposed [5,11,10,3]. These methods propose a consensus partition, given a set of data partitions, based on a combination strategy. Clustering ensembles can be generated based on the

J.M. Sanches, L. Micó, and J.S. Cardoso (Eds.): IbPRIA 2013, LNCS 7887, pp. 535–542, 2013.

choice of data representation or on the choice of clustering algorithms or algorithmic parameters.

Fred and Jain [6] proposed the Evidence Accumulation method, which is a clustering ensemble approach based on the combination of information provided by a set of different partitions of a given dataset. To combine all the different partitions, Fred and Jain [6] proposed a voting scheme, which leads to a pairwise relationships matrix (similarity matrix between pairs of patterns), called *co-association matrix*. The final data partition is obtained by applying a clustering algorithm over the co-association matrix. One main advantage of this voting scheme is that it can deal with partitions having different number of clusters and different data representations.

Usually, clustering algorithms use the Euclidean distance between two points as a dissimilarity measure, but many other measures can be used [12]. However, it is difficult to choose a (dis)similarity measure since one has no prior knowledge about cluster shapes in the data. Recently, a new high order dissimilarity measure, called *dissimilarity increments*, has been proposed [7]. It is a high order dissimilarity measure which can identify sparse clusters, because it is computed over triplets of nearest neighbor points. Furthermore, it can give more information about patterns belonging to the same cluster, since dissimilarity increments change smoothly if the patterns are in the same cluster and high values of increments correspond to points lying in different clusters.

Moreover, the probability density function for dissimilarity increments was derived analytically under mild approximations [2]. That distribution was used to create a partitional clustering algorithm [2] and an hierarchical algorithm [1]. However, in both cases the Euclidean distance was considered to find the triplets of nearest neighbors.

In this paper, we propose to use the co-association matrix, which can be interpreted as a similarity matrix, to compute the dissimilarity increments; we then use hierarchical methods based on this measure to extract the consensus partition. We compare this approach with the Evidence Accumulation method proposed by Fred and Jain [6], and show that there is some improvement in the final partition. This happens due to the fact that the dissimilarity increments measure is more robust to sparse clusters.

This paper is organized as follows: Section 2.1 gives a background for the Evidence Accumulation Clustering and Section 2.2 presents the definition of dissimilarity increments and its distribution, among the hierarchical clustering algorithms based on this distribution. Section 2.3 presents the proposed method (EAC-DID) and experimental results and discussion are in Section 3. Conclusions are drawn in Section 4.

2 Proposed Methodology

We denote as $X = \{x_1, \dots, x_n\}$ a set of n objects represented in some feature space.

2.1 Evidence Accumulation Approach

A clustering algorithm takes X as input and groups the n points into k clusters, forming a partition P. A *clustering ensemble*, $\mathbb{P}$, is a set of N different partitions of the data X:

$$\mathbb{P} = \{P^1, P^2, \dots, P^N\} \tag{1}$$

$$P^1 = \{C_1^1, C_2^1, \ldots, C_{k_1}^1\}$$
$$\vdots$$
$$P^N = \{C_1^N, C_2^N, \ldots, C_{k_N}^N\},$$

where C_j^i is the jth cluster in data partition P^i, which has k_i clusters and n_j^i is the cardinality of C_j^i, with $\sum_{j=1}^{k_i} n_j^i = n, i = 1, \ldots, N$.

One can use several different clustering algorithms, or one algorithm with different initializations or parameters, to obtain multiple partitions of the same data, thus obtaining a clustering ensemble. The *evidence accumulation* approach [6] takes this ensemble and produces a co-association matrix by taking the co-occurrences of pairs of patterns in the same cluster as votes for their association. The idea is that patterns which should be grouped together are probably going to be assigned to the same cluster in different data partitions.

Formally, the N data partitions of n patterns yield a $n \times n$ co-association matrix:

$$\mathcal{C}(i,j) = \frac{n_{ij}}{N}, \tag{2}$$

where n_{ij} is the number of times the pattern pair (i,j) is assigned to the same cluster among the N partitions.

The standard Evidence Accumulation Clustering (EAC) paradigm finds a consensus solution by applying some clustering algorithm over the co-association matrix.

2.2 Clustering Algorithms Based on Dissimilarity Increments

Dissimilarity Increments Distribution. Let $\mathbf{x}_i$ be a point from X. The *dissimilarity increment* (DI) [7] associated with that point is computed as

$$d_{inc}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) = |d(\mathbf{x}_i, \mathbf{x}_j) - d(\mathbf{x}_j, \mathbf{x}_k)|. \tag{3}$$

where $\mathbf{x}_j$ is the nearest neighbor of $\mathbf{x}_i$ and $\mathbf{x}_k$ is the nearest neighbor of $\mathbf{x}_j$, different from $\mathbf{x}_i$.

Here, $d(\cdot,\cdot)$ is a pairwise (dis)similarity measure or distance. The quantity d_{inc} measures higher-order information about the data, since it is a measure for a triplet of points, whereas typical distance measures use only two points.

The DIs distribution (DID) was derived in [2], using the Euclidean distance as the dissimilarity measure $d(\cdot,\cdot)$, under the hypothesis of Gaussian distribution of the data. This distribution was written as a function of the mean value of the DIs, which is denoted as λ. The mathematical expression of the DID is given by

$$\begin{aligned} p_{d_{inc}}(w;\lambda) = \frac{\pi\beta^2}{4\lambda^2} w \exp\left(-\frac{\pi\beta^2}{4\lambda^2}w^2\right) + \frac{\pi^2\beta^3}{8\sqrt{2}\lambda^3}\left(\frac{4\lambda^2}{\pi\beta^2} - w^2\right) \times \\ \times \exp\left(-\frac{\pi\beta^2}{8\lambda^2}w^2\right) \operatorname{erfc}\left(\frac{\sqrt{\pi}\beta}{2\sqrt{2}\lambda}w\right), \end{aligned} \tag{4}$$

where $\operatorname{erfc}(\cdot)$ is the complementary error function, and $\beta = 2 - \sqrt{2}$.

Hierarchical Clustering Algorithms. In [1] an agglomerative hierarchical algorithm was proposed, called SLDID, which was a variant of single-link (SL) using the DID to change the way that clusters are merged to form bigger clusters. That algorithm has two main features: it can adequately identify well separated clusters with arbitrary shapes and densities, and it offers a deeper insight into the structure of touching clusters. It can often find a set of clusters such that each class is the union of a few clusters, *i.e.*, SLDID is able to find classes that are the union of smaller models, each of which is governed by the DID with some parameter λ.

In this paper, we also consider two other clustering algorithms based on dissimilarity increments: ALDID and CLDID. They are variants of average-link (AL) and complete-link (CL) in the same sense that SLDID is a variant of SL. Below, we briefly explain how SLDID differs from SL; ALDID and CLDID are constructed in a similar way. The reader is referred to [1] for further details regarding, *e.g.*, the choice of parameters.

All three algorithms are agglomerative ones; this means that we start by having each data point in a separate cluster, and iteratively make decisions on which pair of clusters to join. In SL, AL and CL, this procedure continues until all points belong to a single cluster; in SLDID, ALDID and CLDID, the final situation may have more than one cluster.

The similarity between two clusters in SLDID is computed exactly in the same way as in SL and, just like in SL, the most similar pair of clusters is selected. The difference is that in SL, the most similar pair of clusters at each iteration is always merged; in SLDID, some tests are made, using the dissimilarity increments distribution, and the results of these tests determine whether that pair of clusters is merged or not. If that pair of clusters is not merged, the second most similar pair is then tested, and so on [1].

These tests essentially check whether the DID of the two clusters combined is better than the DIDs of the two clusters separated. Here, "better" is rigorously defined as a minimum description length (MDL) criterion which selects between the two possibilities.

2.3 The Method: EAC-DID

Having described clustering ensembles and hierarchical clustering algorithms based on dissimilarity increments, we now tie the two concepts together. SL uses the minimum Euclidean distance between points of two clusters to determine the distance between those clusters. Similarly, AL and CL use the average and maximum Euclidean distances, respectively. SLDID, ALDID and CLDID also use these distances (recall that the difference is in whether two clusters are merged or not). In this paper, we propose that this Euclidean distance is replaced by the dissimilarity as measured by the co-association matrix, a procedure we called EAC-SLDID, EAC-ALDID and EAC-CLDID.

Note that $\mathcal{C}(i,j)$, as defined in (2), is 0 for pairs of points which are never clustered together and 1 for pairs of points which are always clustered together. It makes sense, then, to define a distance according to the following expression:

$$d(\mathbf{x}_i, \mathbf{x}_j) = 1 - \mathcal{C}(i,j). \tag{5}$$

The overall procedure is schematically described in figure 1.

The clustering ensembles are constructed using a split and merge strategy. We run the k-means algorithm to produce a total of $N = 200$ data partitions. For each partition, we

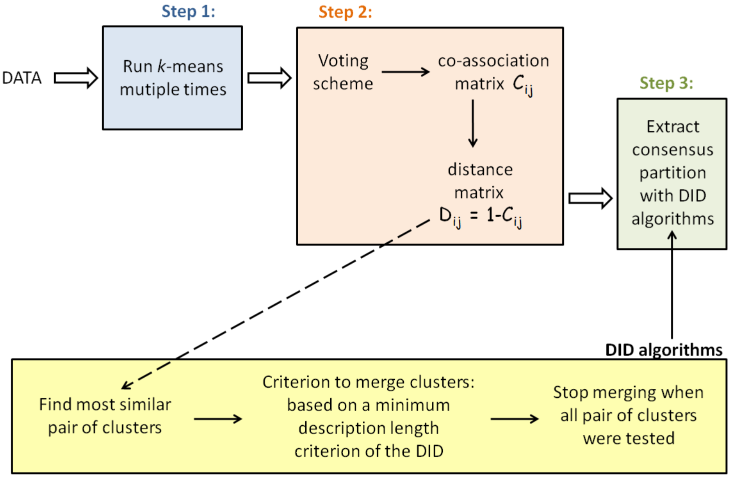
Step 1:
Step 2:
Step 3:
DATA
Run *k*-means mutiple times
Voting scheme
co-association matrix C_{ij}
distance matrix $D_{ij} = 1-C_{ij}$
Extract consensus partition with DID algorithms
DID algorithms
Find most similar pair of clusters
Criterion to merge clusters: based on a minimum description length criterion of the DID
Stop merging when all pair of clusters were tested

Fig. 1. Outline of the proposed method (EAC-DID)

start by determining the number of clusters k randomly, by sampling it from a uniform distribution between $k_{min} = \max\{\sqrt{n}/2, n/50\}$ and $k_{max} = k_{min} + 20$, where n is the number of samples of the dataset. A random initial position is chosen for each centroid, ensuring that even if repeated values of k are encountered, different partitions can be obtained. Each clustering combination was applied 30 times for each dataset.

3 Experimental Results and Discussion

In the experiments we used 14 datasets: six synthetic datasets and eight real-world datasets from the UCI Machine Learning Repository[1]. The synthetic datasets are shown in figure 2 and the real datasets are in table 1.

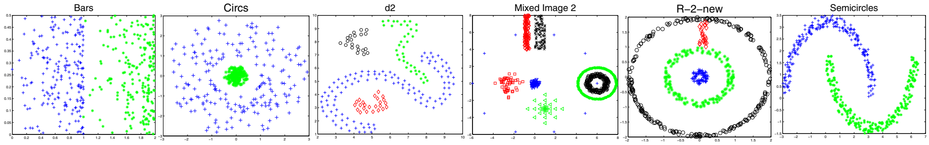

Fig. 2. Synthetic datasets

We used single-link (SL), average-link (AL) and complete-link (CL) to compare with our dissimilarity increments based clustering algorithms: SLDID, ALDID and CLDID. In all the algorithms we set the number of clusters to the true number (*i.e.*, we cut the

[1] `http://archive.ics.uci.edu/ml`

Table 1. Real-world datasets with the corresponding number of samples (Ns), number of features (Nf) and number of clusters (Nc)

Data	Ns	Nf	Nc	Data	Ns	Nf	Nc
breast	683	9	2	iris	150	4	3
crabs	200	5	2	optdigits	1000	64	10
house-votes	232	16	2	pima	768	8	2
ionosphere	351	34	2	wine	178	13	3

dendrogram at the true number of clusters). However, since the DID algorithms have other criteria, most of the times those algorithms stop before they reach that true number of clusters; when that happens, we use the final situation as the result.

To measure the performance of each algorithm we used two measures: consistency index and adjusted consistency index. The consistency index (CI) is the percentage of points that are well clustered compared to true labeling [5]; the adjusted consistency index, denoted as CI*, is a variant of CI which considers each cluster as the union of several subclusters [1]. This is consistent with our consideration mentioned above, regarding the characterization of a class as possibly being composed of more than one cluster, each of which follows a DI distribution. Tables 2 and 3 shows the results of these two measures.

Table 2. Consistency index values of the partitions found by the algorithms

	Euclidean						EAC					
Data	SL	SLDID	AL	ALDID	CL	CLDID	SL	SLDID	AL	ALDID	CL	CLDID
bars	0,50	0,91	**0,99**	0,87	**0,99**	0,56	0,55	0,52	**0,99**	0,22	0,54	0,36
circs	**1,00**	0,99	0,62	0,48	0,71	0,54	**1,00**	0,67	**1,00**	0,26	0,59	0,41
d2	**1,00**	0,92	0,62	0,57	0,62	0,57	**1,00**	0,81	0,61	0,56	0,39	0,59
mixed image 2	0,47	0,77	0,52	0,68	0,51	0,48	**0,82**	0,77	0,54	0,44	0,39	0,38
r-2-new	0,59	0,62	0,34	0,43	0,37	0,37	**0,76**	0,48	0,74	0,28	0,57	0,39
semicircles	**1,00**	0,87	0,79	0,60	0,82	0,36	**1,00**	0,41	**1,00**	0,26	0,60	0,25
breast	0,65	0,45	**0,94**	0,34	0,85	0,52	0,65	0,54	0,85	0,11	0,58	0,36
crabs	0,51	0,48	**0,56**	0,35	0,51	0,25	0,51	0,50	0,51	0,35	0,54	0,50
house-votes	0,53	0,54	**0,91**	0,44	**0,91**	0,52	0,66	0,58	0,87	0,48	0,55	0,57
ionosphere	0,64	0,64	0,64	**0,76**	0,69	0,42	0,65	0,57	0,72	0,30	0,65	0,31
iris	0,68	0,67	**0,91**	0,63	0,84	0,75	0,77	0,53	**0,91**	0,64	0,74	0,45
optdigits	0,11	0,67	0,76	0,52	0,52	0,49	0,46	0,59	**0,80**	0,37	0,47	0,38
pima	**0,65**	0,27	**0,65**	0,18	**0,65**	0,18	**0,65**	0,49	0,52	0,10	0,63	0,34
wine	0,43	0,37	0,61	0,37	0,67	0,39	0,69	0,51	**0,71**	0,48	0,52	0,47
Average	0,63	0,65	0,70	0,52	0,69	0,46	0,73	0,57	**0,77**	0,35	0,55	0,41
Std	0,25	0,22	0,19	0,18	0,18	0,14	0,18	0,11	0,18	0,16	0,10	0,09

Table 2 presents results of the six clustering algorithms discussed in this paper in two scenarios: using the Euclidean distance on the original data space (marked ”Euclidean” on the table) and using the distance obtained from the co-association matrix (marked

Table 3. Adjusted consistency index values of the partitions found by the algorithms

	Euclidean						EAC					
Data	SL	SLDID	AL	ALDID	CL	CLDID	SL	SLDID	AL	ALDID	CL	CLDID
bars	0,50	**1,00**	0,99	0,99	0,99	0,97	0,55	0,92	0,99	0,98	0,54	0,80
circs	**1,00**	0,99	0,62	0,99	0,71	0,99	**1,00**	0,83	**1,00**	0,99	0,59	0,84
d2	**1,00**	**1,00**	0,74	**1,00**	0,73	**1,00**	**1,00**	0,88	0,78	0,72	0,61	0,68
mixed image 2	0,48	0,96	0,52	0,97	0,52	**0,98**	0,82	0,96	0,72	0,95	0,53	0,93
r-2-new	0,65	0,81	0,53	0,85	0,52	0,92	0,83	0,89	0,89	**0,95**	0,66	0,87
semicircles	**1,00**	**1,00**	0,79	0,98	0,82	**1,00**	**1,00**	**1,00**	**1,00**	0,99	0,60	0,98
breast	0,65	0,67	0,94	0,87	0,85	0,94	0,65	0,90	0,85	**0,97**	0,66	0,82
crabs	0,51	**0,62**	0,56	0,59	0,51	0,59	0,51	0,54	0,51	0,57	0,54	0,55
house-votes	0,53	0,54	**0,91**	0,67	0,91	0,72	0,66	0,61	0,87	**0,91**	0,55	0,60
ionosphere	0,64	0,64	0,64	0,82	0,69	0,75	0,65	**0,91**	0,72	0,87	0,68	0,86
iris	0,68	0,67	**0,91**	0,79	0,84	0,75	0,77	0,53	**0,91**	0,77	0,75	0,48
optdigits	0,11	0,80	0,78	0,75	0,59	0,78	0,47	0,65	0,81	**0,91**	0,51	0,72
pima	0,65	0,71	0,65	0,72	0,65	0,71	0,65	0,68	0,65	**0,73**	0,66	0,69
wine	0,43	0,67	0,65	0,53	0,67	0,57	**0,71**	0,58	**0,71**	**0,71**	0,53	0,58
Average	0,63	0,79	0,73	0,82	0,71	0,83	0,73	0,78	0,81	**0,86**	0,60	0,74
Std	0,25	0,17	0,16	0,16	0,15	0,15	0,18	0,17	0,14	0,13	0,07	0,15

”EAC”). The table suggests that SL performs better than other algorithms in synthetic data, but overall AL is the best one in terms of consistency index. This is true whether one uses the clustering algorithms on the original data or using the EAC framework. While there are advantages and disadvantages when comparing the Euclidean and the EAC parts of the table, the best overall algorithm in terms of average CI is EAC-AL.

It seems from this table that DID algorithms performed poorly. This happens because those algorithms stop merging clusters before the true number of clusters is reached, which means that they may have found several clusters for each class. In other words, the DID algorithms did not benefit from knowing the true number of clusters, whereas SL, AL and CL do. This implies that the comparisons in Table 2 are somewhat biased in favor of the non-DID algorithms. This is the reason why we chose to use the adjusted CI.

Table 3 shows that all DID family of clustering algorithms have higher adjusted consistency index compared to the homologous clustering algorithm, *i.e.*, SLDID compared to SL, and so on. In general, the DID algorithms are better than their non-DID counterparts; this is true for both the Euclidean and the EAC frameworks. Furthermore, EAC-ALDID turns out to be the best method, both in terms of average adjusted CI and in how often it is the best algorithm for a given dataset.

The adjusted consistency index is a fairer way to compare algorithms in this situation, since the family of DID algorithms often stops merging clusters before the true number of clusters is attained. However, ideally, some post-processing step should be used where clusters belonging to a class would be merged together to form a single cluster, allowing the original consistency index to be used for a fair comparison.

4 Conclusions

This paper has proposed to join two distinct concepts: the evidence accumulation clustering (EAC) framework and the dissimilarity increments (DI) distribution. Whereas DIs are normally computed using the Euclidean distance, we propose that the co-association matrix, which results from EAC, can be plugged-in to replace that distance.

The conjunction of these two concepts results in a new family of clustering algorithms; those algorithms have obtained promising results when compared to classic hierarchical clustering algorithms using an adjusted consistency index which allows fair comparison between them.

Acknowledgments. This work was supported by the Portuguese Foundation for Science and Technology grants PTDC/EIA-CCO/103230/2008 and PTDC/EEI-SII/2312/2012.

References

1. Aidos, H., Fred, A.: Hierarchical clustering with high order dissimilarities. In: Proc. Int. Conf. on Machine Learning and Data Mining (MLDM 2011), pp. 280–293 (2011)
2. Aidos, H., Fred, A.: Statistical modeling of dissimilarity increments for d-dimensional data: Application in partitional clustering. Pattern Recognition 45(9), 3061–3071
3. Ayad, H.G., Kamel, M.S.: Cluster-based cumulative ensembles. In: Oza, N.C., Polikar, R., Kittler, J., Roli, F. (eds.) MCS 2005. LNCS, vol. 3541, pp. 236–245. Springer, Heidelberg (2005)
4. Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of finite mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(3), 381–396
5. Fred, A.: Finding consistent clusters in data partitions. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 309–318. Springer, Heidelberg (2001)
6. Fred, A., Jain, A.: Combining multiple clusterings using evidence accumulation. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(6), 835–850
7. Fred, A., Leitão, J.: A new cluster isolation criterion based on dissimilarity increments. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(8), 944–958
8. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer Series in Statistics. Springer (2009)
9. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3), 264–323
10. Kuncheva, L.I., Hadjitodorov, S.T.: Using diversity in cluster ensembles. In: Proc. Int. Conf. on Systems, Man and Cybernetics, pp. 1214–1219
11. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3, 583–617
12. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 4th edn. Elsevier Academic Press (2009)
13. Ueda, N., Nakano, R., Ghahramani, Z., Hinton, G.E.: SMEM algorithm for mixture models. Neural Computation 12(9), 2109–2128 (2000)