Learning Distance for Arbitrary Visual Features

Eng-Jon Ong and Richard Bowden
Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford, GU2 7XH, UK
e.ong,r.bowden@surrey.ac.uk

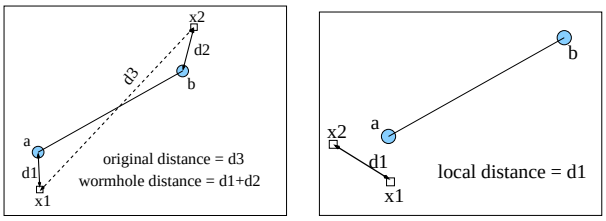
Abstract

This paper presents a method for learning distance functions of arbitrary feature representations that is based on the concept of wormholes. We introduce wormholes and describe how it provides a method for warping the topology of visual representation spaces such that a meaningful distance between examples is available. Additionally, we show how a more general distance function can be learnt through the combination of many wormholes via an inter-wormhole network. We then demonstrate the application of the distance learning method on a variety of problems including nonlinear synthetic data, face illumination detection and the retrieval of images containing natural landscapes and man-made objects (e.g. cities).

1 Introduction

The task of obtaining a distance function for a visual representation feature space is important in many computer vision applications. Many algorithms in vision are reliant upon methods involving clustering or classification, which in turn requires some form of distance function has been *apriori* defined within an input space. One common distance function used is the Euclidean distance. However, this distance function is often inadequate when applied to visual representations due to the inherent non-linearity and discontinuities present in the data. Additionally, the correct distance function is often context dependent. As a result, no one distance function will suffice for all applications. In an attempt to overcome this, many approaches opt for learnt distance functions in the form the Mahalanobis function, with varying methods for learning the required parameters [6, 1]. However, there are disadvantages, for example a discontinuous input-space (e.g. XOR problem) cannot be represented. Another limitation is that they require the fixed dimensional data. Therefore, it is not possible to use such methods on symbolic data.

To address the above problems, this paper proposes a new learnable distance function along with its learning algorithms and applies it to a number of vision tasks. Firstly, Section 2 introduces a novel learnable distance function, based on dual-kernel distance bases or wormholes. These wormholes warp the topology of the input space merging similar examples that were originally far away. The use of kernels is also important as it removes the requirement of fixed dimensional vectors. We also describe a novel method for combining different wormholes into the final distance function using an inter-wormhole network and demonstrate how it improves the generalisation ability of the learnt distance function. Following this, we describe the learning algorithms for both the wormholes and inter-wormhole network in Section 3. An important aspect of these learning algorithms



(a) (b)

Figure 1: Illustration of wormhole distance compared to Euclidean distance. A wormhole basis with kernels *a* and *b* shortens the distance between x1 and x2 considerably.

is that only one assumption is made, the availability of a dataset of relative comparisons, avoiding the need to have labelled or quantitative information. We apply our method to three different applications in Section 4. The first shows the ability of the method to learn a distance function for clustering synthetic 2D non-linear and discontinuous data. We then show how a distance function is learnt for identity invariant face illumination estimation. We also apply the same methods to the problem of image retrieval before concluding in Section 5.

2 Kernel-based Wormholes

The concept of kernel-based wormholes is now described. Central to the distance function is a collection of kernels. Each kernel is built using some form of a local distance between two points (x,y), defined as $K(x,y)$. We will see in Section 4 the different types of kernels used. However, such a local distance is usually not sufficient. Thus, the key to our learnt distance function is its ability to warp the input space to overcome the inadequacies of the local distance provided by a single kernel.

2.1 Primitive Wormholes

In order to pull two distant regions close, two kernels are grouped together into a *distance basis*. Each distance-basis is associated with a pair of kernels: $C_j = \{c_{ji}\}_{i=1}^{2}$, where c_{ji} is the example representing the i^{th} kernel centre for the j^{th} basis respectively. The kernel pair in a distance basis can now be used to provide a local measure of "nearest-distance" as follows:

$$B(x,y,C_j) = \begin{cases} \arg\min_k(K(x,c_{jk})) + \arg\min_l(K(y,c_{jl})) & (k \neq l) \\ K(x,y) & (k = l) \end{cases} \quad (1)$$

where $k,l = \{1,2\}$ is the kernel centre closest to x and y respectively. Here, when x and y are close to the same kernel centre, the local measure of difference provided by the chosen kernel (see Section 4) is used. However, when each point is close to a different kernel centre, Eq. 1 effectively "short-circuits" the distance spanned between the two kernels, greatly reducing the distance between x and y (see Figure 1). We can also think of each distance basis as a zero-distance wormhole with two entrances (kernel centres). From this point on, $B(x,y,C_j)$ will be referred to as a *primitive wormhole basis*.

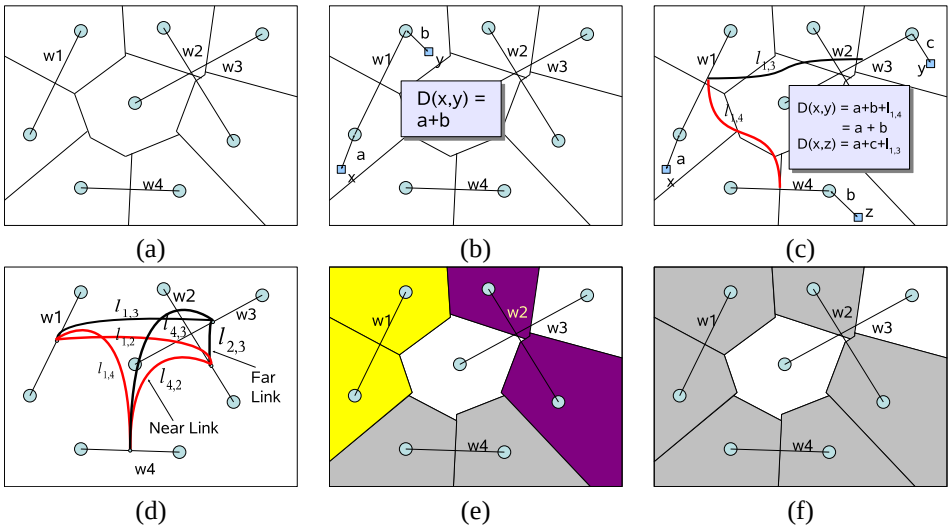


Figure 2: (a) Primitive wormholes & catchment areas. Distance of (x,y) caught by same (b) different (c) primitive wormhole, inter-wormhole network links (d) generalisation improves using the network by tying different wormhole catchment areas together (e,f).

2.2 Inter-wormhole Network: Improving Generalisation

We now see how primitive wormholes are combined into a suitable distance function. A conceptually attractive and intuitive method is to link together primitive wormholes such that each wormhole entrance is associated with a local subspace acting as its catchment area. These catchment areas should be mutually exclusive and unique to a single primitive wormhole entrance. To this end, we have chosen a method of combining the primitive wormholes together in such a way that the input space is partitioned similar to a Voronoi tessellation (see Figure 2a). A primitive wormhole is then associated to two cells. In computing the distance, when both examples fall into catchment areas associated with the same wormhole, Eq. 1 can be used to calculate the distance between (x) and (y) (see Figure 2b).

However, if x and y are each caught by different wormholes (w_x) and (w_y) respectively, we will need to traverse two primitive wormholes to get the final distance between them. To do this, the concept of an inter-wormhole network represented by a link matrix I is introduced. Each matrix element (I_{ij}) is the cost for traversing between two different primitive wormholes w_i and w_j (see Figure 2c). The distance between two points caught by different wormholes is then the sum of distances of the points to their associated wormhole entrances added to the corresponding element from the link matrix (see Figure 2d). We define the total number and set of primitive wormholes as N_B and $\{C\}_{i=1}^{N_B}$ respectively. The final wormhole-based distance function is defined as:

$$W(x,y,C_{i=1}^{N_B}) = \begin{cases} \arg\min_{i,k}(K(x,c_{ik})) + \arg\min_{j,l}(K(y,c_{jl})) + I_{ij} & (i \neq j) \\ B(x,y,C_i) & (i = j) \end{cases} \quad (2)$$

where x is closest to the i^{th} wormhole's k^{th} entrance (c_{ik}) and y closest to the j^{th} wormhole's l^{th} entrance (c_{jl}). Presently, we restrict the inter-wormhole links to be either *far* or *near*. Two primitive wormholes (w_i, w_j) are defined to be near if the cost of traversing them is 0, ($I_{ij} = 0$). Two primitive wormholes can be pushed far away by setting their inter-wormhole distance to a large pre-defined value. One advantage offered by introducing the inter-wormhole network is improved generalisation for the final distance function. When all primitive wormholes are disjoint (i.e. far away from each other), only the two catchment areas of a wormhole is close. However, by closely linking two or more primitive wormholes, more subspaces can be brought together (see Figure 2e,f).

3 Learning the Wormholes

This section will describe how the appropriate primitive wormholes and links between them are learnt. First, we will describe the type of training used, called relative comparisons. Next, the two step learning process is described. The first step involves learning a suitable set of primitive wormholes. The second step then involves establishing the inter-wormhole network links.

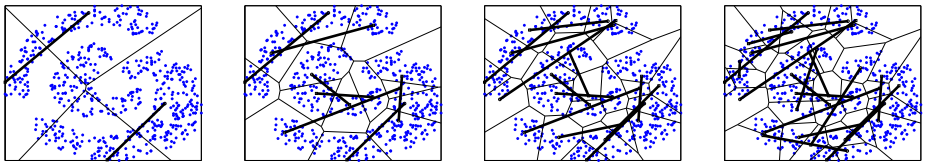
3.1 Training Data: Relative Comparisons

The type of training data we have chosen is called relative comparison triplets [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. The training dataset of N_T relative comparison triplets is defined as: $T_j = \{t_{ji}\}_{i=1}^{3}, j = 1...N_T$, where t_{j1} is closer to t_{j2} than t_{j3}. The entire training dataset is defined as $T = \{T_j\}_{j=1}^{N_T}$. Assuming an even number of training data, we randomly split the training data into two equal sized and mutually exclusive sets, $U = \{U_j\}_{j=1}^{N_U}$ and $V = \{V_j\}_{j=1}^{N_V}$ for learning the primitive wormholes and inter-wormhole network links respectively, where $N_V = N_U = N_T/2$. Each triplet for U and V is defined as $U_j = \{u_{ji}\}_{i=1}^{3}$ and $V_j = \{v_{ji}\}_{i=1}^{3}$ respectively. We will see how these examples are obtained in Section 4.

3.2 Learning Primitive Wormholes

The primitive wormholes are learnt through a selection process that is conceptually similar to Boosting [3], where the primitive wormhole is the equivalent of a weak classifier. Using the training database U, suitable primitive wormholes are identified and added into the wormhole network. Here, all primitive wormholes within the network are set such that they are "far away" from each other. Whilst this is suboptimal, it should be noted that this step is concerned with selecting a suitable set of primitive wormholes for building a distance function.

To generate candidate primitive wormholes, the triplets in U are used. A primitive wormhole K_j can is created by setting its two kernel centres to the "close" examples in the training triplet $K_j = \{u_{j1}, u_{j2}\}$, resulting in a set of N_U primitive wormholes for selection. Given a set of selected wormhole bases C, the training examples U, and their



2 wormholes 7 wormholes 12 wormholes 17 wormholes

Figure 3: Input space divided by primitive wormholes in the wormhole selection process.

weights P, the training error function $E(T,W,C)$ is defined as follows:

$$E(U,P,C) = \sum_{j=1}^{N_U} p_j G(U_j,C) \quad (3)$$

$$G(U_j,C) = \left\{ \begin{array}{ll} 1 & (W(u_{j1},u_{j2},C) > W(u_{j1},u_{j3},C)) \\ 0 & (W(u_{j1},u_{j2},C) < W(u_{j1},u_{j3},C)) \end{array} \right\} \quad (4)$$

where $G(T_j,C)$ is the individual error function for the j^{th} training triplet given a set of distance bases C, and W in the error function G is the wormhole function (Eq. 2). The learning algorithm revolves around a primitive wormhole selection loop. For each iteration, every primitive wormhole is considered a candidate to be added into the final wormhole set. The selected primitive wormhole is that with the lowest training error (Eq. 4) when added into this set. Figure 3 illustrates the selection process and how the input space is divided. The selection procedure is:

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

3.3 Learning the Inter-wormhole Network

In the previous section, the selected primitive wormholes were set such that they are "far away" from each other. This assumption may cause overfitting resulting in poor generalisation. To address this, we describe how to learn the correct inter-wormhole network links using the second half of the training data. Firstly, suitable links between two primitive wormholes can be established by exploiting near/far information from training triplets. To accumulate supporting evidence that two primitive wormholes should have a near or far link, two square matrices are introduced: the *near support matrix* ($O = o_{ij}, i, j = 1...N_B$) and the *far support matrix* ($P = p_{ij}, i, j = 1...N_B$), where the size of both matrices is the

number of wormholes selected (N_B). The near and far matrix off-diagonal elements count how many training triplets confirm the i^{th} wormhole and the j^{th} wormhole have a close or far link respectively. The appropriate elements in both near and far support matrices are updated based on the training triplets ($V_j = \{v_{ji}\}_{i=1}^{3}, j = 1...N_V$) as follows:

1. We firstly deal with updating the near support matrix. Initially, we determine which primitive wormholes catch the "near" examples in the triplet (v_{j1}, v_{j2}) using a part of Eq. 2: $c_{lm} = \arg\min_{c_{lm}}(K(v_{j1}, c_{lm})), c_{no} = \arg\min_{c_{no}}(K(v_{j2}, c_{no}))$,where the first example in the triplet (v_{j1}) is caught by wormhole c_{lm} (i.e. l^{th} wormhole's m^{th} entrance), and the second example is caught by wormhole c_{no}. The near support matrix elements o_{ln} and o_{nl} are both incremented.

2. The far support matrix is updated in a similar way. We now determine the primitive wormhole that catches the "far" example: $c_{rs} = \arg\min_{c_{rs}}(K(v_{j3}, c_{ik}))$, where the third example (v_{j3}) is caught by wormhole c_{rs}. Following this, the far support matrix element p_{rl} and p_{lr} are both incremented.

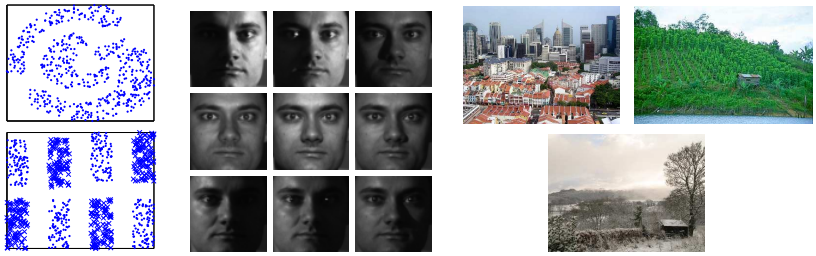
However, insufficient training data can cause ambiguous links for some pairs of primitive wormholes (both corresponding near and far support matrix elements are zero: $o_{ln} = p_{ln} = 0, l \neq n$). To address this, a method for link ambiguity resolution is proposed and works as follows: Suppose that wormhole C_l has an ambiguous link with wormhole C_n. We attempt to resolve this ambiguity by "enquiring" from other primitive wormholes unambiguously connected to C_l:

1. Determine if both wormholes should be closely connected. Wormholes close to C_l are determined using the near support matrix's l^{th} row with non-zero off-diagonal elements. The number of near wormholes and their indices for C_l are defined as N_L and $\{a\}_{i=1}^{N_L}$ respectively. We then enquire whether these wormholes are close to C_n by examining their respective link elements $o_{a_i n}, i = 1...N_L$ (i.e. elements in the n^{th} column of O). The maximum value ($o'_{ln} = \max(o_{a_i n}), i = 1...N_L$) is obtained and acts as a potential replacement for the ambiguous near link value o_{ln}.

2. Determine if both wormholes should be far apart. Similar to above, we determine if there are any wormholes far from C_l, but *close* to C_n instead, thus supporting C_n being far away from C_l. The corresponding number and indices of the far wormholes is defined as N_F and $\{b\}_{i=1}^{N_L}$. We then inspect near matrix elements $o_{b_i n}$. The maximum value ($p'_{ln} = \max(o_{b_i n}), i = 1...N_L$) will then act as the potential replacement for the far ambiguous link value p_{ln}. Finally, the link between C_l and C_n (i.e. I_{ln} from Eq. 2) will then be set as a near link if $o'_{ln} > p'_{ln}$ and vice versa.

4 Experiments and Results

We have tested the proposed distance learning method on three different problem domains: synthetic 2D data, face illumination detection and image retrieval of various image categories (Figure 4). An important point is that whilst each problem has its own kernel function, the rest of the learning method remains unchanged. It is also important to note that no labelled data was used in training. All the training data takes the form of relative comparisons described in Section 3.1.



(a) (b) (c)

Figure 4: The wormhole distance learning was applied to 3 different applications: (a) synthetic 2D data, (b) Differently illuminated faces from PIE database [5], (c) image distances between 3 categories (cities,jungles,winter scenes).

For each of these 3 applications, test experiments were performed in the following manner: Firstly, the obtained database described was randomly split into two equal partitions: training and test data. The relative comparisons triplets were generated from each training example by randomly picking two other examples, one from the same class and another from a different class. A single distance function using the proposed methods in Section 3 is then learnt from the training data. The types of kernels used for the distance function vary according to the application and is discussed in more detail later.

In order to quantify the overall performance of the resulting learnt distance functions, the test data examples were hand labelled with class data and separated into a number of groups of examples. Next, an exhaustive inter/intra-group distance calculation was performed. To obtain the intra-group distances, the distance of every point in a group to every other point in the same group is computed, for all groups. For the inter-group distances, the distance of every point in a group to every point in all other groups was calculated, and again repeated over all groups. Finally, the histograms of the inter and intra group distances were computed. For comparisons, similar distance histograms were obtained using the original local distance function.

Synthetic 2D Data: For this problem, we apply the distance learning function to two sets of synthetically generated 2D data, both non-linear and discontinuous in nature (Fig 4a). One set consists of three non-linear curved blobs. The second "extended-XOR" data set consists of 2 groups of data arranged in an "XOR"-like pattern. In total, 1200 points were generated for each dataset. For both problems, the proposed method is tasked with learning an appropriate distance function that will pull all points within the same group close together and push the remaining points far away. Here, a Euclidean kernel function is used. The resulting learnt distance wormholes for the 3 blob and extended-XOR problem can be seen in Figure 5a,d respectively.

The test results of the learnt distance functions is shown in Figure 5c for the 3 blob test and 5f for the extended-XOR problem. It can be seen that there is a clear separation between distances of points within a cluster to points outside the cluster, across all groups. In comparison, using Euclidean distance results in Figure 5b and e, where a large overlap between the inter and intra group distances exists, caused by nonlinear data.

Face Illumination Detection To test the distance learning method on more complex data with higher dimensionality, we applied it to the problem of detecting the illumination

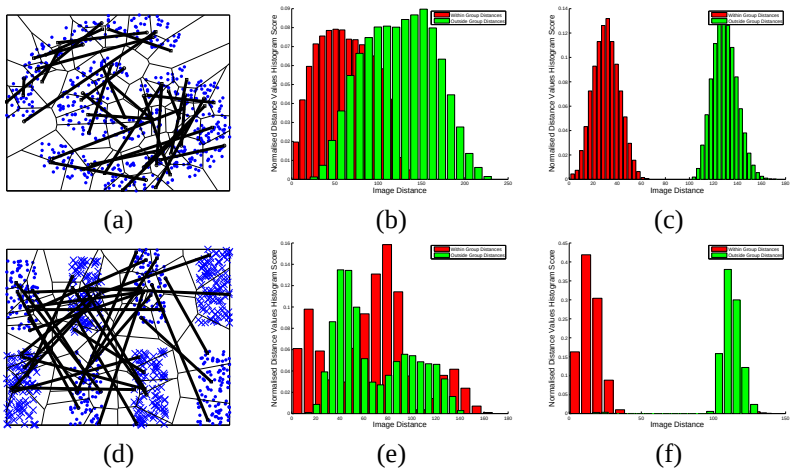


Figure 5: Learnt wormholes for 3 blob (a) and extended XOR (d). Distance histograms using (b,e) Euclidean and (c,f) wormhole-based method.

direction of faces. Faces of subjects under different illumination conditions from the PIE database were used, giving 21 different illumination directions and 66 subjects. The dimensionality of the face images was reduced to 20 using PCA. The kernel function for this problem is also Euclidean. The test distance histograms for the illumination detection in Figure 6b shows the ability for the distance function to generalise to unseen subjects. It can be seen that the wormhole distance has managed to separate a majority of faces within the same illumination group from the other test faces. On the other hand, Figure 6a shows that Euclidean distance alone is inadequate for distinguishing between different illumination groups. Figure 6c shows examples of distances between faces of the test subjects. In terms of false positive rates, the learnt distance method had a rate of 13% whilst the Euclidean distance had a rate of 35%.

Image Retrieval Finally, we have also carried out preliminary tests on learning distances between different images for the purpose of image retrieval. A database of different images belonging to 3 categories of cities, jungles/forests and winter landscapes were obtained. For each category, 15 images were obtained. For the distance between two images, detected line angles and colour segmentation were used. For colour segmentation, the observed colours in the image were transformed into the Munsell space [2] where pixel colours were converted into 11 basic colours. Each image is then represented by the histograms of the detected lines angles (H_A) and 11 basic colours in the image (H_C). The kernel function is then the sum of χ^2 differences between the corresponding histograms of an input image and the kernel centre (another example image). The test distance histograms in Figure 7a,b again shows the advantage of using the learnt distance measure over a simple available distance (e.g. the χ^2 histogram difference). Figure 7b shows the distance histograms of the learnt wormhole distance, where there is a more distinct separation of distances between images of the same category to images of other categories. When only the χ^2 distance measure is used, the separation is less apparent as shown in Figure 7a. Examples of the learnt image distance on test images can be seen in Figure 7c.

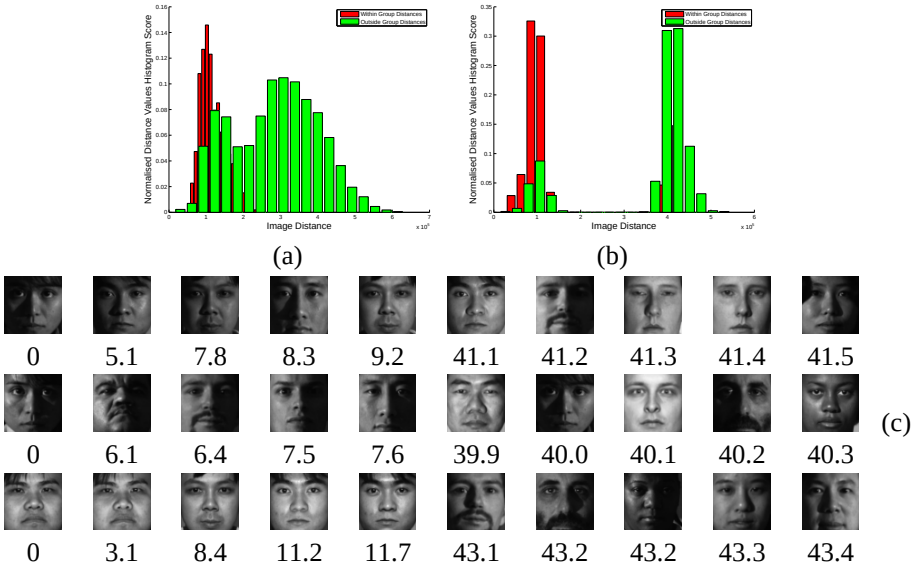


Figure 6: Results of face illumination detection. Distance histograms using Euclidean distance (a) and learnt wormhole distance (b). Examples of learnt wormhole distances (numbers) between face images (base image on the first column) (c).

5 Conclusions

In this paper, a novel distance function based on the concept of wormholes was proposed. To improve generalisation, an inter-wormhole network was created, bringing different wormholes close together or far away. We have also described two novel learning methods: 1) allows us to select appropriate wormholes to form a distance function and 2) build the necessary links between different selected wormholes to improve generalisation. Importantly, both learning methods only assume the availability of relative comparisons training data, removing the need for labelled data. We have shown how the same learning methods can be used to build the appropriate distance functions for three very different applications ranging from non-linear synthetic 2D data to vision problems dealing with face illumination detection and image retrieval.

Acknowledgements

The work and contribution here is 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.

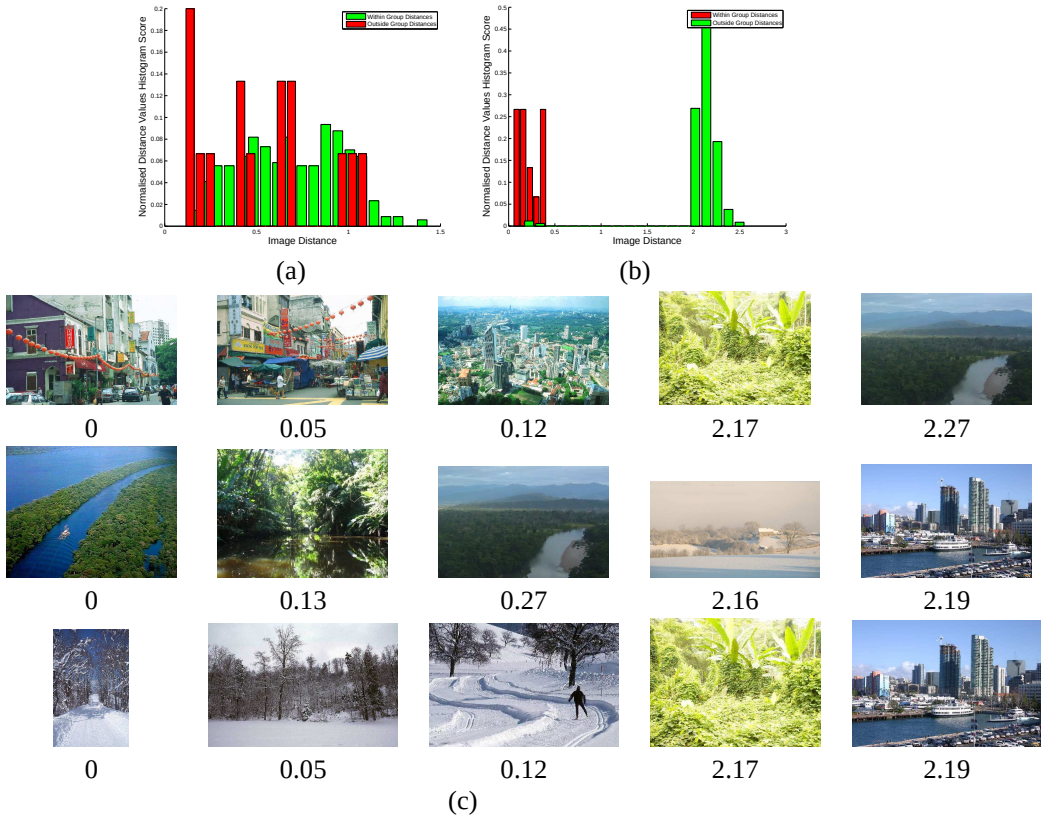


Figure 7: Images' inter/intra group distance histograms using kernel distance (a) and learnt wormhole distance (b). Image distances using the wormhole method (c).

[2] B. Berlin and P. Kay. *Basic Color Terms: Their Universality and Evolution*. University of California, 1991.

[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] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, December 2003.

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