Learning from instances represented as spatially overlapping point sets

No Author Given

No Institute Given

Abstract. This paper promotes the view that in numerous machine learning problems and domains it is valuable to represent instances as point sets and to use the spatial overlap between these point sets as a measure of distance between instances. Our specific contributions are as follows. First, we present a new measure of spatial overlap, and we show that the measure is well-motivated and computationally tractable. Second, we use this measure of spatial overlap to solve three concrete machine learning problems, in clustering, text processing, and protein structure analysis. Finally, we contrast our approach with a variety of other methods that consider the spatial arrangement of point sets.

1 Introduction

One of the most popular and successful representation of instances for learning is as fixed-length feature vectors. In this representation, each feature is represented as a different dimension in a space. This model is popular for several good reasons: (1) it can be used to model a very large class of real problems; (2) it is simple; (3) many powerful inference techniques are easily applied to this model, for example k-nearest neighbors, support vector machines with standard kernels, linear regression, etc.

However, despite its strengths, one weakness of this representation is that it is not easy to encode information about relations among the features themselves directly in this representation. For example, in the bag-of-words model in natural language processing, where each feature represents a term such as "cat", "kitten", and "block", it is not easy to (directly) encode that "cat" and "kitten" are more highly related than "cat" and "block". Similarly, in order to make movie recommendations, we might represent each user as a fixed-length feature vector where each feature encodes the user's rating of a movie; a disadvantage of this approach is that it is not easy to directly encode similarities among movies.

For example, we may represent proteins, documents, movies, and images as collections of atoms, words, reviews, and edges respectively.

In many machine learning problems, we have various features and relations amont

In this paper we present a general construction and we show how it can be used to

In this paper we address the question: how

If the paper's thesis is: it is valuable to encode examples as point sets and to use the spatial overlap between these point sets as a measure of the distance between examples

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 must then find distance measures that enable meaningful comparisons.

Our contribution in this paper is to formulate a new measure, *similarity distance*, that provides an intuitive basis for understanding such comparisons. In this paper, our *things* are finite, weighted point sets. 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 space? The contribution 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. After this, we motivate and derive our measure of similarity. We then examine why this problem defies a number of standard normalization techniques that suggest themselves. This leads us to examine related statistical methods for measuring similarity and explore examples where they fail to capture the simple intuition behind similarity distance and our intended meaning.

1.1 Problem Statement

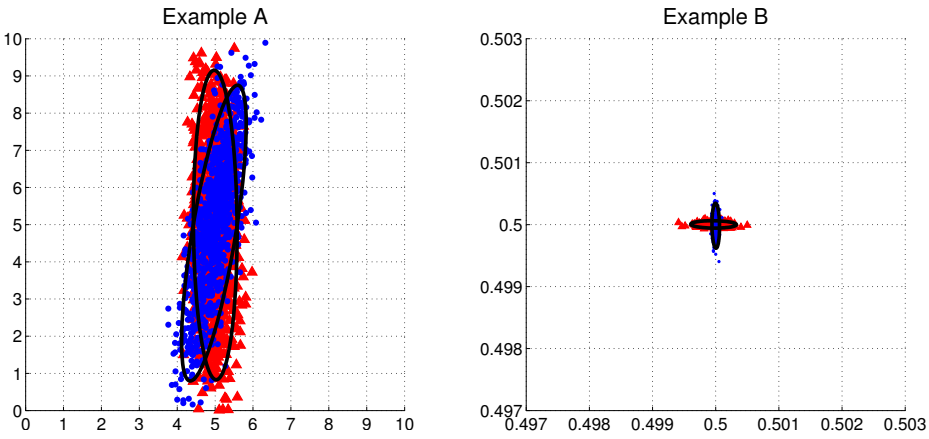

Fig. 1. We consider the two point sets in Example A to be far more similar to one another than those in Example B. This is the case even though they occupy far more area in absolute terms and would be deemed further apart by many distance metrics.

In this paper, we focus on the concept of spatial overlap as our measure of similarity. In other words, we would like to 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. We make no assumptions about the

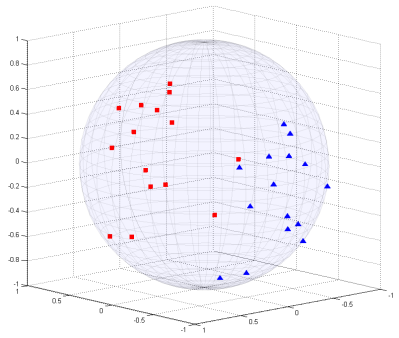

Fig. 2. In this figure, the red squares represent factories (sources) with differing degrees of production and the blue triangles represent warehouse (sinks) with different storage capacities, on the surface of a sphere. The Kantorovich-Wasserstein distance measures the most efficient amount of work necessary to transport from the red squares to the blue triangles. We note the amount of mass being "produced" must be equivalent to the amount of mass being "consumed."

number of points in 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 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. We want to arrive at this result even though the points in Example A cover 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.

In the next section, we formally define this measure and discuss how to compute it, examining the shortcomings of possible alternatives.

2 Similarity Distance

Similarity distance D_S is derived from the Kantorovich-Wasserstein metric D_{kw} [1, 2], 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 as the Earth Mover's Distance [3], which has become popular in computer vision.

We can visualize the problem solved in the computation of D_{kw} in Figure 2. Imagine the red squares are factories located around the world delivering identical goods to the blue triangles, which represent warehouses, also located around the world. We assume the amount of goods to be shipped is equal to amount of goods being received, reflecting the fact that these objects represent probability distributions; they therefore have equal masses of one. D_{kw} is the least amount of work that is required to move the masses contained in the red squares onto the blue triangles.

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 factory delivers its mass to all warehouses independently of any other factory, in proportion to its production. We will call this *naive transportation distance* D_{nt}. In other words, the factories do not communicate. Each simply makes its own deliveries to every warehouse proportionally. Note this is *not* the worst (i.e., most inefficient) transportation schema, which we define below. It is simply what occurs if the factories are oblivious to one another. It happens when they do not take advantage of the potential savings that could be gained by cooperation.

The similarity distance D_S is defined as the ratio D_{kw}/D_{nt}. It measures *the optimization gained by adding cooperation* when moving the source A onto the sink B. Thus, it is a dimensionless quantity that ranges between zero and one.

2.1 Formal Definitions

We now construct similarity distance precisely.

Kantorovich-Wasserstein Distance The discrete formulation of D_{kw} is easily obtained through the discrete version of the Mallow's Distance [4]. Thus, we have the optimization problem for computing $D_{kw}(A, B)$ corresponds to the following minimization problem:

Consider two point sets $A = \{a_1, \ldots, a_m\}$, with associated nonnegative weights p_i, and $B = \{b_1, \ldots, b_n\}$, with associated nonnegative weights q_i, and with both sets of weights summing to one. The Kantorovich-Wasserstein distance is defined as the solution of the linear program

$$\text{minimize } \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}\, d(a_i, b_j) \text{ over } F = (f_{ij}) \text{ subject to}$$

$$\begin{aligned} f_{ij} &\geq 0,\ \ 1 \leq i \leq m,\ 1 \leq j \leq n \\ \textstyle\sum_{j=1}^{n} f_{ij} &= p_i,\ 1 \leq i \leq m \\ \textstyle\sum_{i=1}^{m} f_{ij} &= q_j,\ 1 \leq j \leq n \\ \textstyle\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} &= 1. \end{aligned}$$

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 widely used. Our implementation's runtime fits the function $f(max(m,n)) = \alpha x^{\beta} + \gamma$, where $\alpha = 1.38 \times 10^{-7}, \beta = 2.6, \gamma = -2.5$, with an R^2 value of 1.[1] For enormous point sets, we use standard binning techniques to reduce the computational runtime. A more detailed discussion of binning for this problem is contained in [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. Note that this definition employs a degenerate case of D_{kw}, namely where one of the point sets contains a single point. In this case, $D_{kw} = D_{nt}$, as no optimization is possible and the naive distance is the best one can obtain.

Over weighted point sets corresponding to discrete distributions, we define naive transportation distance D_{nt} defined as:

$$D_{nt}(A,B) = \sum_{i=1}^{m}\sum_{j=1}^{n} p_i q_j \, d(a_i, b_j) = D_{nt}(B,A) \tag{1}$$

The naive distance is the weighted sum of the Kantorovich-Wasserstein distances between each individual points and the entirety of another sample. It is straightforward to directly calculate D_{kw} between a point and a sample and doing so requires $O(k)$ time and therefore calculating D_{nt} requires $O(k^2)$ time, where $k = max(m,n)$. Also, we note from these definitions that D_{nt} is symmetric.

2.2 Asymptotic properties

We consider some properties of similarity distance. First, note 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 "factory" is co-located with a "warehouse" expecting precisely as much mass as it produces.

In contrast, suppose $D_S(A,B) \to 1$. This tells us that cooperation does not help during transportation. When could this occur? It happens 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, $D_{kw}(A,B) \to D_{nt}(A,B)$, implying $D_S(A,B) \to 1$. As $D_{nt}(A,B) \geq D_{kw}(A,B)$ by definition, this provides the upper bound for $D_S(A,B)$ of one. We see this in Figure 3, where the similarity distance between the two illustrated point sets quickly approaches 1 as they are separated. However, as the point sets increasingly overlap, their similarity distance approaches zero rapidly.

[1] Our initial implementation was graciously supplied by Yossi Rubner.

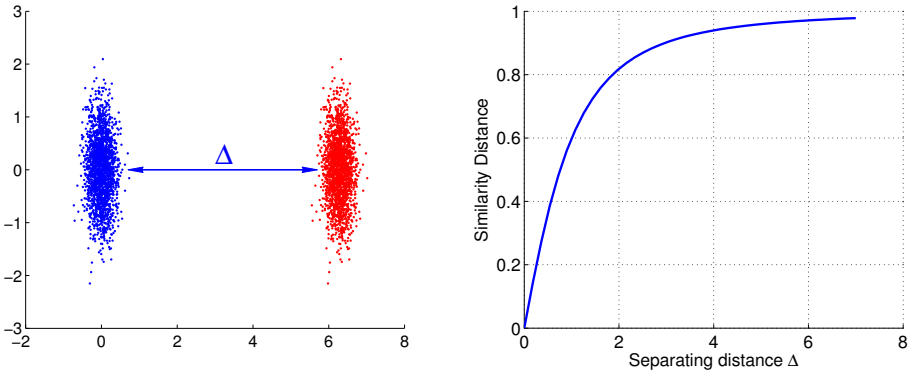

Fig. 3. The graph on the right plots similarity distance as a function of separation distance between the two point sets shown on the left. As can be seen, similarity distance grows non-linearly as the distance between the point sets increases and then asymptotically tapers off to 1.

2.3 Discussion

Given that similarity distance has a domain of $[0, 1]$, one can ask what value indicates "significant" similarity. Is there a particular threshold one can always use? Some applications, e.g., those based on nearest neighbor techniques such as the text classification problem in Section 3.1, do not require such a threshold. In other domains, such as comparing protein conformations (Section 3.2), we use ground truth knowledge about similar and dissimilar proteins to find a domain specific threshold. Given that similarity distance approaches its asymptotic value very quickly, a value of half the mean is often noteworthy, as it is in that domain. However, as with almost all distance measures, establishing a meaningful threshold depends on why one is using it.

3 Applications

3.1 Document Classification

In this section, we use similarity distance, together with k-nearest neighbors, to solve a document classification problem in the 20 Newsgroups dataset [5]. The goal is to determine which newsgroup a given message came from based on the words that occur in the message. We will compare the accuracy of similarity distance with C4.5, Random Forests, and Naive Bayes for classification of this dataset.

Our setup is as follows. Let $\mathcal{D}$ be a collection of documents and $\mathcal{V}$ the collection of distinct words occuring in those documents. We consider each document $\mathbf{d} \in \mathcal{D}$ to be the set of words $\{w \in \mathcal{V} : w \in \mathbf{d}\}$ occuring in the document.

Between any two words $w, v \in \mathcal{V}$, the pointwise mutual information (PMI) between w and v is defined as

$$\text{PMI}(w, v) = \log \frac{P(w, v)}{P(w)P(v)},$$

where $P(w)$ is the probability that word w occurs in a document and $P(w, v)$ is the probability that words w and v both occur in a document. PMI, in this context, can be thought of as a measure of word similarity; many such measures have been proposed [**?**,**?**,**?**,**?**,**?**,**?**,**?**,**?**,**?**,**?**],

[**?**,**?**,**?**,6], but [6] found PMI to be more effective than numerous competitors, at least judged by the task of predicting synonyms on multiple-choice TOEFL exams.

Next we construct a "semantic space". We fix a set $\{w_1, \ldots, w_p\} \subseteq \mathcal{W}$ of "reference words" having high mutual information with the labels. We then define a map $f : \mathcal{V} \to \mathbb{R}^p$ taking each word to the vector of its PMI with each reference word, i.e., $f(w) = (\text{PMI}(w, w_1), \ldots, \text{PMI}(w, w_p))$. Words that have similar PMI with the reference words will be located near each other in this "semantic space".

The image of each document under the map f is a point set in the semantic space $\mathbb{R}^p$, and inference can be performed using similarity distance on $\mathbb{R}^p$. Compared to the most common representation of documents for text classification as "bag of word" vectors, our construction has a distinct advantage because it does not ignore semantic relations between words.

Experimental Procedure We present the results of an experiment on the 20 Newsgroups dataset [5]. For our experiment, we chose 30 articles at random from each of two newsgroups, alt.atheism and sci.med. We applied simple preprocessing to each article: we tokenized, downcased, and removed punctuation, stopwords, and words occuring only once in the collection; 2015 distinct words remained. We selected 6 reference words (*christian*, *doctor*, *god*, *medical*, *say*, *atheists*) having high expected mutual information with the newsgroup label. To estimate the PMI between words, we recorded the number of hits c_w and $c_{w,v}$ reported by Google for each word w individually and for each pair of words (w, v), and we set $\hat{P}(w, v) = c_{w,v}/N$, $\hat{P}(w) = c_w/N$, where N is a normalizing constant. We estimate

$$\widehat{\text{PMI}}(w, v) = \log \frac{\hat{P}(w, v)}{\hat{P}(w)\hat{P}(v)} = \log \frac{c_{w,v}}{c_w c_v} + \text{const},$$

and we set const $= 0$ for convenience. Thus, in this experiment, the map from words into the semantic space becomes

$$\hat{f}(w) = (\widehat{\text{PMI}}(\textit{christian}, w), \ldots, \widehat{\text{PMI}}(\textit{atheists}, w)).$$

We perform classification on the collection of images of documents under $\hat{f}$.

Results As we see from Table 1, similarity distance is able to exploit semantic relationships between words, as reflected by their mutual information, to successfully classify samples in this experiment. This gives it a marked advantage over competing techniques

Input space	Procedure	Accuracy	Precision	Recall	F-Measure
BOW	C4.5	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	1-nearest neighbor	85.00	0.860	0.850	0.849
(Sim. dist.)	2-, 3-, 4-nearest neighbor	85.00	0.854	0.850	0.850
	5-nearest neighbor	81.67	0.835	0.817	0.814
	SVM	90.00			

Table 1. Results of textual experiment using 10-fold cross validation.

and the acquired mutual information over word pairs is additionally reusable. Additionally, similarity distance provides an easy way to visualize and understand the results, which is uncommon in many classification tasks, as shown in Figure 4.

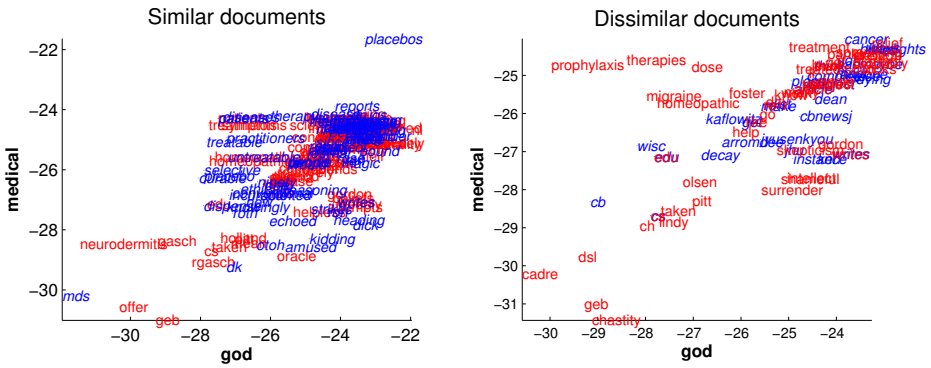

Fig. 4. In the example above, the point sets corresponding to two documents are plotted in the semantic subspace defined by *god* and *medical*. In each plot, one document is display in a blue italic font and the other is displayed in a red non-italic font. On the left, the two documents are from the same newsgroups. On the right, the documents are from different newsgroups. Similarity distance captures the intuitive notion of spatial overlap corresponding to these classifications. (Note that although the similarity distance computations in semantic space are performed in $\mathbb{R}^6$, only two dimensions are visualized here.)

3.2 Protein Structure Similarity

A fundamental problem in protein structure analysis is determining whether two proteins have similar folded conformations, especially when they have low sequence homology. A widely used method for determining similarity between structures is the

DALI algorithm, which is the basis behind the database Families of Structurally Similar Proteins (FSSP) [7].

We approach this problem by representing each protein molecule as a weighted point set of its constituent atoms' positions. Thus, given two proteins, we can derive a measure of their *structural similarity* by finding the similarity distance between their point set representations.

The first step in this process is spatially aligning the proteins to compute their similarity distance. We perform this alignment using simulated annealing over gradient descent, guided by the value of D_{kw} between the two structures.

Once the closest structural match has been found, we measure similarity distance between the two proteins. In the example shown in Figure 5, we aligned and compared two protein structures with PDB IDs **1ABA** and **1GRX**. These two proteins are functionally similar and belong to the Glutaredoxin subgroup; however, they come from different organisms and have different amino acid sequences. We performed the alignment on the first 25 backbone atoms (alpha carbons), and then applied the transformation to the residues corresponding to those atoms. The result of the alignment is shown in Figure 5. The similarity distance between the aligned point sets is 0.236, indicating a structural homology. The number 25 was chosen for visualization purposes; similarity distance between the two full chains is 0.239. Between non-similar protein structures similarity distance averages at 0.70. This type of analysis can be used to automatically determine pairs of proteins with similar structures.

We were able to find this surprising result because similarity is determined between entire protein structures; the biologically interesting question here is how well do two proteins' folded conformations overlap, as similar structure is often an indication of similar function. Similarity distance can then be seen as a measure of how closely the atoms in one structure mirror those in the other, with the magnitude indicating the quality of overlap.

4 Related Work and Comparisons

Prior work on quantifying similarity between point sets or measuring a distance between them generally falls within one of three categories, none of which are specifically designed to measure overlap or similarity.

4.1 Modified statistical distance measures

Metrics for comparing probability distributions - such as Mallows distance - can sometimes be modified to measure distance between point sets. Because Mallows distance computes the infimum of the expected value of functions on random variables, we can transform this into a discrete minimization problem [4].

Other metrics compute differences between probability mass or density functions, which have no immediate applicability in the discrete point set case without an intermediate step. It is possible to view coordinates of points as being values taken by discrete random variables but it is rarely the case that multiple points have precisely the same

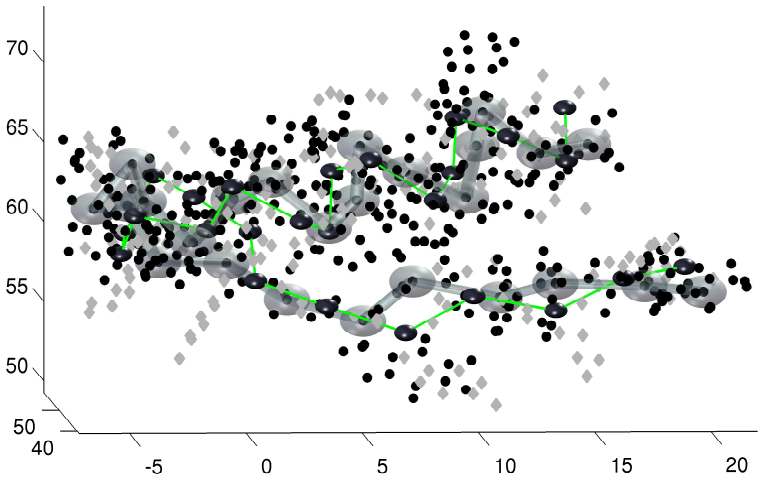

Fig. 5. Similarity distance between parts of two proteins, 1ABA and 1GRX. The larger spheres and sticks represent the backbone carbon chain, and the points represent atoms from their residues.

coordinates, making probabilities for each location degenerate into zeros or $\frac{1}{n}$, where n is the number of points.

For sets with large numbers of points we can bin them into regions, treat each region as having a probability value proportionate to the number of points lying within it, and apply any of a number of probability divergence measures such as Bhattacharya distance, KL-divergence, Hellinger distance or any of the family of such measures [2]. We note that this is an approximation that degrades with point sets of low density. Comparisons are provided in the next section.

4.2 Point-set Distance extensions

Other 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 the modified Hausdorff metric and Busemann metric [2]. We discuss Hausdorff distance further below.

4.3 Procrustes Distance and Variations

A third method of computing distances between point sets is to assume an order between them and calculate distortion 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. Modifications exist to overcome the cardinality problem by only considering pairs up to the cardinality of the smaller set and ignoring the rest. These general methods of summing distances between pairs of points do not yield any information about similarity or shape congruence, as seen in the next section.

4.4 Others

The previous two types of distances reduce two point sets to pairs of points or a single pair of points. In many domains this appears to work suitably, especially image matching. But ignoring all points except for one pair (or a restricted set of pairs) yields no information about how similar the shapes of the entire point sets are. It collapses all information down to a single distance (or the sum of a few distances), stripping away all information about the internal layout and structure of each point set, as well as the relationship of points within each point set. There is also no easy way to determine how different or similar two point sets are just by examining the distance returned because there is no reference point for similarity or dissimilarity.

Finding similarity between multi-dimensional point sets is a core problem in image matching. The driving concern in that domain however is to locate objects similar to each other but transformed in some simple way, such as being rotated, reflected or translated in one of the two images [8]. The focus is on preserving distance across transformations and so the distance measures used are very primitive, e.g. minimal symmetric set difference across all translations.

4.5 Normalization techniques

Similarity distance measures the amount of optimization provided by cooperative vs. independent, naive transportation; intuitively, it measures the spatial overlap between two weighted point sets. One might ask how else similarity might be computed from D_{kw}. A number of schemes have been devised to rescale data in order to normalize it, e.g., [9] for overviews and empirical evaluations. We compared Similarity distance with linear scaling; sample mean and variance normalization; sample mean normalization; sample variance normalization; Gaussianization, and Distribution matching. It was straightforward to find examples for all of these where they did not capture any notion of spatial overlap.

4.6 Comparisons

We compare similarity distance with several of the measures mentioned in the previous section and demonstrate with representative examples that similarity distance captures the notion of similarity more accurately.

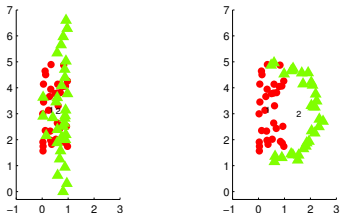

Fig. 6. The two pairs of point sets shown here are clearly different. There is no overlap in the second case, and yet the Hausdorff distance and Procrustes distance between the two are almost identical.

In Figure 6 we compare Hausdorff distance with similarity distance. Note that the point sets on the left overlap more with each other than the ones on the right and are more alike in their shape. However Hausdorff distance is unable to differentiate between them, reporting a distance of 1.75 in both cases. Similarity distance reports 0.38 in one case and 0.61 in the other.

Next we look at Procrustes distance between two point sets. In this case as well, Procrustes returns almost equal distances of 1.87 and 1.91, unable to tell the two pairs of point sets apart.

The final comparisons are with probability divergence measures. Each dataset is processed into a set of Voronoi regions using k-means clustering, and each region is treated as a value of a random variable, whose probability is equal to the fraction of points of that point set lying within that region. In this way we make sure to operate over the same domain, which allows the use of these measures. We chose Hellinger distance as a representative of the family of measures.

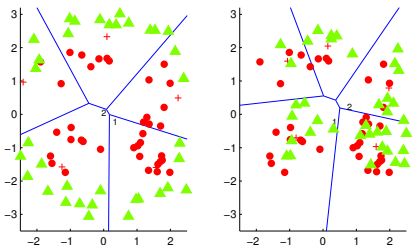

Fig. 7. The two pairs of point sets shown here are clearly different. There is no overlap in the first case, and yet the distance according to probability divergence measures is 0.37 and 0.34 respectively.

In Figure 7(a) with $k = 5$ the Hellinger distance is 0.37 and in Figure 7(b) it is 0.34 . To put these values in perspective, note that the Hellinger distances between the point sets in Figure 6 are 0.64 and 1.47 respectively. In contrast, similarity distance values are 0.45 and 0.32 respectively, indicating a clear difference.

A conceptual drawback of using this technique is that it is approximate - the points may lie anywhere within their Voronoi region and the divergence measure would return the same value. There is also no clear way of choosing the regions or even their number. Other divergence measures such as Bhattacharya distance, chi-squared distance, and Jeffrey divergence result in similar values and behavior. None are suited to the measure of similarity between point sets.

Normalization schemes.

Linear scaling. Fix a lower bound L_i and an upper bound U_i on feature i. For example, L_i and U_i can be set to the largest and smallest values of feature i in the corpus, or they can be set so as to capture some fraction of the variance of feature i in the corpus. Define $T_i(x_i) = (\min(\max(x_i, L_i), U_i) - L_i)/(U_i - L_i)$. Then $0 \leq T_i(x_i) \leq 1$.

Sample mean and variance normalization. Given a corpus $\{\mathbf{x}_j\}_{j=1}^n$ of examples, let $m_i = n^{-1} \sum_{j=1}^n x_{ji}$ denote the sample mean of the ith feature and $s_i^2 = (n-1)^{-1} \sum_{j=1}^n (x_{ji} - m_i)^2$ denote the sample variance. Define $T_i(x_i) = (x_i - m_i)/s_i$. Then the collection $\{T_i(x_{ji})\}_{j=1}^n$ has sample mean zero and sample variance 1. Used by [9], who .

Sample mean and variance normalization. Given a corpus $\{\mathbf{x}_j\}_{j=1}^n$ of examples, let $m_i = n^{-1} \sum_{j=1}^n x_{ji}$ denote the sample mean of the ith feature and $s_i^2 = (n-1)^{-1} \sum_{j=1}^n (x_{ji} - m_i)^2$ denote the sample variance. Define $T_i(x_i) = (x_i - m_i)/s_i$. Then the collection $\{T_i(x_{ji})\}_{j=1}^n$ has sample mean zero and sample variance 1.

Sample mean normalization. Given a corpus $\{\mathbf{x}_j\}_{j=1}^n$ of examples, let $m_i = n^{-1} \sum_{j=1}^n x_{ji}$ denote the sample mean of the ith feature. Define $T_i(x_i) = (x_i - m_i)/s_i$. Then the collection $\{T_i(x_{ji})\}_{j=1}^n$ has sample mean zero.

Sample variance normalization. Given a corpus $\{\mathbf{x}_j\}_{j=1}^n$ of examples, let $m_i = n^{-1} \sum_{j=1}^n x_{ji}$ denote the sample mean of the ith feature and $s_i^2 = (n-1)^{-1} \sum_{j=1}^n (x_{ji} - m_i)^2$ denote the sample variance. Define $T_i(x_i) = x_i/s_i$. Then the collection $\{T_i(x_{ji})\}_{j=1}^n$ has sample variance 1.

Rank normalization. Given a corpus $\{\mathbf{x}_j\}_{j=1}^n$ of examples, define $T_i(x_i) = n^{-1} \sum_{j=1}^n I(x_{ji} \leq x_i)$, where $I(A)$ is the indicator function of A. Then $0 \leq T_i(x_i) \leq 1$, and the collection $\{T_i(x_{ji})\}_{j=1}^n$ becomes uniformly distributed as n increases,[2] since T_i is feature i's empirical distribution function.

Distribution matching. Given a corpus $\{\mathbf{x}_j\}_{j=1}^n$ of examples, with cdf F_i and desired cdf G_i, define $T_i(x_i) = G_i^{-1}(F_i(x_i))$. If F_i is the empirical cdf and G_i is the identity function $r \mapsto r$, then distribution matching reduces to rank normalization. Claim: Distribution matching also generalizes gaussianization.

5 Conclusion

References

1. Kantorovich, L.V.: On the translocation of masses. Journal of Mathematical Sciences (1942)
2. Deza, M.M., Deza, E.: Encyclopedia of Distances. Springer (2009)

[2] Right?

3. Rubner, Y., Tomasi, C., Guibas, L.: The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision **40**(2) (2000) 99–121
4. Levina, E., Bickel, P.: The earth mover's distance is the mallows distance: Some insights from statistics. IEEE International Conference on Computer Vision **2** (2001) 251
5. Lang, K.: Newsweeder: Learning to filter netnews. In: Proceedings of the Twelfth International Conference on Machine Learning. (1995) 331–339
6. Terra, E.L., Clarke, C.L.A.: Frequency estimates for statistical word similarity measures. Proceedings of the 2003 Human Language Technology Conference of NAACL (2003)
7. Holm, L., Sander, C.: Touring protein fold space with dali/fssp. Nucleic Acids Res **26** (1998) 316–319
8. Hubo, E., Mertens, T., Haber, T., Bekaert, P.: Special section: Point-based graphics: Self-similarity based compression of point set surfaces with application to ray tracing. Comput. Graph. **32**(2) (2008) 221–234
9. Stolcke, A., Kajarekar, S., Ferrer, L.: Nonparametric feature normalization for svm-based speaker verification. In: Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on. (31 2008-April 4 2008) 1577–1580