Learning Wormholes for Sparsely Labelled Clustering

Eng-Jon Ong and Richard Bowden
Centre for Vision, Speech and Signal Processing
University of Surrey, Guidlford, Surrey, UK
(e.ong,r.bowden)@surrey.ac.uk

Abstract

Distance functions are an important component in many learning applications. However, the correct function is context dependent, therefore it is advantageous to learn a distance function using available training data. Many existing distance functions is the requirement for data to exist in a space of constant dimensionality and not possible to be directly used on symbolic data. To address these problems, this paper introduces an alternative learnable distance function, based on multi-kernel distance bases or "wormholes that connects spaces belonging to similar examples that were originally far away close together. This work only assumes the availability of a set data in the form of relative comparisons, avoiding the need for having labelled or quantitative information. To learn the distance function, two algorithms were proposed: 1) Building a set of basic wormhole bases using a Boosting-inspired algorithm. 2) Merging different distance bases together for better generalisation. The learning algorithms were then shown to successfully extract suitable distance functions in various clustering problems, ranging from synthetic 2D data to symbolic representations of unlabelled images.

1 Introduction

Distance functions are an important component in many learning applications. Clustering or classification algorithms typically rely on some form of distance function that has been *apriori* defined within an input space. Additionally, it is often the case that the correct function is context dependent, in fact, it is often not possible to choose a specific distance function. It would therefore be advantageous to learn a distance function using available training data.

Many approaches to learning distance functions take the form of a weighted Euclidean function. A popular approach is the Mahalanobis function[5]. The exact approach to learning the parameters of the transformation matrix for the Mahalanobis distance varies according to author[5, 1]. The Mahalanobis function can cause problems in cases where a discontinuous input-space is present(e.g. XOR problem). Another limitation of these distance functions is that they require the data to exist in a space of constant dimensionality. Therefore, it is not possible to use such methods on symbolic data that often have varying size.

To address the above problems, this paper introduces an alternative learnable distance function, based upon a combination of multi-kernel distance bases. Each distance basis essentially introduces "wormholes that connects areas that may be far away together. This effectively allows us to bring spaces belonging to similar examples that were originally far away close together. The use of kernels is also important as it allows us to remove the requirement for fix dimensional vector-based data.

This work only assumes the availability of a set of relative comparisons as in [4], where given three variables, A, B and C, A is closer to B than C. This avoids the need for having labelled or quantitative information. To learn the distance function, two algorithms are proposed. The first algorithm builds a set of basic distance bases. To improve the generalisation capability of the final distance function, a second algorithm is proposed to merge different distance bases together. We will show how the learning algorithms were then successfully used to extract suitable distance functions in various clustering problems, ranging from synthetic 2D data to symbolic representations of sparsely labelled images.

The next section will introduce the learnable wormhole-based distance function. Following this, we explain the learning algorithm for the distance functions in Section 3. We describe how this method is applied to the clustering of unlabelled images and show some experimental results in Section 4 before concluding in Section 5.

2 Learnt Distances using Wormholes

In this section, we will introduce a learnable distance function. At its heart is a collection of kernel functions, $K(x, c)$, where c is the centre of the kernel, taken from a

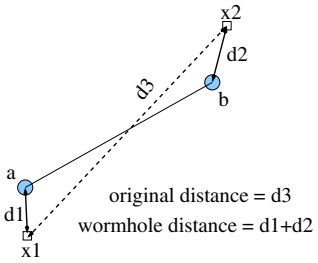


Figure 1. Illustration of the wormhole distance, as compared to normal Eucledian distance. Introducing a wormhole basis with kernels a and b has shortened the distance between x1 and x2 considerably

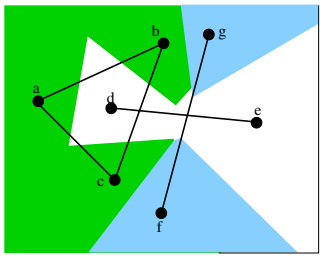


Figure 2. The use of multiple wormhole bases ($\{a,b,c\},\{d,e\},\{f,g\}$)allows different regions (colours) to be pulled close together.

training example and x is some new input example. Such a kernel function provides some form of a primitive similarity measure between x and c. For example, in the case of a space of fix dimensionality, one possible kernel is the Euclidean distance, such that, given a new input vector x, the kernel then takes the form of:

$$K(x,c) = \sqrt{|(x-c)^2|} \tag{1}$$

For illustration purposes, the Euclidean distance kernel will be assumed for the rest of this section and the next section. However, we will see later the use of other types of kernels for more learning distance functions for solving more complicated problems.

Key to the learnt distance function is to warp the space to overcome the inadequacies of a single kernel. Thus, in order to pull two or more far away areas close together, these kernels are grouped together into a distance basis. Our final distance function will contain a number (N_B) of separate wormhole bases. Each distance-basis is associated with a set of kernels: $C_j = \{c_{ji}\}_{i=1}^{b_j}$, where b_j is the number of kernels and c_{ji} is the i^{th} kernel center for the j^{th} basis respectively. The set of kernels in a distance-basis can now be used to provide a measure of "nearest-distance" as follows:

$$B(x,y,C_j) = \min(K(x,c_{ji}))+\min(K(y,c_{ji})), i = 1...b_j \tag{2}$$

We can also think of each distance basis as a zero-distance wormhole with multiple entrances, where each entrance is the kernel centre. Thus, from 2, all the kernel centres associated with the same basis is zero distance to each other (see Fig. 1. For this reason, from this point on, $B(x,y,C_j)$ will be referred to as a *wormhole basis*.

All the wormhole kernels that form a wormhole basis, can be grouped into $C = \{C_j\}, j = 1...N_B$. The distance function between two points (x and y) is defined as:

$$D(x,y,C) = \min(B(x,y,C_j)), j = 1,...N_B \tag{3}$$

Such a combination of wormhole bases allows us to partition the space into smaller sub-spaces that can then be pulled close together (see Fig. 2).

3 Distance function Learning using Relative Examples

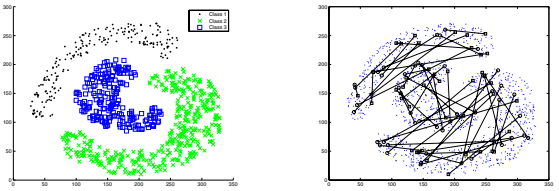
This section shows how the kernel centres are chosen and subsequently how the distance-bases are formed using an algorithm inspired by the Boosting method [3] of learning. The learning framework consists of two major steps. The first step learns primitive wormhole-bases, each containing only two kernels. It was found that this is enough to give a very low training error. However, it does not provide good generalisation capabilities. To address this, a second step whereby these simple distance-bases are merged together to form a more general distance function is introduced.

Before the algorithm is described, a few definitions are provided. A training dataset consisting of N_T relative comparison triplets is defined as: $T_j = \{t_{ji}\}_{i=1}^{3}, j = 1...N_T$. These training triplets are created in such a way that t_{j1} is closer to t_{j2} than t_{j3}. The entire training dataset is defined as $T = \{T_j\}_{j=1}^{N_T}$. Additionally, each training example is associated with a weight $W = \{w_j\}_{j=1}^{N_T}$. Given a set of wormhole bases C, the training examples T, and their weights W, the training error function $E(T,W,C)$ is defined as follows:

$$E(T,W,C) = \sum_{j=1}^{N_T} w_j G(T_j,C) \tag{4}$$

$$G(T_j,C) = \begin{cases} 1 & (D(t_{j1},t_{j2},C) > D(t_{j1},t_{j3},C)) \\ 0 & (D(t_{j1},t_{j2},C) < D(t_{j1},t_{j3},C)) \end{cases} \tag{5}$$

where $G(T_j,C)$ is the individual error function for the j^{th} triplet given a set of distance bases C. To obtain the set of potential primitive distance-bases, the first two examples of each training triplet is used: $K_k = \{t_{k1},t_{k2}\}$. The entire set of primitive distance bases are denoted as: $K = \{K_k\}_{k=1}^{N_T}$.



(a) (b)

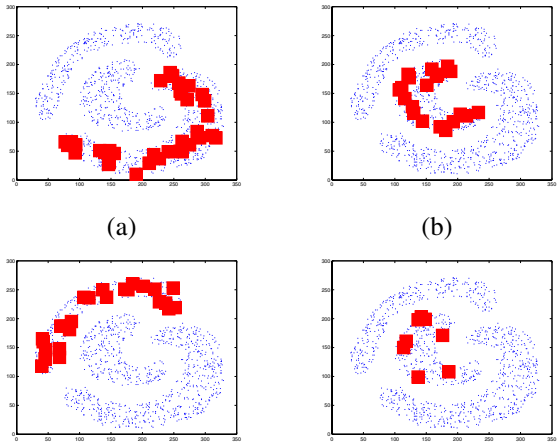
Figure 3. The learnt distance applied to the 3 cluster problem. a) groundtruth, b) learnt primitive wormholes.

3.1 Learning and Merging Primitive Wormholes

The algorithm for learning the primitive distance-bases primarily revolves around a distance-basis selection loop. Within this loop, a new primitive wormhole-basis is chosen that provides the smallest training error and is then added into the existing set. The loop terminates when the training error falls below a threshold t. An example of the result of the algorithm applied to the three non-linear cluster problem (see Section 3.2 for details) can be seen in Fig. 3b. The primitive wormhole-bases are shown as lines linking two kernel centres. The algorithm will result in a set of N_B wormhole-bases C as follows:

1: Initialisation Step
   (i)$N_B = 0$, $M = 1$, $w_j = 1, j = 1...N_T$
   (ii) $C_0 = \{\}$ {No distance bases found yet}
2: **while** $\sum_{j=1}^{N_T} w_j > t$ **do**
3: $K_{best} = \arg\min_{K_{best} \in K} E(T, W, \{C_{M-1}, K_{best}\})$
   { Find least training error distance-basis }
4: $C_M = \{C_{M-1}, K_{best}\}$
5: $w_j = G(T_j, C_M), j = 1...N_T$ { Update the weights }
6: $M = M + 1$
7: **end while**
8: $N_B = M, C = C_M$, break

To obtain a distance function with better generalisation capabilities, it is necessary to merge the above primitive distance-bases together. This is achieved by merging two distance bases together if the result of this action does not cause the existing training error to increase. An example of the merging algorithm applied to the 3 cluster problem is shown in Fig. 3. Each figure shows a group of kernels (red squares), the result of merging various primitive wormhole bases into a single larger wormhole.



(a) (b) (c) (d)

Figure 4. Example of 4 different merged wormhole bases kernels

3.2 Clustering Synthetic Data

To illustrate the algorithm working on synthetic data, the learning method was applied to a non-linear set of 3 clusters (see Fig. 3a). Training data triplets are produced by randomly selecting two points from the same cluster and the third from any other random cluster. A distance function is learnt using the training data. The resulting wormholes before merging in Fig. 3and after merging can be seen in Fig. 4. The results of the learnt distance function are shown in Fig. 5. A random point from each class is chosen from the test dataset (black box in Fig. 5a, c, e). The distance of this point to all the other points in the figure is calculated using the learnt distance function. The points which are "close" (i.e. distance less than a pre-defined threshold) to the selected point are filled red circles in Fig. 5(b,d,f). A distorted space using the distance measures is shown in Fig. 5c,f. Here, the selected point is made the origin, all other points are projected onto a unit circle around the origin and scaled using their respective distances. The circle defines the isocontour boundary from the origin.

4 Clustering Sparsely-labelled Images

In this section, we will describe how we perform object clustering on only partially labelled images. The COIL image database was used for this purpose. This database consists of a series of images of objects rotating on a turn-table. Initially, salient feature points using the method proposed by Kadir and Brady [2]. A kernel function computing the sum of distance between the nearest neighbours of a set of

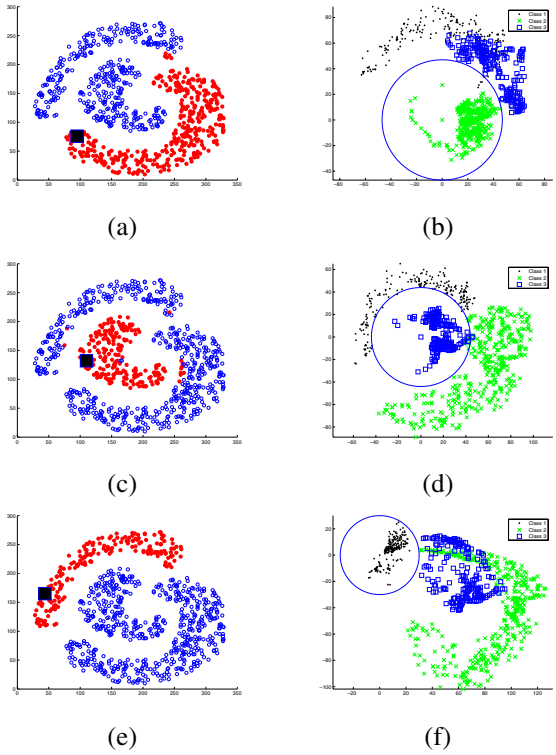


Figure 5. 3 cluster experiment results

feature points to another set of feature points is used. We note the above kernel is far from adequate for clustering images. Thus, we aim to improve on this by learning a distance function on top of these non-robust kernels. The test triplets are generated using two successive frames of the rotation sequence of each object as similar points and a random frame from some other object as being dis-similar. A subset of images for each object was used for training the two distance functions, whilst the remaining used for testing. The inadequacies of the original distance function are shown in the MDS visualisation of its test-set distance matrix (see Fig. 6a), where there is greater overlap between images of different objects. Using the wormhole distance function, we can successfully separate data in each class and discovering the total number of classes (see Fig. 6b).

5 Conclusion

In this paper, an alternative learnable distance function using multi-kernel wormhole-bases was proposed. Each wormhole basis has the effect of connecting areas that may be far away together. Crucially, the use of kernels allowed the application of this function to not only fixed-dimension vector data but also symbolic data. The use of relative comparisons for training data avoided needing labelled or

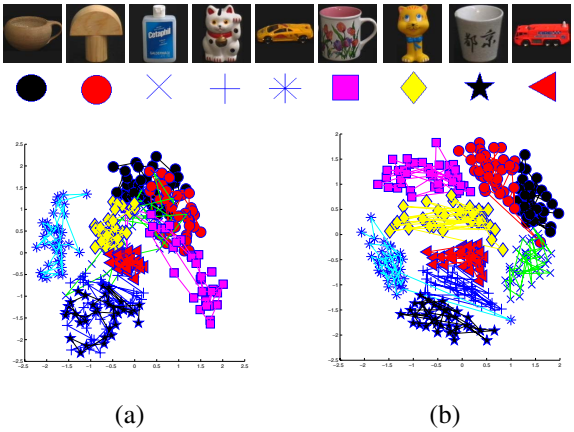


Figure 6. Visualisation of distance matrices of COIL database. On top are the 9 object used and their representative glyphs.

quantitative information. Two algorithms were proposed for learning the distance functions: The first algorithm for building a set of basic distance bases; a second algorithm for increasing generalisation by merging different distance bases together. The learning algorithms were shown to have been successfully applied in learning suitable distance functions for various clustering problems, ranging from synthetic non-linear 2D data to symbolic representations of sparsely labelled images.

Acknowledgements

This investigation reported in this contribution has been supported by the European Union (FP6-project 'COSPAL', IST-2003-2.3.2.4).

References

[1] A. Bar-Hillel and D. Weinshall. Learning distance functions equivalence relations. In *Proceedings of the 16th Conference on Learning Theory (COLT)*, August 2003.

[2] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *Proceedings of the European Conference on Computer Vision 2004*, pages 228–241, 2004.

[3] R. Meir and G. Rätsch. *Advanced Lectures on Machine Learning*, chapter An introduction to boosting and leveraging, pages 119–184. Springer Verlag, 2003.

[4] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Proceedings of the Conference on Advance in Neural Information Processing Systems (NIPS)*, 2003.

[5] I. Tsang and J. Kwok. Distance metric learning with kernels. In *Proceedings of the International Conference on Artificial Neural Networks*, 2003.