2006

Clustering Similarity Comparison Using Density Profiles

Eric Bae

James Bailey

Guozhu Dong
Wright State University - Main Campus, guozhu.dong@wright.edu

Repository Citation

Bae, E., Bailey, J., & Dong, G. (2006). Clustering Similarity Comparison Using Density Profiles. *Lecture Notes in Computer Science, 4304*, 342-351.
https://corescholar.libraries.wright.edu/knoesis/115

Clustering Similarity Comparison Using Density Profiles

Eric Bae[1], James Bailey[1], and Guozhu Dong[2]

[1] NICTA Victoria Laboratory, Department of Computer Science and Software Engineering, University of Melbourne, Australia
[2] Department of Computer Science and Engineering, Wright State University, USA

Abstract. The unsupervised nature of cluster analysis means that objects can be clustered in many different ways. This means that different clustering algorithms can lead to vastly different results. To address this, clustering similarity comparison methods have traditionally been used to quantify the degree of similarity between alternative clusterings. However, existing techniques utilize only the point-to-cluster memberships to calculate the similarity, which can lead to unintuitive results. They also can't be applied to analyze clusterings which only partially share points, which can be the case in stream clustering. In this paper we introduce a new measure named ADCO, which takes into account density profiles for each attribute and aims to address these problems. We provide experiments to demonstrate this new measure can often provide a more reasonable similarity comparison between different clusterings than existing methods.

1 Introduction

Cluster analysis is a fundamental machine learning task in which patterns, relationships and structures of interest in data are discovered in an unsupervised manner. It has been used in a wide variety of fields, including biomedicine, information retrieval and financial institutions, to discover hidden knowledge and information.

However, clustering is naturally an ill-posed problem [19], where the act of grouping similar data objects is a subjective notion and highly dependent on the clustering criterion used. For this reason, a vast number of algorithms have been developed, each aiming to address different aspects of the problem, yet such algorithms often provide very different results. Moreover, even when a single algorithm is used, different alternative clusterings[3] can easily be generated, simply by changing the initial conditions of the algorithm.

Therefore, in order to provide a measure of comparison between clusterings, cluster analysis has been often accompanied by a comparison method. Formally called external validation [15], this provides a quantitative measure of the degree to which two different clusterings are similar/different.

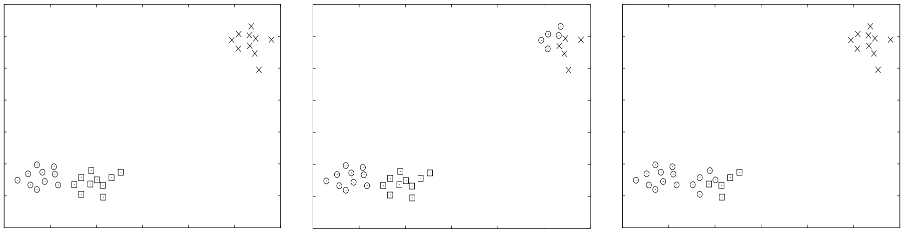
However, the current comparison measures suffer from a fundamental problem of judging the clustering similarity/difference purely on the membership of points to clusters. While these point-to-cluster assignments can be an important

[3] A clustering is a set of clusters

determining factor in defining clusterings, they completely neglect other important aspects of data, which can seriously affect the outcome. These measures also suffer from the limitation that they are not applicable for comparing clusterings which may partially or not at all share points.

2 Problems and motivations

We illustrate the problem in figure 1. Here we have three clusterings, each with three clusters. Figure 1(a) is a pre-defined clustering which is compared against 1(b) and 1(c). Both clusterings 1(b) and 1(c) have five points clustered differently compared to 1(a).



(a) pre-defined clustering

(b) clustering A

(c) clustering B

Fig. 1. Pre-defined clustering 1(a) containing 3 clusters, compared to two clusterings 1(b) and 1(c). Membership based measures give the exactly same values for both comparisons

Let clustering comparison A be between 1(a) and 1(c), while comparison B is between 1(a) and 1(b). When comparing in terms of either cluster representatives (i.e. centroids), shapes or point distributions of clusters it seems intuitive that the degree of similarity for comparison A should not be the same as the degree of similarity for comparison B. For example, suppose a new point were added to the dataset and it was merged with the closest cluster. It seems more probable that in both 1(a) and 1(c) it would join the same cluster. However, for 1(b), it is more likely that it might join a different cluster than 1(a). This is because 1(a) and 1(c) share a higher structural similarity, than 1(a) and 1(b). However, the available comparison measures are not able to recognize this difference. For example, a popular pair-counting measure, the Rand Index [7], gives a similarity value of 0.44 for both comparisons[4]. In fact, it is easily possible to generate arbitrary clusterings, which give the exactly same Rand index value when compared to 1(a), provided it has just five points clustered differently. Therefore treating point-to-cluster assignments as a primary (if not only) measure of comparison has limitations and does not necessarily correspond with intuition.

In this paper, we address this problem by developing a new clustering (dis)similarity measure we term ADCO[5]. The contribution of ADCO is to address two main limitations of existing methods.:

[4] see section 3 for details of this and other measures and section 5 for more detailed experimental results.

[5] **Attribute Distribution Clustering Orthogonality**

- **Addressing Non intuitive Behaviour** : ADCO incorporates distribution information of data points along each attribute, allowing consideration of the shapes or density profiles of the clusters. This can provide more detailed and intuitive comparisons than simple membership based techniques such as Rand [7] or Jaccard [8] indices.
- **Applicability to stream data clustering** : ADCO can compare clusterings that may be built upon entirely different point sets and thus support the post-analysis phase in stream data clustering, where clusterings using different stream windows are compared. Comparison of clusterings using different sets of points is impossible for membership based techniques.

3 Related Work

The traditional clustering comparison methods are divided into three categories : 1) pair counting, 2) set matching and 3) variation of information (see table 1). In the pair-counting category, Rand Index [7] and Jaccard Index [8] have been widely used for their simplicity. These methods are based on whether a pair of points belong to the same or different clusters in each clustering and these methods have also been extended in [4, 6]. For set-matching methods, Clustering Error [17] has been a popular choice which matches the 'best' clusters between two clusterings based on the number of points they share. The comparison is given by the total number of points shared between pairs of matching clusters over all the points. Other set-matching methods are described in [3, 17]. Finally, Variation of Information introduced in [18] is based on information theory measuring the amount of mutual information between two clusterings via the number of points they share. More recently, authors in [20] applied Mallows distance function to cluster representatives to calculate a comparison. Its method, although it addresses a similar problem to ours, is nevertheless still more similar to the membership-based approaches, supplementing them additional information about cluster centroids.

Table 1. Definitions of Rand index (RI), Jaccard index (JI), Clustering Error (CE) and Variation of Information (VI). For RI and JI, N_{11} and N_{00} refer to the 'agreement' while N_{10} and N_{01} are 'disagreement' values between two clusterings. For CE, n is the number of objects and K is the number of clusters in each clustering. $n_{k,\sigma(k)}$ finds the 'best match' between pairwise clusters. For VI, $H(C)$ refers to the entropy of the clustering C, while $H(C, C')$ is the joint entropy of two clusterings.

RI	$RI(C,C') = \frac{N_{11}+N_{00}}{N}$	JI	$JI(C,C') = \frac{N_{11}}{N_{11}+N_{10}+N_{01}}$
CE	$CE(C,C') = 1 - \frac{1}{n} max \sum_{k=1}^{K} n_{k,\sigma(k)}$	VI	$VI(C,C') = 2H(C,C') - H(C) - H(C')$

Clustering comparison methods have also been applied within the context of *ensemble clustering*, where several clusterings are merged to form a consensus clustering. A popular technique for merging is called 'majority voting' [11, 12] which is a pair-counting method extended over multiple clusterings. Using a co-association matrix of data points, where pairs of points are given a score if they appear in the same cluster over all available clusterings. The pairs with a score higher than predefined threshold are then 'voted' to be in the same cluster. In [14], clusterings are represented as a set of connected hypergraphs. Here, vertices connected by edges are objects in the same cluster over all clusterings. HyperGraph Partitioning algorithm [2] is then applied to find the consensus clustering by cutting a minimum number of hyper-edges. Although the approach is different, its underlying idea is to find highly dense intersections between clusterings and the method is considered as a variant to membership based methods.

Another area where comparison methods are used is in stream data clustering [5, 6], which has become increasingly popular in recent times. This raises an interesting analysis task, as clusterings can evolve over time and studying this evolution can uncover valuable information. In [5] Aggarwal describes this evolution and its evaluation where clusterings at different time periods are compared. In this work, clusterings at different periods are compared by observing any newly formed, removed or modified clusters. The technique used is membership-based and it is assumed that clusterings have at least some non-empty overlaps of data points, meaning windows for which clusterings do not share any points cannot be compared.

4 The ADCO Similarity Measure

We now present our new measure for comparing (dis)similarity between clusterings. Firstly though, we provide some necessary definitions.

4.1 Background and Terminology

Let $D = \{d_1, d_2, .., d_n\}$ be a dataset of n objects, described by r attributes $\{a_1, \ldots, a_r\}$. Let $d_i[a_j]$ refer to the value of object d_i on attribute a_j. A *clustering* C, is partition of d into a set of clusters. i.e. $C = \{c_1, \ldots, c_k\}$, where each c_i is a *cluster* (set of points).

Let $C_1 = \{c_1, \ldots, c_k\}$ and $C_2 = \{c'_1, \ldots, c'_k\}$ be the two clusterings which will be compared. Note that we assume the number of clusters in each clustering must be the same (an assumption also shared by existing measures). The *similarity* between two clusterings, $Sim(C_1, C_2)$, is a function which computes their similarity, with higher values of the measure indicating higher dissimilarity (less similarity). Various similarity measures have been defined in existing work, e.g. Rand index (Sim_{Rand}) [7], Jaccard index ($Sim_{Jaccard}$ [8]). Our measure will be referred to as ADCO, or Sim_{ADCO}.

4.2 Computing the ADCO Similarity Measure

Our ADCO similarity measure aims to determine the similarity between two clusterings based on their density profiles along each attribute. Essentially, r-dimensional space is chopped up into a "hyper grid". Points from the dataset occupy exactly one of the cells in this grid. The similarity between two clusters corresponds to how similarly the point sets from each cluster are distributed across the grid. The similarity between two clusterings then corresponds to the amount of similarity between their component clusters.

Suppose (the range of) each attribute a_i is divided into q bins, using some discretisation method. Let a_i^j refer to the set of values of attribute a_i within bin j.

Figure 2 shows two clusterings, each with two clusters ($k = 2$) and each attribute has been divided into two bins ($q = 2$).

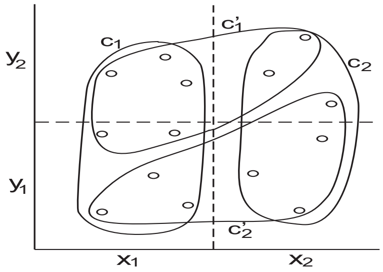


Fig. 2. Two Clusterings C and C' with attributes X and Y binned into q number of intervals where $C = \{c_1, c_2\}$ and $C' = \{c'_1, c'_2\}$

To compute Sim_{ADCO}, we start by computing the density of cluster c for each bin j along each attribute a_i.

$$dens_c(a_i, j) = |\{d \in c \mid d[a_i] \in a_i^j\}| \tag{1}$$

where $dens_{c_i}(a_i, j)$ is the density (or number of points) of cluster c for an attribute-bin pair (a_i, j). For a given cluster c, we can compute the $dens_c(a_i, j)$ measure for all bins (r of them), of all attributes (q of them), giving $r \times q$ measures in total. Using some arbitrary ordering scheme, we can then form a vector of length $r \times q$ containing these measures, $dens_c = \{dens_c(a_1, 1), dens_c(a_1, 2), \ldots, dens_c(a_r, q)\}$.

Example : For figure 2 where two clusterings C and C' are present, let $(X, x_1), (X, x_2), (Y, y_1), (Y, y_2)$ be the ordering of the attribute bin pairs. Then $dens_{c_1} = (8, 0, 5, 3)$, $dens_{c_2} = (0, 6, 3, 3)$, $dens_{c'_1} = (5, 2, 2, 5)$, $dens_{c'_2} = (3, 4, 6, 1)$.

It is now possible to compare two clusterings, by measuring the similarity between their component clusters with a dot product operation as follows:

$$C \cdot C' = \sum_{i=1}^{k} dens_{c_i} \cdot dens_{c'_i} . \tag{2}$$

where the $\cdot$ refers to the dot product between two vectors, with a zero value indicating that the two clusterings are orthogonal (dissimilar) and a large value indicating that they are similar.

Equation 2 compares clusters on a pairwise basis, (c_1 versus c'_1, c_2 versus c'_2, etc). However, it is important that our similarity measure be independent of the actual cluster names assigned. We thus need to be able to permute the second clustering, so as to calculate all possible pairings of clusters from C_1 and C_2. From these permutations, we select the pairing which gives a maximum value. i.e.

$$PairwiseSim(C, C') = max_P[C \cdot P(C')] \tag{3}$$

where P ranges over all permutations of C'. For figure 2 with a pairing $C = (c_1, c_2)$ and $C' = (c'_1, c'_2)$, we get $C \cdot C' = 110$. In contrast, if C' is permuted such that c'_2 is renamed to c'_1 and c'_1 is renamed to c'_2, we get $C \cdot P(C') = 90$. This indicates that the first pairing of c_1 and c'_1 and c_2 and c'_2 gives the largest value.

The final step is then to normalize this pairwise similarity value, with respect to the maximum possible similarity. This is then subtracted from 1, so that 0 indicates highly similar and 1 indicates highly dissimilar.

$$Sim_{ADCO}(C, C') = 1 - \frac{PairwiseSim_D(C, C')}{MaxSim(C, C')} \tag{4}$$

where $MaxSim(C, C') = max(C \cdot C, C' \cdot C')$. Note that $MaxSim(C, C') = max(C \cdot C, C' \cdot C')$ is the upper bound on the dot product values involving at least one of C and C'. For the example of figure 2, $Sim_{ADCO}(C, C') = 0.276$.

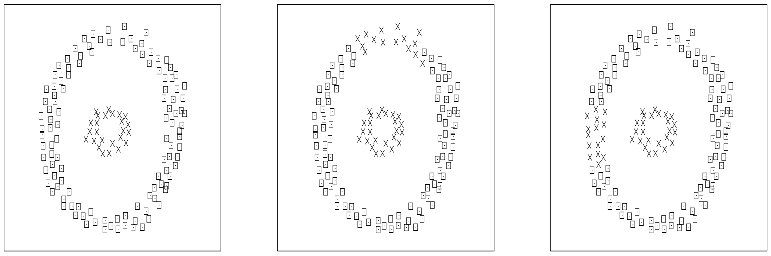
ADCO Properties Finally, we briefly describe properties ADCO.

- **Positivity**[6] : The value of $ADCO(C, C') = 0$ if and only if $C = C'$
- **Symmetry** : $ADCO(C, C') = ADCO(C', C)$
- **Triangle inequality** : For any three clusterings C, C' and C'',

$$ADCO(C, C') + ADCO(C', C'') \geq ADCO(C, C''). \tag{5}$$

The advantages of the above two properties are well understood and described in [17].

5 Experiments



(a) pre-defined clustering

(b) clustering A

(c) clustering B

Fig. 3. Pre-defined clustering compared to two clusterings 3(b) and 3(c). Membership based measures give exactly same values for both comparisons.

We compared the behaviour of ADCO with several of the existing measures for comparing clusterings. The following measures were compared against: Rand Index (RI) [7], Jaccard Index (JI) [8], Clustering Error (CE) [17], Variation of Information (VI) [17]. Clusterings were generated using k-means, Expectation Maximization (EM), CURE, FarthestFirst (FF), Average-Linkage (AL), Complete-Linkage (CL) and Single-Linkage (SL). All initial parameters of these algorithms were kept constant throughout the experiment and values are measured between 0 and 1 where a high value indicates a high dissimilarity. For all experiments, we set the number

[6] This property may not apply when ADCO is used for stream clusterings where no points are shared between clusterings

of bins q to 10, which is a commonly used choice for discretising data [1].

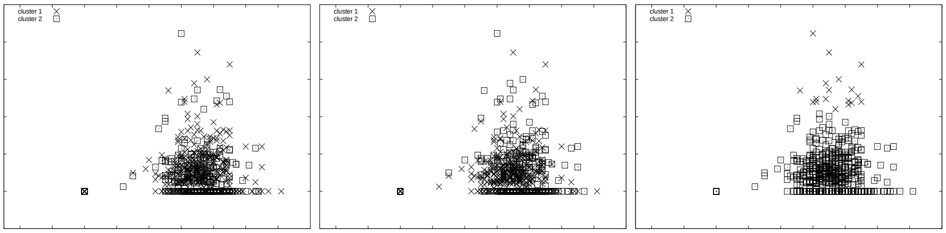
Synthetic Datasets: Two synthetic datasets are shown in figure 1 and 3. As already mentioned in section 2, in figure 1 clusterings in figure 1(a) and 1(c) are more similar than 1(a) and 1(b). Similarly in figure 3, figure 3(a) and 3(c) are closer with in regards to point distributions in clusters than 3(a) and 3(b). Table 2 clearly shows how ADCO can recognise this distinction, while other measures completely fail to do so, giving the same value for all comparisons.

Table 2. Dissimilarity values when comparing 3(a) with 3(c) and 3(b) as well as comparing 1(a) with 1(c) and 1(b). For both datasets, ADCO is the only measure that detects the structural difference.

	ADCO	RI	JI	CEM	VI
figures 1(a) vs. 1(c)	**0.16**	0.17	0.41	0.17	0.16
figures 1(a) vs. 1(b)	**0.47**	0.17	0.41	0.17	0.16
figures 3(a) vs. 3(c)	**0.16**	0.44	0.45	0.32	0.15
figures 3(a) vs. 3(b)	**0.33**	0.44	0.45	0.32	0.15

Real Datasets: We looked at two real world datasets, 'diabetes' and 'credit'. Each dataset comes with a pre-defined clustering (the natural clusters, identified using the class labels), which we then compared against clusterings generated by each of the clustering algorithms.

The dataset 'diabetes' in Fig. 4, contains two natural clusters. In Fig. 4 we also display the clusterings of k-means and AL, projected onto two attributes, to assist in visualisation. Similar to previous examples, we can see that figure 4(c) is more dissimilar to the pre-defined clustering than the clustering in figure 4(b). However, when we observe the table of comparison measures in 3, this dissimilarity is not reflected by four membership based measures. In fact, ADCO is the only measure that can correctly describe this increase in dissimilarity from k-means to AL.



(a) pre-defined

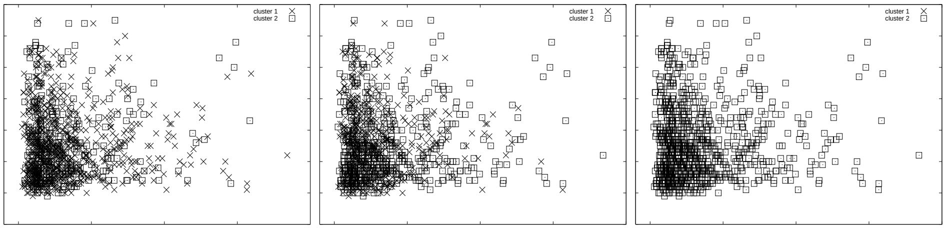
(b) k-means

(c) average-link

Fig. 4. Three clusterings of the diabetes dataset projected onto two attributes

On another dataset 'credit', we also see a similar trend. In figure 5, the comparison between 5(a) and 5(c) is more dissimilar than the comparison between 5(a) and 5(b). Looking at table 3, ADCO is the only measure that can recognise this correctly.

We can connect these results to the problems of traditional membership based methods mentioned in the section 2. It is clear that from the figures 4 and 5, the clusterings differ in membership of the points, as well as their point distributions.



(a) pre-defined (b) CURE (c) single-link

Fig. 5. Three clusterings of the credit dataset.

Table 3. Dissimilarity values when comparing seven clusterings using five dissimilarity measures for dataset diabetes and credit

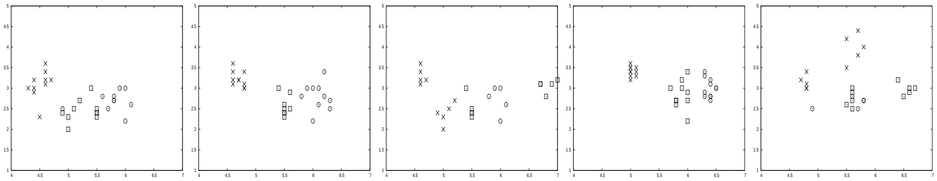
dataset	measures	k-means	EM	FF	CURE	AL	CL	SL
diabetes	ADCO	0.02	0.03	0.03	0.07	0.25	0.07	0.3
	RI	0.5	0.5	0.49	0.5	0.44	0.5	0.42
	JI	0.61	0.61	0.61	0.62	0.46	0.62	0.42
	CE	0.44	0.43	0.42	0.45	0.33	0.46	0.3
	VI	0.18	0.18	0.18	0.19	0.12	0.19	0.09
credit	ADCO	0.17	0.1	0.2	0.16	0.33	0.26	0.26
	RI	0.5	0.5	0.49	0.5	0.45	0.5	0.5
	JI	0.63	0.64	0.57	0.62	0.47	0.63	0.64
	CE	0.45	0.48	0.42	0.45	0.35	0.45	0.45
	VI	0.2	0.2	0.17	0.19	0.11	0.2	0.2

In particular, 4(c) and 5(c) show higher dissimilarity when compared to the pre-defined clusterings 4(a) and 5(a), than clusterings of other algorithms. This dissimilarity is correctly displayed through the ADCO measure. However, all other measures incorrectly capture these comparisons by actually giving a smaller value, implying that 4(c) and 4(c) are actually more similar to the pre-defined clusterings than others. This is because these methods consider only the intersected groups of points between two clusterings, regardless of the overall structures of clusterings. For all other real world datasets that were tested, we have observed the same problem and ADCO was the only measure which was able to highlight the differences accurately.

5.1 Using ADCO for evolution analysis in data stream clustering

As mentioned in section 2, clustering comparison methods can be useful for stream data, for determining how clusters/clusterings evolve over time [5]. However, the membership based measures have the requirement that the data points in each clustering should be the same. Hence clusterings with a large time gap between them in the stream (thus using totally different points) cannot be compared. In contrast, ADCO's use of attribute based density profiles of attributes makes such a comparison possible, as we now illustrate.

In figure 6 we have divided the dataset 'IRIS' into five subsets, where subset 1 and 2 share 50% of the points while rest of the points are different, though similar in their values. Therefore, when comparing 1 and 2, ADCO should return a low dissimilarity value. Subsets 2 and 3 are similar to 1 and 2, but non-overlapping points are different, hence ADCO value should be higher. Subsets 3 and 4 are independent, but their values are similar and subsets 4 and 5 are independent and their values are highly different. Hence, ADCO values should be low for the comparison of 3 and 4, while comparing 4 and 5 should be high. These comparisons can be seen as evaluating clustering evolution by comparing data at different time periods.



(a) sample 1 (b) sample 2 (c) sample 3 (d) sample 4 (e) sample 5

Fig. 6. 5 samples IRIS dataset. Each sample has 30 points and 3 clusters. 6(a) and 6(b) share 15 points and other 15 do not overlap but of similar values. 6(b) and 6(c) also share 15 points, but other 15 are quite different. 6(c) and 6(d) do not share any points yet the points are of similar values and 6(d) and 6(e) do not share any points and are quite different.

As can be seen from the table 4, ADCO measure corresponds with expectation. This means that using ADCO may be able to help speed up the stream analysis process, by allowing users to compare clusterings from any time windows, and they can investigate further if ADCO value indicates there is significant dissimilarity.

Table 4. Dissimilarity values between different samples of the dataset IRIS

sample pairs	ADCO
sample 1 & sample 2	0.15
sample 2 & sample 3	0.37
sample 3 & sample 4	0.31
sample 3 & sample 5	0.41

5.2 ADCO Limitations

Although providing more accurate measures by corresponding to the intuition, current state of ADCO is limited to only serve clusterings with equal number of clusters. Moreover, we have not discussed methods to handle soft clusterings or subspace clusterings. ADCO would need to be generalized to handle these cases. Finally while we have used $q = 10$ as a best-practice value for the number of bins, the real impact of this variable can be studied further.

6 Future Work and Conclusion

Clustering comparison is an important task in the overall cluster analysis process. In this paper we have discussed the limitations of existing methods, which consider only point-to-cluster assignments as the determining factor for dissimilarity between clusterings. This ignores other important feature-related information and may mislead users with inaccurate or even incorrect evaluations.

We have presented a new measure, ADCO, which takes a different approach to determining clustering dissimilarity, by using the distribution information for each attribute. We have experimentally shown that ADCO can indeed lead to more reasonable measures, than existing methods. Furthermore, we have identified an important application of ADCO for stream data clustering, where it is able post analysis on clusterings from a stream drawn from non overlapping windows. This kind of analysis is impossible with the other clustering comparison measures.

For future work, we would like to investigate generalizing ADCO for comparison of clusterings which may contain different numbers of clusters. This could facilitate the tighter integration of ADCO into areas such as ensemble and stream clustering. We would also like examine the use other cluster features for comparison, such as boundaries and centroids.

References

1. Ratanamahatana, C.: CloNI : Clustering of square root of N interval discretization. Data Mining IV, Info. and Comm. Tech. **29** (2003)
2. Karypis, G., Aggarwal, R., Kumar, V.: Multilevel hypergraph partitioning: application in VLSI domain. Ann. Conf. on Design Automation. (1997) 526–529
3. Larsen, B., Aone, C.: Fast and effective text mining using linear-time document clustering. Inter. Conf. on Knowledge Discovery and Data Mining. (1999)16–22
4. Hubert, L., Arabie, P.: Comparing partitions. Journ. of classification. (1985) 193–218
5. Aggarwal, C., Han, J., Wang, J., Yu, P.: A framework for clustering evolving data streams. 29th VLDB Conference. (2003)
6. Aggarwal, C.: A framework for diagnosing changes in evolving data streams. Intern. Conf. on Management of Data (2003) 575–586
7. Rand, W.: Objective criteria for the evaluation of clustering methods. Journ. of the American Statistical Association. **66** (1971) 846–850
8. Hamers, L., Hemeryck, Y.: Similarity measures in scientometric research: the Jaccard index vs. Salton's cosine formula. Info. Process. and Management. **25** (1989) 315–318
9. O'Callaghan, L., Mishra, N., Meyerson, A.: Streaming-Data Algorithms for High-Quality Clustering Intern. Conf. on Data Engineering. (2002)
10. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On Clustering Validation Techniques. Journ. of Intelligent Info. Sys. **17** (2001) 107–145
11. Fred, A., Jain A.: Combining Multiple Clusterings Using Evidence Accumulation. Transac. on Pattern Analysis and Machine Intelligence. **27** (2005) 835–850
12. Fred, A.: Finding consistent clusters in data partitions. Multiple Classifier Systems, Second International Workshop. (2001) 309–318
13. Fred, A., Jain, A.: Robust data clustering. Comp. Soc. Conf. on Computer Vision and Pattern Recognition. (2003) 128–133
14. Strehl, A., Ghosh, J.: Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions. Jour. on Machine Learning. **3** (2002) 583–617
15. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Academic Press. (1999)
16. Grossman, S.: Elementary Linear Algebra. Saunders College Publishing. (1994)
17. Meila, M.: Comparing Clusterings. Statistics Technical Report. http://www.stat.washington.edu/www/research/reports/2002/ (2005)
18. Meila, M.: Comparing Clusterings - An Axiomatic View. 22nd International Conference on Machine Learning. (2005)
19. Kleinberg, J.: An impossibility theorem for clustering. Conf. on Neural Information Processing Systems. (2002)
20. Zhou, D., Li, J., Zha, H.: A new Mallows distance based metric for comparing clusterings. Intern. Conf. on Machine Learning. (2005)