

Learning from Spatial Overlap

Michael H. Coen^{1,2}, M. Hidayath Ansari¹, and Nathanael Fillmore¹

¹Dept. of Computer Sciences

²Dept. of Biostatistics and Medical Informatics

University of Wisconsin-Madison

Madison, WI 53706, USA

Abstract

This paper explores a new measure of similarity between point sets in arbitrary metric spaces. The measure is based on the spatial overlap of the shapes and densities of these point sets. It is applicable in any domain where point sets are a natural representation for data. Specifically, we show examples of its use in natural language processing, object recognition in images, and multidimensional point set classification. We provide a geometric interpretation of this measure and show that it is well-motivated, intuitive, parameter-free, and straightforward to use. We further demonstrate that it is computationally tractable and applicable to both supervised and unsupervised learning problems.

1 Introduction

What does it mean for two things to be *similar*? This type of question is commonplace in computational sciences but its interpretation varies widely. For example, we may represent proteins, documents, movies, and images as collections of atoms, words, reviews, and edges respectively. For each of these representations, we often want to find distance measures that enable meaningful comparisons between sample instances.

Our contribution in this paper is to formulate and examine a new measure, *similarity distance*, that provides an intuitive basis for understanding such comparisons. In this paper, our *things* are finite, weighted point sets of varying cardinality. The notion of *similarity* presented here refers to *a measure of the spatial overlap* between these point sets. Namely, when we consider the similarity of two objects, we are asking: to what degree do their point set representations occupy the same region in a metric space? The goal of this paper is to formalize and answer this question; to compare our solution to other approaches; and to demonstrate its utility in solving real-world problems.

It is easiest to begin with an intuitive, visual presentation of the problem and definition.

1.1 Problem Statement

In this paper, we focus on the concept of spatial overlap as a measure of similarity. In other words, we would like to

Copyright © 2011, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

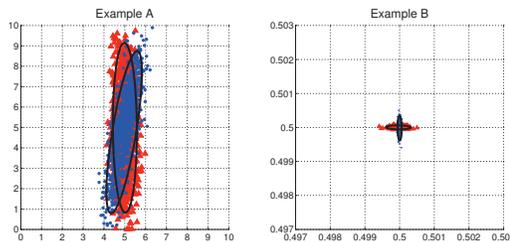


Figure 1: We consider the two point sets in Example A to be far more similar to one another than those in Example B. This is the case even though they occupy far more area in absolute terms and would be deemed further apart by many distance metrics.

define a distance function with a range over $[0, 1]$, where a value of 0 means two point sets perfectly overlap and a value near 1 means they occupy extremely different regions of space. To turn this into a similarity function (instead of a distance) we simply subtract the distance from 1. We make no assumptions about the cardinality of each set or how they were generated. Nor do we care about the sizes of the regions of space involved, e.g., the hyper-volumes of their convex hulls.

An image is useful for illustrating this idea. Consider the two examples in Figure 1. Each shows two overlapping samples (shown in red and blue respectively) drawn from Gaussian distributions; we would like to compare the similarity of these samples, each of which is commonly called a point set. Our intention is that the point sets in Example A should be judged much more similar than those in Example B based on their degree of spatial overlap, despite the points in Example A covering orders of magnitude more area than those in Example B. We discuss the relationship between similarity and distance below, but we note that the relatively tiny distances involved in Example B would lead many distance metrics to indicate they are “closer” to one another; this is the opposite of what we would like to find.

2 Similarity Distance

Similarity distance (d_s) is derived from the Kantorovich-Wasserstein metric (d_{kw}) (Kantorovich 1942; Deza and Deza 2009), which proposed a solution to the Transportation

Problem posed by Monge in 1781. This problem may be stated: *What is the optimal way to move a set of masses from suppliers to receivers, who are some distance away?* Optimal in this definition means minimizing the amount of total work performed, where work is defined as $mass \times distance$. For example, we might imagine a set of factories that stock a set of warehouses, and we would like to situate them to minimize the amount of driving necessary between the two. This problem has been rediscovered in many guises, most recently in a modified form as the Earth Mover’s Distance (Rubner, Tomasi, and Guibas 2000), which has become popular in computer vision.

It is useful to view the Kantorovich-Wasserstein distance as 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. In other words, they communicate to determine how to minimize the amount of shipping required.

Let us contrast this optimal view with the notion that each source delivers its mass to all sinks independently of any other sources, in proportion to its production. We will call this *naive transportation distance* (d_{NT}). In other words, the sources do not communicate. Each simply makes its own deliveries to every sink proportionally. Note this is *not* the worst (i.e., most inefficient) transportation schema. It is simply what occurs if the sources are oblivious to one another - when they do not take advantage of the potential savings that could be gained by cooperation.

2.1 Preliminary Definitions

We define a weighted point set A as finite collection of points $\{a_i \in \mathcal{X}\}$, where each point has an associated weight $\omega_i \in [0, 1]$, such that $\sum_i \omega_i = 1$. Thus, ω corresponds to a discrete probability distribution over some domain, for example, \mathcal{X} could be \mathbb{R}^d .

The similarity distance $d_s(A, B)$ between two such point sets A and B is simply the *ratio* of these two metrics, namely:

$$d_s(A, B) = \frac{d_{KW}(A, B)}{d_{NT}(A, B)}. \quad (1)$$

By this definition, $d_s(A, B)$ measures the *optimization gained by adding cooperation* when moving the source A onto the sink B .¹ Thus, it is a dimensionless quantity that ranges between zero and one. For clarity, let us examine d_s at its two extremes. If $d_s(A, B) = 0$, then $d_{KW}(A, B) = 0$, implying the maximally cooperative distance between A and B is zero. This can occur only when $A = B$; namely they perfectly overlap; this means each source is co-located with a sink expecting precisely as much mass as it produces.

In contrast, suppose $d_s(A, B) \rightarrow 1$. This tells us that cooperation does not help during transportation. This occurs when A and B are so far apart that the points in A are much closer to other points in A than those in B and vice-versa. Thus, cooperation does not yield any significant benefit. In this case, $d_{KW}(A, B) \rightarrow d_{NT}(A, B)$, implying $d_s(A, B) \rightarrow 1$.

¹Note that $d_{NT} = 0$ iff both point sets contain exactly the same singleton point. In this case, d_s is undefined.

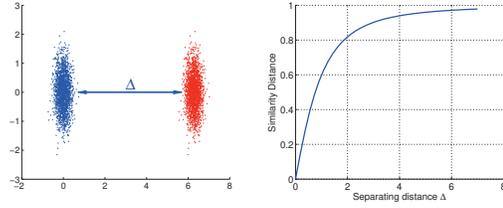


Figure 2: The graph on the right plots similarity distance as a function of separation distance between the two point sets shown on the left. As can be seen, similarity distance grows non-linearly as the distance between the point sets increases and then quickly approaches its asymptotic limit of 1.

As $d_{NT}(A, B) \geq d_{KW}(A, B)$ by definition, this provides the upper bound for $d_s(A, B)$ of 1. We see this in Figure 2, where the similarity distance between the two illustrated point sets quickly approaches 1 as they are separated. Conversely, as the point sets increasingly overlap, their similarity distance approaches zero rapidly.²

2.2 Formal Definitions

Kantorovich-Wasserstein Distance The discrete formulation of d_{KW} is easily obtained through the discrete version of the Mallow’s Distance (Levina and Bickel 2001). The optimization problem for computing $d_{KW}(A, B)$ thus corresponds to the following minimization problem:

Consider two point sets $A = \{a_1, \dots, a_m\}$, with associated weights p_i and $B = \{b_1, \dots, b_n\}$, with associated weights q_j , with both sets of weights summing to one. Treating A and B as random variables taking values $\{a_i\}$ and $\{b_j\}$ with probabilities $\{p_i\}$ and $\{q_j\}$ respectively, d_{KW} is obtained by minimizing the expected distance between A and B over all joint distributions $F = (f_{ij})$ of A and B :

$$E_F \|A - B\| = \sum_{i=1}^m \sum_{j=1}^n f_{ij} \|a_i - b_j\|_2 = \sum_{i=1}^m \sum_{j=1}^n f_{ij} d_{ij}$$

where F is subject to:

$$f_{ij} \geq 0, 1 \leq i \leq m, 1 \leq j \leq n \quad (2)$$

$$\sum_{j=1}^n f_{ij} = p_i, 1 \leq i \leq m \quad (3)$$

$$\sum_{i=1}^m f_{ij} = q_j, 1 \leq j \leq n \quad (4)$$

$$\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \sum_{i=1}^m p_i = \sum_{j=1}^n q_j = 1 \quad (5)$$

Once so formulated, this optimization problem may be solved using the transportation simplex algorithm. Although this algorithm is known to have exponential worst case runtime, it is remarkably efficient on most inputs and therefore

²Code implementing our approach and all data used in this paper are freely available at <http://biocomp.wisc.edu/data>.

widely used. We discuss runtime complexity and an approximation technique for enormous point sets in sections 2.3 and 2.4.

Naive Transportation Distance We now define a naive solution to the transportation problem. Here, each “supply” point is individually responsible for delivering its mass proportionally to each “receiving” point. In this instance, none of the shippers cooperate, leading to inefficiency in shipping the overall mass from one probability distribution to the other.

Over weighted point sets corresponding to discrete distributions, we define naive transportation distance d_{NT} as:

$$d_{NT}(A, B) = \sum_{i=1}^m p_i d_{KW}(\{a_i\}, B) = \sum_{i=1}^m \sum_{j=1}^n p_i q_j d(a_i, b_j) \quad (6)$$

The naive distance is therefore the weighted sum of the “ground” distances d between individual points. It is straightforward to directly calculate d_{NT} in $O(k^2)$ time, where $k = \max(m, n)$.

2.3 Computational complexity

The complexity of computing similarity distance is dominated by computation of the Kantorovich-Wasserstein distance d_{KW} , which is a well-studied problem; using the Hungarian method has worst case complexity $O(n^3)$ (Li 2010) in unrestricted metric spaces. Recently a number of linear or sublinear time approximation algorithms have been developed for this problem and several variations, e.g., (Li 2010; Ba et al. 2009). We have tested our implementation, which uses the transportation simplex algorithm, over several hundred thousand pairs of point sets drawn from standard statistical distributions and real world data sets. The runtime has expected time complexity of $(1.38 \times 10^{-7})n^{2.6}$ seconds, fit with an R^2 value of 1, where n is the size of the larger of the two point sets being compared. (We are particular to provide the quadratic coefficient, rather than describe the runtime using order notation, as its small value is what allows this approach to be used on larger scale problems.)

2.4 Hyperclustering

Because similarity distance measures the relative density differences between point sets, it is not overly sensitive to their exact numbers or locations. We use this intuition to approximate similarity distance by grouping nearby points into a single weighted point.

We call these groups of nearby points “hyperclusters” and construct them by recursively splitting the original point sets via k-means (with random initialization) until the maximum interpoint distance within each hypercluster is less than a specified threshold. In Figure 3, we show how the error and runtime change for a pair of point sets as the number of hyperclusters change. Empirically, this technique allows similarity distance to be approximated closely for sets of millions of points. For example, precisely computing similarity distance for point sets of size 100,000 would take almost 16 days, but an approximate answer can be computed in 46.9

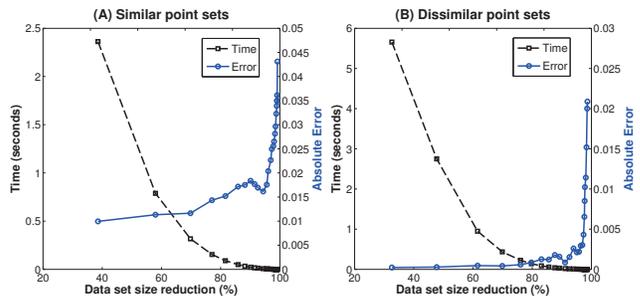


Figure 3: Error in similarity distance when approximated by hyperclustering, averaged over 30 runs. (A) Here, we sample two sets of size 1000 from the same distribution. Their exact similarity distance is 0.108, which takes 14.3 seconds to compute precisely. We vary the number of hyperclusters, corresponding to a reduction in problem size, and plot the error and overall computation time. (B) Here, we sample two sets of size 100 from poorly-overlapping distributions. The actual similarity distance is 0.879, which takes 16.12 seconds to compute precisely. Note in both cases there is negligible loss in accuracy even when the point set size is reduced by up to 80%.

seconds to within 0.01 of the true value.³ In extensive experimentation with this approximate form of similarity distance, errors of up to 0.05 have little effect and correspond to natural variation in samples drawn from the same distribution.

3 Related Work

Prior work on quantifying similarity between point sets or measuring a distance between them generally falls within one of three categories, most of which are not specifically designed to measure overlap or similarity. These measures are also very sensitive to their parameters, which often requires extensive search for a given problem, making their use problematic in unsupervised learning problems.

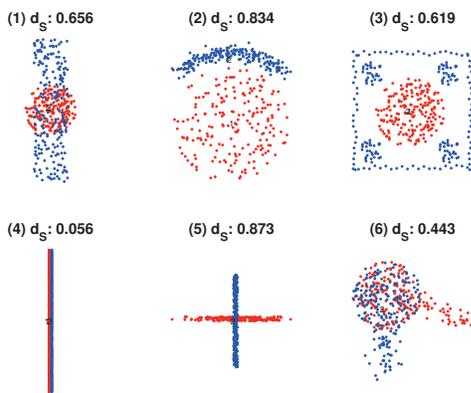
3.1 Point-set Distance extensions

The first set of approaches are inspired from point-set and Hausdorff distances. Point-set distance is defined between a single point x and a set of points A as $\inf_{y \in A} d(x, y)$. Hausdorff distance is an extension of this concept. The directed Hausdorff distance $D_{Haus}(A, B)$ between two sets of points A and B is $\sup_{x \in A} \inf_{y \in B} d(x, y)$ and the Hausdorff distance between sets A and B is the larger of $D_{Haus}(A, B)$ and $D_{Haus}(B, A)$. Other metrics inspired from point-set distance are discussed in Deza and Deza (2009). We discuss this class of distances further in section 3.4.

3.2 Root Mean Square Distance

A second method of computing distances between point sets is to assume an order between the points in them and align

³We determined this by solving similarity distance analytically for several common distributions, thereby providing a way to evaluate approximations.



Panel	1	2	3	4	5	6	ρ
d_{KW}	0.337	0.337	0.337	0.337	0.337	0.337	0.000
Lin. rescaling	0.205	0.478	0.325	0.871	0.282	0.162	0.097
Mean-var norm.	0.914	1.698	1.176	1.991	1.599	0.801	-0.108
Rank norm.	0.205	0.501	0.320	0.500	0.339	0.182	-0.053
d_S	0.656	0.834	0.619	0.056	0.873	0.443	1.000

Figure 4: All six examples in this figure were constructed to have the same d_{KW} and Earth Mover’s Distance ($= .337$), between the blue and red point sets, while having markedly different spatial properties from each other. This is reflected in their similarity distances, as shown above each example. The table further illustrates that one cannot simply normalize d_{KW} to obtain the measure provided by d_S . The final column shows Pearson correlation coefficients of each normalization with similarity distance, demonstrating that none of them capture the notion of spatial overlap.

them using an algorithm such as Kabsch (1976) or Procrustes (Goodall 1991). Once an alignment is found, a distortion measure (such as least root mean square distance) can be calculated by summing up distances between corresponding pairs of points. Clearly this method can only work for point sets of the same cardinality and is susceptible to disproportionate influence by outlying points. While modifications exist to overcome these problems, these general methods of summing distances between pairs of points yield little information about similarity or shape congruence.

3.3 Match Kernels

Pyramid match (Grauman and Darrell 2007) and other match kernels have been developed as efficient ways to determine similarity between point sets especially with vision applications in mind. The focus in match kernels however is to find similarity *while not penalizing non-similarity*. These kernels find closest pairs among individual points and only take into account these pairs for the kernel computation. Thus, such methods do not capture a notion of the “shape” of the point sets, but instead only their intersection, regardless of the importance of their non-overlap.

3.4 Others

Kondor and Jebara (2003) propose a kernel which takes into account the density of point sets. They require a Gaussian

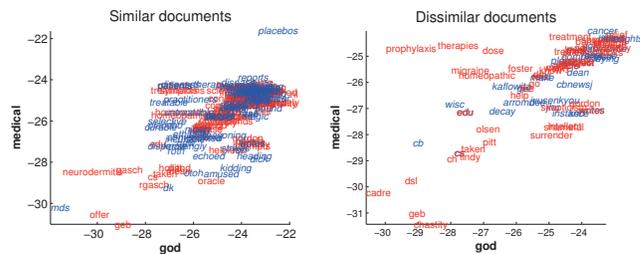


Figure 5: In the example above, point sets corresponding to two documents are plotted in the semantic subspace defined by *god* and *medical*. In each plot, one document is displayed in a blue italic font and the other is displayed in a red non-italic font. On the left, the two documents are from the same newsgroups whereas on the right they are from different newsgroups. Similarity distance captures the intuitive notion of spatial overlap corresponding to these classifications. (Only two of 6 dimensions are visualized here.)

distribution to be fit to each point set (or another distribution) and define a kernel based on a probabilistic divergence measure such as Bhattacharyya distance. This approach is further kernelized by mapping the elements of each point set to a new Hilbert space before fitting the parametric model. The two main issues with this approach are that it assumes a fixed distribution and is quite inefficient due to expensive computations involving matrix multiplications, inverses and determinants.

Many of the approaches mentioned above are lossy in the sense they rely on only some of the pairwise interactions between points. In doing so, they collapse the problem into calculating the distances between small sets of the original points. However, this provides little information about how similar the overall shapes of the entire points sets are. They are also neither bounded nor scale-invariant, making absolute judgments of similarity difficult.

3.5 Normalizations of d_{KW}

Another possible approach to measuring similarity between two point sets would be to first normalize the point sets and then apply the Kantorovich-Wasserstein distance. Natural examples of normalization schemes include linear scaling, mean-variance normalization, and rank normalization (Stolcke, Kajarekar, and Ferrer 2008). These normalizations can be useful in various circumstances, but Figure 4 and the Pearson correlation coefficients in the table show that they do not capture any notion of spatial overlap.

4 Applications

In this section we examine applications of similarity distance used in isolation and as a kernel to a variety of supervised and unsupervised machine learning problems.

4.1 Document Classification

By modeling the *topic* of a document as a *shape*, we can use similarity distance for text classification. We demonstrate this using the 20 Newsgroups dataset (Lang 1995) as

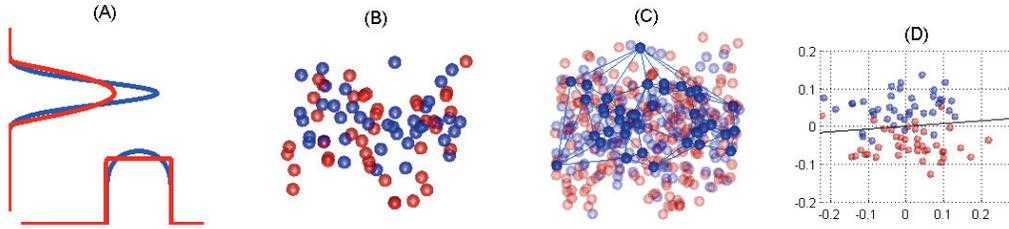


Figure 6: (A) Probability density function plots along each dimension for point sets sampled in Experiment 2. (B) Examples of two point sets that are sampled from distributions shown in red and blue in (A). (C) All points in all point sets used in Experiment 2. The points in one example point set are connected with lines. (D) This panel shows the same data as (C), but in a 2-D space reconstructed via approximate multidimensional scaling using pairwise similarity distances between the point sets. Each point in this panel thus represents an entire point set from the original data, and distances between points in this panel correspond to similarity distances between the point sets they represent. The separating line is an imaginary separator that a support vector machine might create using a kernel based on similarity distance.

	Classifier	Accuracy	Precision	Recall	F-Measure
Baseline (bag-of-words)	C4.5 (J48)	73.33%	0.763	0.733	0.726
	Naive Bayes	75.00%	0.789	0.750	0.741
	Random forest	78.33%	0.784	0.783	0.783
	SVM (RBF kernel)	76.67%	0.800	0.767	0.760
	SVM (polynomial kernel)	83.33%	0.847	0.833	0.832
Semantic space	SVM (Pyramid match kernel)	75.36%	0.742	0.719	0.730
	1-nearest neighbor (d_S)	85.00%	0.860	0.850	0.849
	2-, 3-, 4-nearest neighbor (d_S)	85.00%	0.854	0.850	0.850
	5-nearest neighbor (d_S)	81.67%	0.835	0.817	0.814
	SVM ($1 - d_S$) kernel	92.75%	0.909	0.938	0.923

Table 1: Results of text experiment using 10-fold cross validation. Results from our approach are shown in red, the best of which is in bold face. See text for details.

a testbed. The task here is to determine which of two newsgroups a given message came from. We do this by mapping the words in each message to points in a “semantic space” so that similar sets of words (documents) have similar shapes (See Pado and Lapata (2007) for an overview of work on semantic spaces). The basis for this space is chosen by selecting a set of reference words occurring in documents that have high mutual information for predicting the source newsgroups. Each word is mapped to a vector consisting of its similarities with each of these reference words, with similarity between two words being defined by their pointwise mutual information (PMI) (Terra and Clarke 2003). We estimate these PMIs using ratios of the number of hits reported by Google for individual words and pairs of words (Turney and Littman 2005). This construction has a distinct advantage compared to the standard bag-of-words approach because it makes use of semantic relations between words.

For our experiment we chose 30 articles at random from each of two newsgroups: alt.atheism and sci.med, and selected 6 reference words: (*christian, doctor, god, medical, say, atheists*). We mapped each word to a vector in \mathbb{R}^6 as follows:

$$f(w) = (\text{PMI}(\text{christian}, w), \dots, \text{PMI}(\text{atheists}, w))$$

We performed classification using k -nearest neighbors (k NN) and support vector machines using pyramid match kernel and a kernel derived from similarity distance ($1 - d_S$)

to compare documents. To establish a baseline we also used C4.5, Naive Bayes, random forest and SVMs with common kernels on indicator bag-of-words vectors. Classification metrics in Table 1 show that similarity distance is able to exploit semantic relationships between words (reflected by their mutual information) to successfully classify samples in this experiment. Additionally, similarity distance provides an easy way to visualize and understand the results, something which is uncommon in many classification tasks; an example is shown in Figure 5.

4.2 Object Recognition in Images

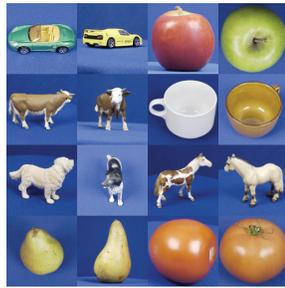
We applied similarity distance to an image classification task on a subset of the publicly available ETH-80 dataset (Leibe and Schiele 2003), using the data and experimental setup of (Grauman and Darrell 2007). The subset contains 5 views of each object in the database. Our experiment used a total of 256 descriptors in 128 dimensions per image. We trained an SVM classifier using a variety of kernels on the following problem: how well can the category of a holdout object be identified after training on the rest of the data including other instances of objects from that category? Accuracy results are shown in Figure 7.

4.3 Classification of Synthetic Data

A common assumption in machine learning is that data from different classes come from different underlying distributions. It may be the case that instances come in “bags” of points from the same distribution (for example multiple observations at a single time point). We simulate an example where we sample sets of points from two different multivariate statistical distributions and see how well similarity distance can classify instances.

The density functions in one dimension of both distributions are plotted along the x -axis, and along the other dimension on the y -axis in Figure 6.

We sample point sets from each distribution with varying numbers of points and train an SVM to separate between them using Pyramid Match and $(1 - d_S)$ as similarity functions. Similarity distance was able to achieve a 79.4% 10-



Algorithm	Accuracy
Similarity distance kernel	94%
Match Kernel (Wallraven et al. 2003)	90%
Pyramid Match Kernel	89%

Figure 7: Example images and classification results from the ETH-80 dataset. Two instances from the 8 classes are shown.

fold cross-validated classification accuracy, whereas Pyramid Match achieved an accuracy of 68.3%. Note that point sets from each class appear very similar (an example from each class is shown in Figure 6(B)), and it is the relative density at various locations that separates them. In this particular case, the means and variances of the two distributions are nearly identical along each dimension.

4.4 Clustering

Similarity distance has been used in clustering (Coen 2005; 2006), e.g., in learning the vowel structure of an unknown language, and in comparing different clusterings (Coen, Ansari, and Fillmore 2010). In the latter of these, set theoretic approaches have long dominated partitionial analyses of cluster assignments. Similarity distance lets us compare clusterings spatially in terms of their actual geometric arrangements in addition to their category assignments.

5 Conclusion

This paper has formally examined a new measure of similarity between point sets that is based on their spatial overlap. It captures an inherent mathematical property between the datasets that has strong intuitive appeal. In measuring overlap, it takes no parameters, making it suited for both supervised and unsupervised learning problems. Its spatial dependence also suggests how to approach various problems, i.e., by mapping instances to shapes that can be distinguished. Thus, it is well-suited to problems that can be viewed spatially and has a number of surprising mathematical properties that we are currently investigating.

6 Acknowledgments

This work has been supported by the Department of Biostatistics and Medical Informatics, the Department of Computer Sciences, the Wisconsin Alumni Research Foundation, the Vilas Trust, and the School of Medicine and Public Health, at the University of Wisconsin-Madison. The authors thank Grace Wahba for helpful discussion and the anonymous reviewers for their comments.

References

- Ba, K. D.; Nguyen, H. L.; Nguyen, H. N.; and Rubinfeld, R. 2009. Sublinear time algorithms for earth mover’s distance. *arXiv abs/0904.0292*.
- Coen, M. H.; Ansari, M. H.; and Fillmore, N. 2010. Comparing clusterings in space. In *ICML 2010: Proceedings of the 27th International Conference on Machine Learning*.
- Coen, M. H. 2005. Cross-modal clustering. In *AAAI’05: Proceedings of the 20th National Conference on Artificial Intelligence*, 932–937. AAAI Press.
- Coen, M. H. 2006. Self-supervised acquisition of vowels in American English. In *AAAI’06: Proceedings of the 21st National Conference on Artificial Intelligence*, 1451–1456. AAAI Press.
- Deza, M. M., and Deza, E. 2009. *Encyclopedia of Distances*. Springer.
- Goodall, C. 1991. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society. Series B (Methodological)* 53(2):285–339.
- Grauman, K., and Darrell, T. 2007. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research* 8.
- Kantorovich, L. V. 1942. On the transfer of masses. *Dokl Akad Nauk* 37. Translated in *Management Science* (1959) 4:1–4.
- Lang, K. 1995. Newsweeder: Learning to filter netnews. In *ICML 1995: Proceedings of the 12th International Conference on Machine Learning*, 331–339.
- Leibe, B., and Schiele, B. 2003. Analyzing appearance and contour based methods for object categorization. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, II – 409–15.
- Levina, E., and Bickel, P. 2001. The Earth Mover’s Distance is the Mallows Distance: Some insights from statistics. *IEEE International Conference on Computer Vision* 2:251.
- Li, S. 2010. On constant factor approximation for earth mover distance over doubling metrics. *arXiv abs/1002.4034*.
- Pado, S., and Lapata, M. 2007. Dependency-based construction of semantic space models. *Computational Linguistics* 33(2):161–199.
- Rubner, Y.; Tomasi, C.; and Guibas, L. 2000. The Earth Mover’s Distance as a metric for image retrieval. *International Journal of Computer Vision* 40(2):99–121.
- Stolcke, A.; Kajarekar, S.; and Ferrer, L. 2008. Nonparametric feature normalization for svm-based speaker verification. In *ICASSP 2008: IEEE International Conference on Acoustics, Speech and Signal Processing, 2008.*, 1577–1580.
- Terra, E. L., and Clarke, C. L. A. 2003. Frequency estimates for statistical word similarity measures. *Proceedings of the 2003 Human Language Technology Conference of NAACL*.
- Turney, P. D., and Littman, M. L. 2005. Corpus-based learning of analogies and semantic relations. In *Machine Learning* 60(1-3):251–278.