Comparing Clusterings in Space

Michael H. Coen MHCOEN@CS.WISC.EDU
M. Hidayath Ansari ANSARI@CS.WISC.EDU
Nathanael Fillmore NATHANAE@CS.WISC.EDU
University of Wisconsin-Madison, 1300 University Ave, Madison, WI 53706 USA

Abstract

This paper proposes a new method for comparing clusterings both partitionally and geometrically. Our approach is motivated by the following observation: the vast majority of previous techniques for comparing clusterings are entirely partitional, i.e., they examine assignments of points in set theoretic terms after they have been partitioned. In doing so, these methods ignore the spatial layout of the data, disregarding the fact that this information is responsible for generating the clusterings to begin with. We demonstrate that this leads to a variety of failure modes. Previous comparison techniques often fail to differentiate between significant changes made in data being clustered.

We formulate a new measure for comparing clusterings that combines spatial and partitional information into a single measure using optimization theory. Doing so eliminates pathological conditions in previous approaches. It also simultaneously removes common limitations, such as that each clustering must have the same number of clusters or they are over identical datasets. This approach is stable, easily implemented, and has strong intuitive appeal.

1. Introduction

This paper proposes a new method for comparing clusterings both partitionally and geometrically. We call our new comparison function clustering distance or *CDistance*. We believe this is the first principled approach for comparing clusterings according to their spatial properties as well as their cluster membership assignments.

We view a *clustering* as a partition of a set of points located in a space with an associated distance function. This view is natural, since popular clustering algorithms, e.g., k-means, spectral clustering, affinity propagation, etc., take as input not only a collection of points to be clustered but also a distance function on the space in which the points lie. This distance function may be specified implicitly and it may be transformed by a kernel, but it must be defined one way or another and its properties are crucial to a clustering algorithm's output.

In contrast, almost all existing clustering comparison techniques ignore the distances between points, treating clusterings as partitions of disembodied atoms. While this approach has merit under some circumstances, it seems surprising to ignore the distance func-

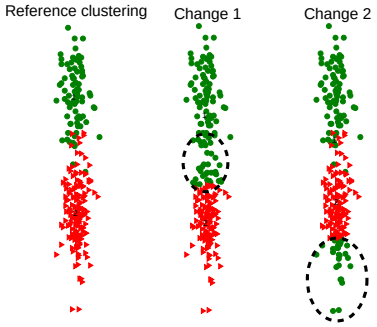


Figure 1. This figure displays three clusterings. Each one contains two clusters, whose members are indicated by green circles and red triangles. In the two changed clusterings, the circled points have been reassigned from the red to the green cluster. We might expect that the Reference Clustering is more similar to Change 1 than Change 2, because the modified points are *closer* to it. However, all of (Rand, 1971; Hubert & Arabie, 1985; van Dongen, 2000; Fowlkes & Mallows, 1983; Meilă, 2005; Zhou et al., 2005) are incapable of distinguishing between the two changes.

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010. Copyright 2010 by the author(s)/owner(s).

tion that was used to construct the clusterings. Doing so seems to discard what is in some sense the most basic information we have about them. Indeed, in Section 3, we exhibit a number of clusterings that have substantially different spatial properties but are indistinguishable by almost all previous clustering comparison techniques. One such example is presented in Figure 1. Only one of the existing clustering comparison techniques we have found can distinguish between the leftmost reference clustering and its two modifications to the right; we examine weaknesses in the one exception in Section 3.

The main contribution of this paper is a new technique for comparing clusterings that takes into account their spatial properties. In particular, our technique answers the question: *how well do two clusterings of points "overlap" in a given space?* Thus our technique does not only evaluate the assignments of points to partitions; it also takes into account the locations of the points in each cluster, the shapes of the clusters, and the spatial relations among the clusters.[1]

By incorporating spatial information, our approach provides several additional benefits. First, we are able to compare clusterings that cannot be considered by many other techniques; specifically, we can compare clusterings: (1) over different sets of points; (2) over different numbers of points; and (3) over different number of clusters. We know of no other clustering comparison technique that allows comparison under all such conditions simultaneously, particularly conditions (1) and (2), which are largely unaddressed in the literature. Finally, in contrast to some other approaches, our work also extends in a straightforward way to soft (non-partitional) clustering.

We briefly review some applications in which clustering distance is useful. (1) Because clustering is an unsupervised learning technique, comparing the output of clustering algorithms is difficult. In some cases, there may be a gold standard with which we would like our algorithm to agree. Measuring the distance between the gold standard clustering and an algorithm's output provides important insight into whether the algorithm is suitable for a given domain. (2) We can explore the stability of a clustering algorithm's results on a dataset by repeatedly subsampling the dataset and comparing the algorithm's results against each other. (3) If the outputs of two clustering algorithms tend to agree on certain kinds of data, we may prefer to use the algorithm with lower computational complexity; comparing the algorithms' outputs helps us make this determination. (4) Finally, in ensemble clustering, we may employ a variety of clustering algorithms that exploit different mathematical properties of the data. Asking if their outputs are both partitionally and geometrically compatible adds an extra dimension for comparison.

[1] Code implementing our approach and all data used in this paper are freely available at http://biocomp.wisc.edu/data.

2. Approach

In this section, we describe our new clustering comparison technique together with an algorithm for computing its value on a pair of clusterings. We begin by making precise the concept of a clustering and by describing some prerequisites that are needed to formulate our approach.

2.1. Clustering

A (hard) clustering $\mathcal{A}$ is a partition of a dataset D into sets $A_1, A_2, \ldots, A_K$ called clusters such that

$$A_i \cap A_j = \emptyset \text{ for all } i \neq j, \text{ and } \bigcup_{k=1}^{K} A_k = D.$$

We assume D, and therefore $A_1, \ldots, A_K$, are finite subsets of a metric space (Ω, d_Ω).

2.2. Optimal Transportation Distance

The optimal transportation problem (Hillier & Lieberman, 1995; Rachev & Ruschendorf, 1998) asks *what is the cheapest way to move a set of masses from sources to sinks, who are some distance away?* Here cost is defined as the total *mass* $\times$ *distance* moved. For example, one can think of the sources as factories and the sinks as warehouses to make the problem concrete. We assume that the sources are shipping exactly as much mass as the sinks are expecting.

Formally, in the optimal transportation problem, we are given weighted point sets (A, p) and (B, q), where $A = \{a_1, \ldots, a_{|A|}\}$ is a finite subset of the metric space (Ω, d_Ω), $p = (p_1, \ldots, p_{|A|})$ is a vector of associated nonnegative weights summing to one, and similar definitions hold for B and q.

The *optimal transportation distance* between (A, p) and (B, q) is defined as

$$d_{\mathrm{OT}}(A, B; p, q, d_\Omega) = \sum_{i=1}^{|A|} \sum_{j=1}^{|B|} f_{ij}^* \, d_\Omega(a_i, b_j),$$

where the optimal flow $F^* = (f_{ij}^*)$ between (A, p) and

(B, q) is the solution of the linear program

$$\text{minimize} \sum_{i=1}^{|A|} \sum_{j=1}^{|B|} f_{ij}\, d_\Omega(a_i, b_j) \text{ over } F = (f_{ij}) \text{ subj. to}$$

$$\begin{array}{rcll} f_{ij} & \geq & 0, & 1 \leq i \leq |A|,\ 1 \leq j \leq |B| \\ \sum_{j=1}^{|B|} f_{ij} & = & p_i, & 1 \leq i \leq |A| \\ \sum_{i=1}^{|A|} f_{ij} & = & q_j, & 1 \leq j \leq |B| \\ \sum_{i=1}^{|A|} \sum_{j=1}^{|B|} f_{ij} & = & 1. & \end{array}$$

It is useful to view the optimal flow as a representation of the *maximally cooperative* way to transport masses between sources and sinks. Here, cooperative means that the sources "agree" to transport their masses with a globally minimal cost.

2.3. Naive Transportation Distance

In contrast with the optimal algorithm proposed above, we define here a naive solution to the transportation problem. Here, the sources are all responsible for individually distributing their masses proportionally to the sinks. In this case, none of the sources cooperate, leading to inefficiency in shipping the overall mass to the sinks.

Formally, in the naive transportation problem, we are given weighted point sets (A, p) and (B, q) as above. We define the *naive transportation distance* between (A, p) and (B, q) as

$$d_{\mathrm{NT}}(A, B; p, q, d_\Omega) = \sum_{i=1}^{|A|} \sum_{j=1}^{|B|} p_i\, q_j\, d_\Omega(a_i, b_j).$$

2.4. Similarity Distance

Our next definition concerns the relationship between the optimal transportation distance and the naive transportation distance. Here we are interested in the *degree to which cooperation reduces the cost* of moving the source A onto the sink B.

Formally, we are given weighted point sets (A, p) and (B, q) as above. We define the *similarity distance* between (A, p) and (B, q) as the ratio

$$d_{\mathrm{S}}(A, B; p, q, d_\Omega) = \frac{d_{\mathrm{OT}}(A, B; p, q, d_\Omega)}{d_{\mathrm{NT}}(A, B; p, q, d_\Omega)}.$$

When A and B perfectly overlap, there is no cost to moving A onto B, so both the optimal transportation distance and the similarity distance between A and B are 0. On the other hand, when A and B are very distant from each other, each point in A is much closer to all other points in A than to any points in B, and vice-versa. Thus, in this case, cooperation does not yield any significant benefit, so d_{OT} is nearly as large as d_{NT}, and d_{S} is close to 1. (This behavior is illustrated in Figure 2.)

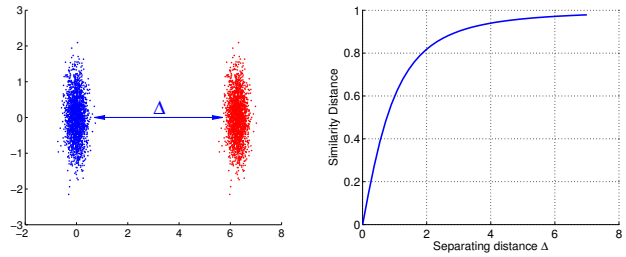


Figure 2. The plot on the right shows similarity distance as a function of separation distance between the two point sets displayed on the left. Similarity distance is zero when the point sets perfectly overlap and approaches one as the distance increases.

In this sense, the similarity distance between two point sets measures the degree to which they spatially overlap, or are "similar", to each other. (Note, the relationship is actually inverse; similarity distance is the degree of spatial overlap subtracted from one.) Another illustration of this behavior is given in Figure 3. Each panel in this figure shows two overlapping point sets to which we assign uniform weight distributions. The degree of spatial overlap of the two point sets in Example A is much higher than the degree of spatial overlap of the point sets in Example B, even though in absolute terms the amount of work required to optimally move one point set onto the other is much higher in Example A than in Example B.

2.5. Clustering Distance

We next define our new measure of clustering distance. Our goal is to construct a distance measure between clusterings that captures both spatial and partitional information about the clusterings.

Conceptually, our approach is as follows. Given a pair of clusterings $\mathcal{A}$ and $\mathcal{B}$, we first construct a *new* metric space—distinct from the metric space in which the original data lie—which contains one distinct element for each cluster in $\mathcal{A}$ and $\mathcal{B}$. We define the distance between any two elements of this new space to be the optimal transportation distance between the corresponding clusters. The clusterings $\mathcal{A}$ and $\mathcal{B}$ can now be thought of as weighted point sets in this new space, and the degree of similarity between $\mathcal{A}$ and $\mathcal{B}$ can be thought of as the degree of spatial overlap between the corresponding weighted point sets in this new space.

Formally, we are given clusterings $\mathcal{A}$ and $\mathcal{B}$ of datasets D and E in the metric space (Ω, d_Ω). We define the *clustering distance* (CDistance) between $\mathcal{A}$ and $\mathcal{B}$ as the quantity

$$d(\mathcal{A}, \mathcal{B}) = d_{\mathrm{S}}(\mathcal{A}, \mathcal{B}; \pi, \rho, d'_{\mathrm{OT}}),$$

where the weights $\pi = (|A|/|D| : A \in \mathcal{A})$ and $\rho = (|B|/|E| : B \in \mathcal{B})$ are proportional to the number of points in the clusters, and where the distance d'_{OT} between clusters $A \in \mathcal{A}$ and $B \in \mathcal{B}$ is the optimal transportation distance

$$d'_{\mathrm{OT}}(A, B) = d_{\mathrm{OT}}(A, B; p, q, d_\Omega)$$

with uniform weights $p = (1/|A|)$ and $q = (1/|B|)$.

An efficient algorithm to compute clustering distance is easily derived from the definition. Let $\mathcal{A}, \mathcal{B}, \pi, \rho, p$, and q be as above.

Step 1. For every pair of clusters $(A, B) \in \mathcal{A} \times \mathcal{B}$, compute the optimal transportation distance $d'_{\mathrm{OT}}(A, B) = d_{\mathrm{OT}}(A, B; p, q, d_\Omega)$ between A and B, based on the distances according to d_Ω between points in A and B.

Step 2. Compute the similarity distance $d_{\mathrm{S}}(\mathcal{A}, \mathcal{B}; \pi, \rho, d'_{\mathrm{OT}})$ between the clusterings $\mathcal{A}$ and $\mathcal{B}$, based on the optimal transportation distances d'_{OT} between clusters in $\mathcal{A}$ and $\mathcal{B}$ that were computed in Step 1. Call this quantity the clustering distance $d(\mathcal{A}, \mathcal{B})$ between the clusterings $\mathcal{A}$ and $\mathcal{B}$.

The reader will notice that we use optimal transportation distance to measure the distance between individual clusters (Step 1), while we use similarity distance to measure the distance between the clusterings as a whole (Step 2). The reason for this difference is as follows. In Step 1, we are interested in the *absolute* distances between clusters: we want to know how much work is needed to move all the points in one cluster onto the other cluster. We will use this as a measure of how "far" one cluster is from another. In contrast, in Step 2, we want to know the degree to which the clusters in one clustering spatially overlap with those in another clustering using the distances derived in Step 1.

Our choice to use uniform weights in Step 1 but proportional weights in Step 2 has a similar motivation. In a (hard) clustering, each point in a given cluster contributes as much to that cluster as any other point in the cluster contributes, so we weight points uniformly when comparing individual clusters. In contrast, Step 2 proportionally distributes the influence of each cluster in the overall computation of CDistance according to its relative weight, as determined by the number of data points it contains.

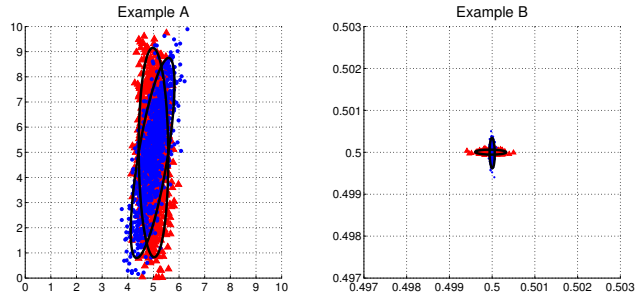


Figure 3. Similarity distance measures the degree of spatial overlap. We consider the two point sets in Example A to be far more similar to one another than those in Example B. This is the case even though they occupy far more area in absolute terms and would be deemed further apart by many distance metrics. Here, $d_{\mathrm{S}}(\text{Example A}) = 0.15$ and $d_{\mathrm{S}}(\text{Example B}) = 0.78$.

2.5.1. Computational Complexity

The complexity of the algorithm presented above for computing CDistance is dominated by the complexity of computing optimal transportation distance. Optimal transportation distance between two point sets of cardinality at most n can be computed in worst case $O(n^3 \log n)$ time (Shirdhonkar & Jacobs, 2008). Recently a number of linear or sublinear time approximation algorithms have been developed for several variants of the optimal transportation problem, e.g., (Shirdhonkar & Jacobs, 2008; Ba et al., 2009).

To verify that our method is efficient in practice, we tested our implementation for computing optimal transportation distance, which uses the transportation simplex algorithm, on several hundred thousand pairs of large point sets sampled from a variety of Gaussian distributions. The average running times fit the expression $(1.38 \times 10^{-7})n^{2.6} - 2.5$ seconds with an R^2 value of 1, where n is the cardinality of the larger of the two point sets being compared. For enormous point sets, one can employ standard binning techniques to further reduce the runtime (Levina & Bickel, 2001).

In contrast, the only other spatially aware clustering comparison method that we know of requires explicitly enumerating the exponential space of all possible permutations of matches between clusters in $\mathcal{A}$ to clusters in $\mathcal{B}$ (Bae et al., 2006).

3. Comparisons to Related Work

In Table 1, we present simple examples that demonstrate the insensitivity of popular comparison techniques to spatial differences between clusterings. Our approach is straightforward: for each technique, we

Table 1. Distances to Modified Clusterings. Each row in Table 1(a) depicts a dataset with points colored according to a reference clustering $\mathcal{R}$ (left column) and two different modifications of this clustering (center and right columns). For each example, Table 1(b) presents the distance between the reference clustering and each modification for the indicated clustering comparison techniques. The column labeled "?" indicates whether the technique provides sensible output.

In Examples 1–3, we only modify cluster assignments; the points are stationary. Since the pairwise relationships among the points change in the same way in each modification, only ADCO and CDistance detect that the modifications are not identical with respect to the reference.

In Example 4, the reference clustering is modified by moving three points of the bottom right cluster by small amounts. However, in Modification 1, the points do not move across a bin boundary, whereas in Modification 2, the points do move across a bin boundary. As a result, ADCO detects no change between Modification 1 and the reference clustering but detects a large change between Modification 2 and the reference clustering, even though the two modifications differ by only a small amount. CDistance correctly reports a similar change between Modification 1 and the reference and Modification 2 and the reference. (Since the points actually move, other clustering comparison techniques are not applicable to this example.)

Table 1(a)

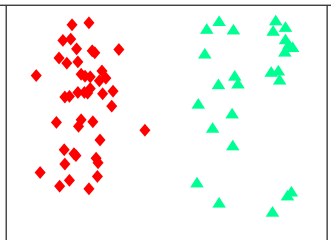
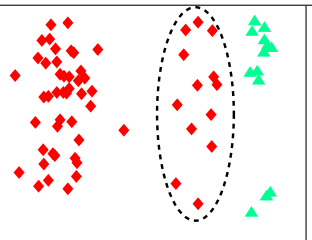
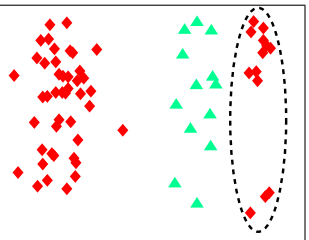
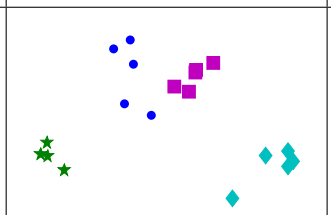
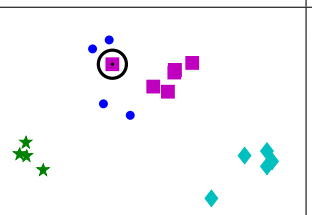
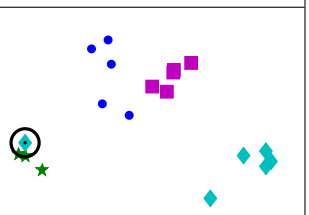
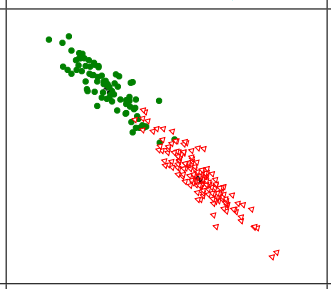
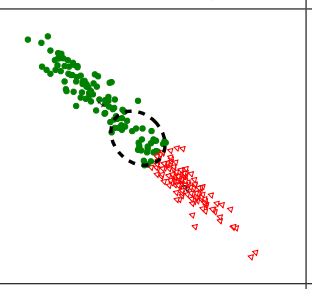
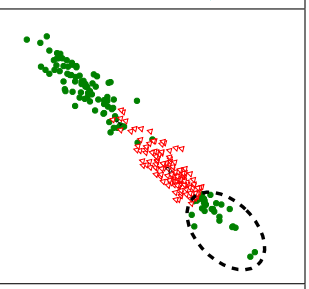
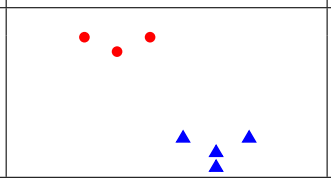
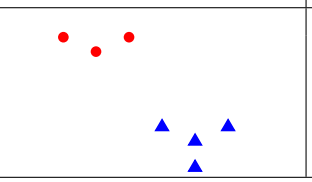
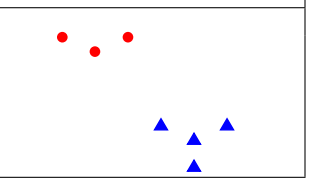
	Reference Clustering ($\mathcal{R}$)	Modification 1	Modification 2
Example 1			
Example 2			
Example 3			
Example 4			

Table 1(b)

Technique Name	Example 1			Example 2			Example 3			Example 4		
	$d(\mathcal{R},1)$	$d(\mathcal{R},2)$	?	$d(\mathcal{R},1)$	$d(\mathcal{R},2)$	?	$d(\mathcal{R},1)$	$d(\mathcal{R},2)$	?	$d(\mathcal{R},1)$	$d(\mathcal{R},2)$	?
Hubert	0.38	0.38	✗	0.04	0.04	✗	0.25	0.25	✗	N/A	N/A	✗
$1 - \text{Rand}$	0.00	0.00	✗	0.05	0.05	✗	0.00	0.00	✗	N/A	N/A	✗
van Dongen	0.18	0.18	✗	0.05	0.05	✗	0.13	0.13	✗	N/A	N/A	✗
VI	5.22	5.22	✗	11.31	11.31	✗	5.89	5.89	✗	N/A	N/A	✗
Zhou	8.24	8.24	✗	0.90	0.90	✗	10.0	10.0	✗	N/A	N/A	✗
ADCO	0.11	0.17	✓	0.02	0.04	✓	0.07	0.09	✓	0.00	0.07	✗
CDistance	**0.61**	**0.73**	✓	**0.06**	**0.18**	✓	**0.41**	**0.56**	✓	**0.08**	**0.09**	✓

present a baseline reference clustering. We then compare it to two other clusterings that it cannot distinguish, i.e., the comparison method assigns the same distance from the baseline to each of these clusterings. However, in each case, the clusterings compared to the baseline are significantly different from one another, both from a spatial, intuitive perspective and according to our measure, CDistance.

Our intent in Table 1 is to demonstrate that competing methods are unable to detect changes even when it appears clear that a non-trivial change has occurred. These examples are intended to be didactic, and as such, are composed of only enough points to be illustrative. Because of limitations in most of these approaches, all but one of the examples in Table 1 shows clusterings over identical sets of points. More complex datasets and additional discussion of other techniques for comparing clusterings are examined in Section 4.

We briefly review the techniques shown in Table 1. Among the earliest known methods for comparing clusterings is the Jaccard index (Ben-Hur et al., 2002). It measures the fraction of assignments on which different partitionings agree. The Rand index (Rand, 1971), among the best known of these techniques, is based on changes in point assignments. It is calculated by the fraction of points—taken pairwise—whose assignments are consistent between two clusterings. This approach has been built upon by many others. For example, (Hubert & Arabie, 1985) addresses the Rand index's well-known problem of overestimating similarity on randomly clustered datasets. These three measures are part of a general class of clustering comparison techniques based on tuple-counting or set-based membership. Other measures in this category include the Mirkin, van Dongen, Fowlkes-Mallows, and Wallace indices, a discussion of which can be found in (Ben-Hur et al., 2002; Meilă, 2005).

The Variation of Information approach was introduced by (Meilă, 2005), who defined an information theoretic metric between clusterings. It measures the information lost and gained in moving from one clustering to another, over the same set of points.

The work presented in (Zhou et al., 2005) shares our motivation of incorporating some notion of distance into comparing clusterings. However, the distance is computed over a space of indicator vectors for each cluster, representing whether each data point is a member of it. Thus, similarity over clusters is measured only in terms of their shared points' identities and does not take into account the spatial locations of these points.

Finally, we examine ADCO (Bae et al., 2006), which presents the only other method that directly employs spatial information. This approach bins all points and then determines the density of each cluster over the bins. The distance between two clusterings is defined as the minimal sum of pairwise cluster-density dot products (derived from the binning), taken over all possible permutations matching the clusters. This is in general not a feasible computation, as the number of bins grows exponentially with the dimensionality of the space, and more intractably, examining all matchings between clusters requires $O(n!)$ time, where n is the number of clusters.

4. Further Evaluation

In this section, we examine CDistance on a wider range of datasets. Our goal is to understand its dynamic behavior and how it may be used in ensemble methods and for evaluating stability.

4.1. Example Comparisons

Figure 4 illustrates some sample outputs of CDistance. In each subfigure, we are comparing two clusterings, one of which has been translated for visualization purposes. For example, in Figure 4(a), the two clusterings spatially overlap perfectly so their clustering distance is zero. Matching clusters are connected by lines to illustrate their correspondence. (These lines are drawn solely for visualization purposes.) The most interesting panel is Figure 4(d), which demonstrates that symmetries in a shape produce unstable clusterings with the algorithm used. Repeated applications of spectral clustering to this data produce very different results. Multiply clustering a dataset allows us to gauge whether an algorithm/dataset combination are mutually compatible.

4.2. Continuity and Stability of CDistance

Figure 5 illustrates the smoothness of CDistance as clusterings change in small increments. Figure 5(d) is a graph of the CDistance between a reference clustering, plotted in Figure 5(a), and rotations of the clustering by $0, \ldots, 2\pi$ radians.

Consider the clustering in Figure 6(a), which contains the same dataset as Example 4 from Table 1(a). Suppose we incrementally increase the y-coordinates of the cluster consisting of blue triangles. At each step, we compute both ADCO and CDistance between the modified dataset and the reference clustering in Figure 6(a). The result is plotted in Figure 6(b). We see that ADCO suffers from swings and discontinuities due

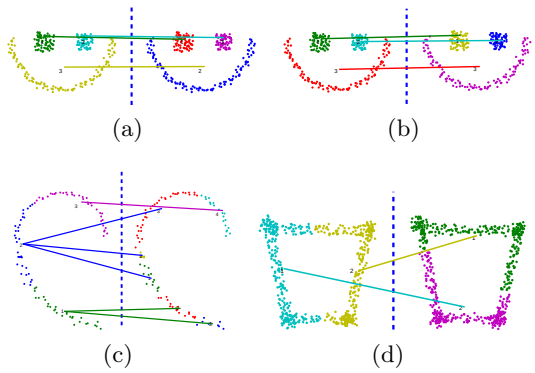


Figure 4. (a) Identical clusterings and identical datasets (see text for explanation of visualization). CDistance is 0. (b) Similar clusterings, but over slightly perturbed datasets. CDistance is 0.09. (c) Two different algorithms were used to cluster the same dataset. CDistance is 0.4, indicating a moderate mismatch of clusterings. (d) Two very different clusterings generated by spectral clustering over almost-identical datasets. CDistance is 0.90, suggesting instability in the clustering algorithm's output paired with this dataset.

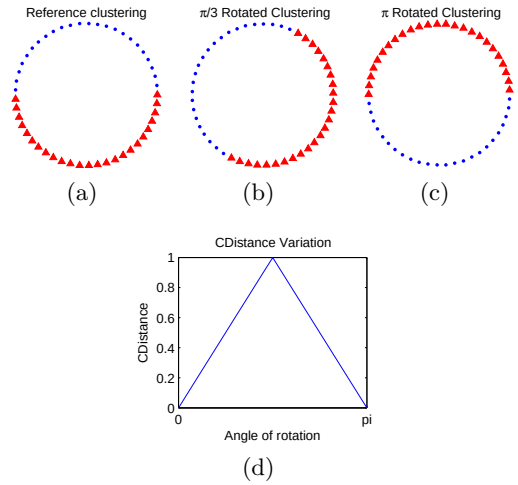


Figure 5. (a) Reference clustering. This dataset is a subset of the unit circle with geodesic distance function. (b) An intermediate clustering. (c) The completely rotated clustering; CDistance between this clustering and the reference clustering is 0. (d) The graph of variation of CDistance with angle of rotation is linear, as it should be.

to the abrupt transition of data moving between discrete bins. As a result of this behavior, ADCO's values are difficult to interpret intuitively.

4.3. Subsampling and Stability

In this subsection, we illustrate how CDistance can be used to compare clusterings from subsampled datasets. Ben-Hur et al. (2002) proposed that the stability of a clustering could be determined by repeatedly clustering subsamples of its dataset. Finding consistent high similarity across clusterings indicates consistency in locating similar substructures in the dataset. This can increase confidence in the applicability of an algorithm to a particular distribution of data. In other words, by comparing the resultant clusterings, one can obtain a goodness-of-fit between a dataset and a clustering algorithm. The clustering comparisons in Ben-Hur et al. (2002) were all via partitional methods.

We can instead use CDistance to perform this comparison. In addition, CDistance enables us to formulate a new type of ensemble clustering using this methodology. This is depicted in Figure 7, where we repeatedly cluster subsamples of a dataset with both Self-Tuning Spectral Clustering (Zelnik-Manor & Perona, 2004) and Affinity Propagation (Dueck & Frey, 2007). Although these algorithms rely on mathematically distinct properties, we see their resultant clusterings on subsampled data agree to a surprising extent according to CDistance.

Because CDistance is able to compare clusterings of different cardinality, we can use it with algorithms that self-determine how many clusters to generate. Thus, we can use a wider assortment of clustering algorithms in ensemble methods and for stability testing. The intuition behind CDistance is reflected in the spatial overlap between connected clusters (across clusterings) in the figures.

5. Discussion

This paper has presented a new method for comparing clusterings that incorporates both spatial and categorical information into a single distance function that we call CDistance. Our method is unique in enabling comparisons between clusterings that differ in their

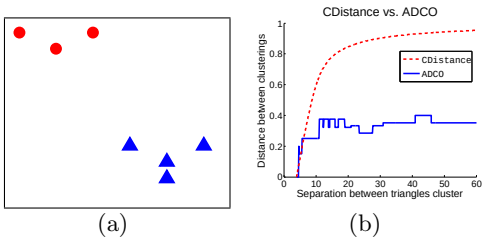


Figure 6. Examining the smoothness of CDistance compared to ADCO. (a) Our reference clustering; (b) plotting the changes in CDistance and ADCO as a function of moving the "blue triangle" cluster upwards.

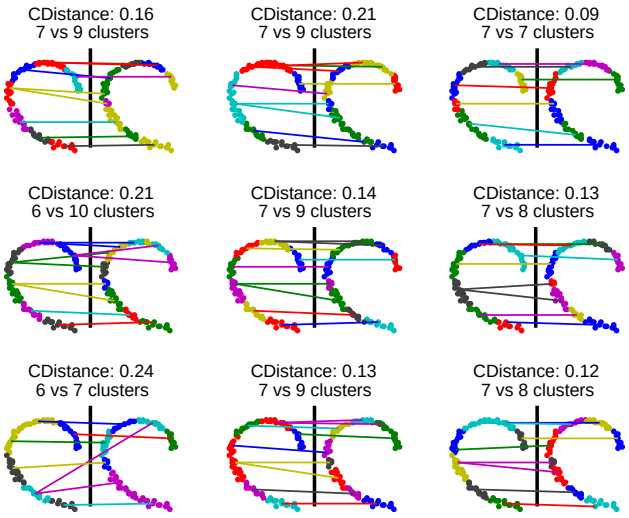


Figure 7. Comparing outputs of different clustering algorithms over randomly sampled subsets of a dataset. The low values of CDistance indicate relative similarity and stability between the clusterings. The numbers of clusters involved in each comparison are displayed below CDistance value.

datasets, number of points, and number of clusters. This significantly broadens the range of applications of our measure in comparison with other approaches to comparing clusterings. Our approach is extensible to comparing soft clusterings (such as those generated by Expectation-Maximization techniques and other probabilistic methods) by replacing the uniform weight distributions in Step 1 with distributions describing fractional cluster memberships.

One component of our approach, similarity distance, has also proved useful in a variety of clustering problems, e.g., in learning the vowel structure of an unknown language (Coen, 2006). It comes as little surprise that exploiting spatial overlap is as useful while clustering as it is in comparing clusterings.

Acknowledgements

This work was supported by the School of Medicine and Public Health, the Wisconsin Alumni Research Foundation, the Department of Biostatistics and Medical Informatics, and the Department of Computer Sciences at the University of Wisconsin-Madison. Thanks to Jordan Ellenberg and our anonymous reviewers for insightful discussion.

References

Ba, Khanh Do, Nguyen, Huy L., Nguyen, Huy N., and Rubinfeld, Ronitt. Sublinear time algorithms for earth mover's distance. *arXiv abs/0904.0292*, 2009.

Bae, Eric, Bailey, James, and Dong, Guozhu. Clustering similarity comparison using density profiles. In *Proc. of the 19th Australian Joint Conf. on Artificial Intelligence*, pp. 342–351. Springer LNCS, 2006.

Ben-Hur, A., Elisseeff, A., and Guyon, I. A stability based method for discovering structure in clustered data. In *Pacific Symp. on Biocomputing*, 2002.

Coen, Michael H. Self-supervised acquisition of vowels in american english. In *AAAI*, 2006.

Dueck, D. and Frey, B, J. Non-metric affinity propagation for unsupervised image categorization. *IEEE ICCV*, 0:1–8, 2007.

Fowlkes, E.B. and Mallows, C.L. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78:553–569, 1983.

Hillier, F.S. and Lieberman, G.J. *Introduction to Mathematical Programming*. McGraw-Hill, 1995.

Hubert, L. and Arabie, P. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

Levina, E. and Bickel, P. The earth mover's distance is the mallows distance: Some insights from statistics. In *ICCV*, 2001.

Meilă, Marina. Comparing clusterings: an axiomatic view. In *ICML*, 2005.

Rachev, S. T. and Ruschendorf, L. *Mass Transportation Problems: Volume I: Theory (Probability and its Applications)*. Springer, 1998.

Rand, W. M. Objective criteria for the evaluation of clustering methods. *J. Amer. Statist. Assoc.*, pp. 846–850, 1971.

Shirdhonkar, S. and Jacobs, D.W. Approximate earth movers distance in linear time. In *CVPR*, 2008.

van Dongen, S. Performance criteria for graph clustering and markov cluster experiments. Technical report, National Research Institute for Mathematics and Computer Science in the Netherlands, 2000.

Zelnik-Manor, L and Perona, P. Self-tuning spectral clustering. In *NIPS*, 2004.

Zhou, D., Li, J., and Zha, H. A new mallows distance based metric for comparing clusterings. In *ICML*, 2005.