

Regularising the Ricci Flow Embedding

Weiping Xu, Edwin R. Hancock, and Richard C. Wilson

Dept. of Computer Science

University of York, UK

{elizaxu, erh, wilson}@cs.york.ac.uk

Abstract. This paper concerns the analysis of patterns that are specified in terms of non-Euclidean dissimilarity or proximity rather than ordinal values. In prior work we have reported a means of correcting or rectifying the similarities so that the non-Euclidean artifacts are minimized. This is achieved by representing the data using a graph, and evolving the manifold embedding of the graph using Ricci flow. Although the method provides encouraging results, it can prove to be unstable. In this paper we explore how this problem can be overcome using a graph regularisation technique. Specifically, by regularising the curvature of the manifold on which the graph is embedded, then we can improve both the stability and performance of the method. We demonstrate the utility of our method on the standard “Chicken pieces” dataset and show that we can transform the non-Euclidean distances into Euclidean space.

1 Introduction

Dissimilarity representations [1] provide a powerful and natural way of capturing the relationships between objects that are not characterised by ordinal measurements or feature vectors. The idea is to use a pairwise dissimilarity (or proximity) measure [2,3] to describe the properties of objects in terms of their attribute differences. Examples of such representations are provided by weighted proximity graphs. The advantages of such a representation are that if characterised in terms of a dissimilarity matrix, then pattern matching can be effected without the need for explicit alignment. However, the dissimilarities are quite frequently non-Euclidean and this prevents the use of many geometrically based learning techniques.

One way to overcome these problems is to represent the dissimilarity data using a weighted graph, and to embed the graph on a manifold. This produces a vectorial representation of the data by projecting dissimilarity data into a fixed-dimensional vector space. Examples of this approach include multidimensional scaling (MDS) [4], Isomap [5], locally linear embedding [6] and the Laplacian eigenmap embedding [7]. The common aim is to locate a low-dimensional representation. In order to apply non-Euclidean dissimilarity data with traditional geometric learning techniques, we must attempt to rectify the data so as to minimize the non-Euclidean artifacts. One route is to consider the positive definite subspace of the distances [8]. An alternative route adopted by Pekalska et al. [9] is to add a suitable constant to the squared off-diagonal elements of the dissimilarity matrix. It is equivalent to adding a certain constant to all eigenvalues of the related Gram matrix, and thus compensating for the effect of the negative eigenvalues, while maintaining the same eigenvector structure.

In prior work [10] we have shown how to correct the dissimilarity data, giving a set of new Euclidean distances. The method uses Ricci flow on a constant curvature Riemannian manifold to evolve the distance measures. This is effected by updating the curvatures on the edges of the graph representing the data. Unfortunately, the method can prove unstable due to local fluctuations in edge curvature. To overcome this problem, in this paper we show how to stabilise the method by regularising the curvatures of the embedded graph. To do this we use the heat kernel to smooth the curvatures on the edges. The result shows both improved numerical stability and lower classification error in the embedded space.

2 Embedding Non-Euclidean Data

In this paper we are concerned with embedded data represented in terms of pairwise dissimilarities or distances, and in particular the case where the data is non-Euclidean. Our overall aim is to rectify a given set of non-Euclidean dissimilarity data so as to make them more Euclidean. One way to gauge the degree to which a pairwise distance matrix contains non-Euclidean artefacts is to analyse the properties of its centralised Gram matrix. For an $N \times N$ symmetric pairwise dissimilarity matrix D with the pairwise distance as elements, the centered Gram matrix $G = -\frac{1}{2}JD^2J$, where D^2 is element-wise squaring of elements in D , $J = I - \frac{1}{N}11^T$ is the centering matrix and 1 is the all-ones vector of length N . The degree to which the distance matrix departs from being Euclidean can be measured by using the relative mass of negative eigenvalues or “negative eigenfraction” $F_{eigS} = \sum_{\lambda_i < 0} |\lambda_i| / \sum_{i=1}^N |\lambda_i|$ [11]. This measure is zero for Euclidean distances and increases as the distance becomes increasingly non-Euclidean.

The kernel embedding is obtained from the centered Gram matrix using the factorisation $G = YY^T$, where Y is the $N \times N$ matrix with the embedded co-ordinates of the data as columns. To determine whether the Gram matrix is positive semi definite [11], we perform the eigendecomposition $G = \Phi\Lambda\Phi^T$ on the Gram matrix, where $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$ is the diagonal matrix with the ordered eigenvalues as elements and $\Phi = (\phi_1 | \dots | \phi_N)$ is the eigenvector matrix with the ordered eigenvectors ϕ_1, \dots, ϕ_N as columns. In terms of the eigenvalues and eigenvectors, the matrix of embedded co-ordinates is given by $Y = \Phi\sqrt{\Lambda}$ where the eigenvalues Λ are positive. In Isomap embedding, the dimension and the number of nearest neighbors are estimated to be the optimal values by looking at the residue variances[5].

3 Ricci Flow

Our aim is to develop a method that can be used to rectify the non-Euclidean artefacts in such a dissimilarity matrix. The approach is as follows. Firstly, we consider the objects of interest to be represented by points on a manifold, and the given dissimilarities to be the geodesic distances on the manifold between these points (geodesic distances). For an arbitrary set of non-Euclidean similarities the manifold will be curved. By contrast, a Euclidean space will be flat and the geodesic and Euclidean distances will be identical. Our task is then to remove the curvature from the manifold to create a corrected set of Euclidean distances. We achieve this by evolving the manifold using Ricci flow.

The Ricci flow [12] evolves a manifold so that the rate of change of the metric tensor is controlled by the Ricci curvature. Essentially, this is an analogue of a diffusion process for a manifold. The geometric evolution equation is:

$$\frac{dg_{ij}}{dt} = -2R_{ij} \tag{1}$$

where g_{ij} is the metric tensor of the manifold and R_{ij} is the Ricci curvature.

We model the embedding manifold as consisting of a set of local patches with individual constant Ricci curvatures. These patches can be either elliptic (of positive sectional curvature) or hyperbolic (of negative sectional curvature). It is straightforward to re-express the Ricci flow in terms of the sectional curvature K :

$$\frac{dK}{dt} = \begin{cases} -2K^2 & \text{elliptic hypersphere,} \\ 2K^2 & \text{hyperbolic space.} \end{cases} \tag{2}$$

Under this evolution, the curvature moves towards zero for both types of patch, flattening the manifold. The solution of the differential equation is straightforward. Commencing with the initial conditions $K = K_0$ at time $t = 0$, then at time t we have

$$K = \frac{K_0}{1 \pm 2K_0t} \tag{3}$$

with the positive sign for the elliptic space.

4 Curvature Computation

Our aim is to evolve a non-Euclidean dissimilarity measure into a Euclidean one using the Ricci flow described in the previous section. We commence by representing the dissimilarity data using a weighted graph $G = (V, E, D)$, where the node set V represents the set of objects and the edges E are weighted with the pairwise dissimilarities. We embed the graph onto a manifold so that the geodesic distance $d_g(u, v), (u, v) \in E$ between the positions of the nodes u and v is equal to the dissimilarity on the edges. Let \mathbf{y}_u be the embedded co-ordinates of the node $u \in V$ and $Y = (\mathbf{y}_1 | \dots | \mathbf{y}_{|V|})$ be the matrix with the embedded co-ordinates as columns. Under this embedding the edges acquire a curvature determined by the difference between geodesic distance (dissimilarity) $d_G(u, v)$ and Euclidean distance $d_E(u, v) = \sqrt{(\mathbf{y}_u - \mathbf{y}_v)^T (\mathbf{y}_u - \mathbf{y}_v)}$. The Ricci flow, modifies the Gaussian curvatures on the edges, so as to flatten the manifold. Adopted from [13] we use a Euclidean embedding of the points and use the difference between the geodesic distance d_G on the manifold (from the similarity or dissimilarity matrix) and the Euclidean distance in the embedded space d_E to compute the curvature. We compare experimental results for embeddings obtained with both Isomap [5] and the kernel embedding in Section 7. Lindman and Caelli [14] give the relationship between the two distances on elliptic, hyperbolic and Euclidean constant curvature manifolds as

$$d_E = \begin{cases} \frac{2}{K^{\frac{1}{2}}} \sin(\frac{K^{\frac{1}{2}}}{2} d_G) & \text{Elliptic,} \\ \frac{2}{|K|^{\frac{1}{2}}} \sinh(\frac{|K|^{\frac{1}{2}}}{2} d_G) & \text{Hyperbolic,} \\ d_G & \text{Euclidean.} \end{cases}$$

However, the adopted curvature approximations used only hold for small curvatures. In the data under study here, we find that the curvatures are too large for these approximations to hold. We therefore use it as the initialization and estimate curvature from Equation 4 using Newton’s method. Taking the curvature in an elliptic space as an example, the Newton iteration is

$$K_{n+1}^{\frac{1}{2}} = K_n^{\frac{1}{2}} - \frac{K_n^{\frac{1}{2}} d_E - 2 \sin \frac{K_n^{\frac{1}{2}}}{2} d_G}{d_E - d_G \cos \frac{K_n^{\frac{1}{2}}}{2} d_G} \tag{4}$$

Finally, we can compute new geodesic distances for the points based on the updated curvature. We keep the Euclidean distance between the points fixed, while updating the curvature. The updated geodesic distance under the new Gaussian curvature can be represented in terms of the old geodesic distance at the previous iteration. The update equation for the geodesic distance is

$$d_{G_{n+1}} = \begin{cases} \frac{2}{K_{n+1}^{\frac{1}{2}}} \arcsin \left(\frac{K_{n+1}^{\frac{1}{2}}}{K_n^{\frac{1}{2}}} \sin \left(\frac{K_n^{\frac{1}{2}}}{2} d_{G_n} \right) \right) & \text{elliptic hypersphere} \\ \frac{2}{|K_{n+1}|^{\frac{1}{2}}} \operatorname{arcsinh} \left(\frac{|K_{n+1}|^{\frac{1}{2}}}{|K_n|^{\frac{1}{2}}} \sinh \left(\frac{|K_n|^{\frac{1}{2}}}{2} d_{G_n} \right) \right) & \text{hyperbolic space} \end{cases} \tag{5}$$

This equation can be applied to each element of the dissimilarity matrix in turn.

5 The Algorithm

Given a set $X = \{x_1, \dots, x_N\}$ of N objects and a dissimilarity measure d , a dissimilarity representation is an $N \times N$ matrix D_G . The following algorithm can be used to rectify the distance matrix from being non-Euclidean to Euclidean.

Begin with a pairwise distance matrix $D_G^{(0)}$,

1. Embed the objects in a Euclidean space using either Isomap or the kernel embedding. In the embedded space compute the Euclidean distances d_E .
2. From the geodesic distance d_G and Euclidean distance d_E , compute the constant curvature space with curvature K for a pair of objects using Equation 4.
3. Update the Gaussian curvature with a small time step using Equation 3.
4. Obtain the new geodesic distance $d_{G_{n+1}}$ from the previously available geodesic distance matrix together with the curvatures under a fixed Euclidean distance using Equation 5.
5. Obtain the updated distance matrix $D_G^{(1)}$ containing rectified geodesic distances between objects, and repeat from step 1 until D_G is Euclidean, that is its centered Gram matrix has no negative eigenvalues.

6 Regularizing Curvature

As posed above, the Ricci flow embedding updates the Gaussian curvature separately for each individual edge. This is because we use piecewise constant curvature manifolds

for each edge. This places no constraint on the smoothness of the manifold, and this can lead to numerical instability in the embedding. Graph regularization provides a way to smooth data samples over a graph and overcome the numerical stability problems. One such regularization process is a graph diffusion. A diffusion process is analogous to the flow of heat, which flows from high to low concentrations, and over time creates a smooth distribution of heat. In a similar way, a diffusion of a function on the graph will create a smoother function. The diffusion is defined in terms of a random walk on the edges of the graph[15], and is represented by the diffusion (or heat) kernel:

$$H = \exp(-Lt) \tag{6}$$

The evolution of a function under this kernel is simply

$$f(t) = H(t)f(0) \tag{7}$$

The evolution is ‘mass-preserving’ in the sense that the sum of the values of the function over vertices is preserved.

We can use this process for smoothing curvatures before the application of the Ricci flow, to remove extreme values. However, our curvatures are defined pairs of objects and we therefore need to construct a graph which has vertices corresponding to object-pairs and edges describing a neighbourhood structure of these pairs. We construct this graph as follows. Firstly, we build the nearest-neighbours graph of the objects $G = \{V, E\}$. Each vertex represents an object u and an undirected edge E_{uv} exists if u is in the n nearest neighbours of v or v is in the n nearest neighbours of u . We then construct the dual of this graph $G_D = \{V_D, E_D\}$; each edge of the original graph becomes a vertex V_{uv} and an edge exist between two vertices if they share a common vertex from the original graph. In the dual graph, each vertex represents a pair of objects and the edges reflect the neighbourhood structure of the pairs. We can then define the curvature between object pairs as a function over the vertices of this graph and apply the diffusion kernel.

We therefore add an additional step in which we smooth the Gaussian curvatures over the dual of the nearest neighbor graph prior to performing the Ricci flow updating of the curvatures. All of the remaining steps of the algorithm remain as above. The following steps shows how to smooth Gaussian curvatures over the nearest neighbour edges.

Commence with initial Gaussian curvatures K from step 2 above,

1. Construct the n nearest neighbour graph over the available dissimilarity data. Node u and v are connected by an edge if u is among n nearest dissimilarity neighbors of v or v is among n nearest dissimilarity neighbors of u .
2. Construct the dual graph of the nearest neighbour graph. Each edge in the nearest neighbour graph is a vertex of the dual graph. If two edges in the nearest neighbour graph share a one common vertex, then the corresponding two vertices in the dual graph are connected by an edge.
3. Obtain the updated and regularised curvature K . Suppose that \hat{L} is the normalized Laplacian of the dual nearest neighbour graph, then the heat-kernel of the dual graph is $\exp[-\hat{L}t]$. If V_D is the node-set of the dual graph, then we construct a

vector \mathbf{K} of Gaussian curvatures $\mathbf{K} = (K_1, \dots, K_{|V_D|})^T$. The vector of regularised Gaussian curvatures after heat kernel smoothing is $\mathbf{K}_{reg} = \exp[-\hat{L}t]\mathbf{K}$.

In summary, the above approach commences from a nearest neighbor graph over the dissimilarity matrix, and then constructs the dual graph where a node corresponds to an edge in the original graph. The heat kernel on the dual graph smooths the curvatures on the original nearest neighbour graph.

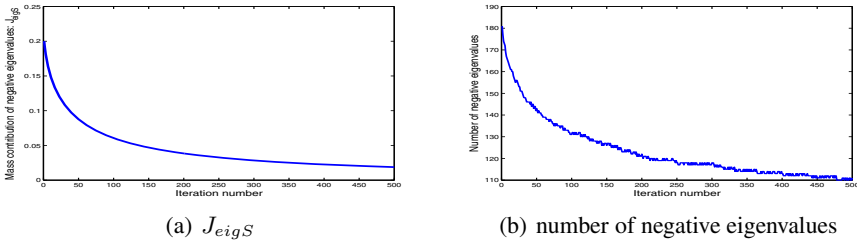


Fig. 1. (a) is the negative eigenfraction during iteration. (b) is the number of negative eigenvalues during iteration.

7 Experiments

We use the well known “Chicken pieces” dataset [8] for experimentation. The data-set concerns classifying binary images of a different types of chicken joint into shape-classes. It contains 446 binary images falling into five shape classes, namely a) breast (96 examples), b) back (76 examples), c) thigh and back (61 examples), d) wing (117 examples) and e) drumstick (96 examples). The data exists in the form of a set of non-Euclidean shape dissimilarity matrices, generated using different settings for the parameters in which, L is the length of straight line segments of chicken contours and C is the insertion and deletion costs for computing edit distances between boundary segments. Our experimental results are for the dissimilarity data with $C = 45$ and $L = 5, 10, 15, 20, 25$ and 30 .

The negative eigenfraction for the Chicken Pieces data with $L = 5.0, C = 45$ is shown in Figure 1 as the manifold evolves with iteration number. As the curvatures are updated both the negative eigenfraction and the negative eigenvalues decrease, indicating that the dissimilarity measure becomes increasingly Euclidean. Figure 2 shows the curvatures as a function of distances obtained using the kernel embedding and Isomap embedding. It demonstrates how the Ricci flow process affects distances commencing from the two embedding methods with and without regularization. It indicates that the embedding method affects the magnitude of curvatures. The figure also shows that the Kernel embedding preserves the global distances. Here, the larger the distances, the smaller the curvatures. On the other hand, the Isomap embedding preserves some of the local distances. This maybe the due to the fact that the chicken pieces data does not reside on simple manifold such as Swiss roll. The embedding method determines

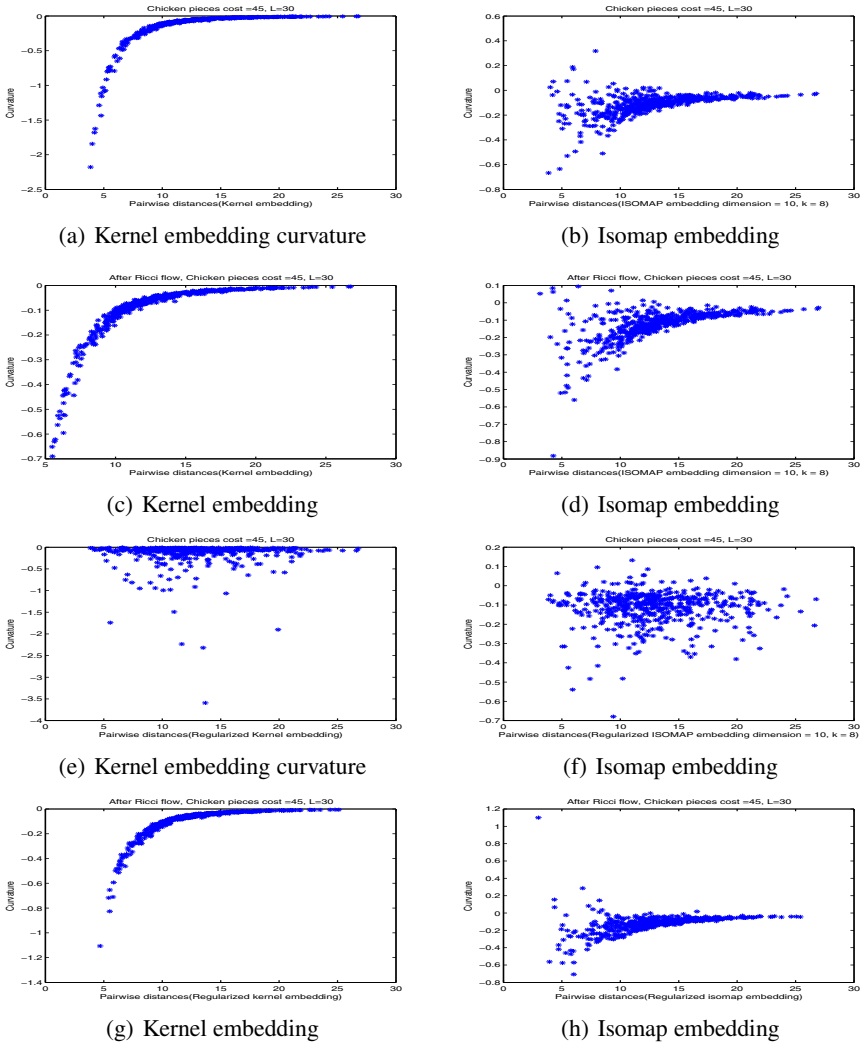


Fig. 2. (a) and (b) are initial edge curvatures for the kernel and Isomap embeddings. (c) and (d) are edge curvatures after Ricci Flow for the kernel and Isomap embeddings. (e) and (f) are initial regularised edge curvatures for the kernel and Isomap embeddings. (g) and (h) are edge curvatures after Ricci Flow for the kernel and Isomap embeddings.

the magnitude of curvatures. From our Ricci flow curvature updating process, the larger the magnitude of the original curvatures, the larger the curvature reduction in the update process. As a result in the case of the kernel embedding, those locations associated with large curvature expand more rapidly than those associated with small curvatures. In other words, the initial smaller distances expand more rapidly than larger distances.

This effect can be observed from Figure 2(a) and Figure 2(c). As a result, it disrupts the local pattern of distances without influencing the larger ones.

During the regularization step, the curvatures are smoothed over nearest neighbour edges. The result is to reduce local curvature fluctuations, and this may reduce some locally large curvature values. Figure 2(e) shows that when regularisation is used, the curvatures are smoothed over local distance scales compared to the initial curvatures in Figure 2(a). Hence the local distance structure is preserved under the embedding, and this is demonstrated in Figure 2(g). As a result the regularization step preserves local distances and stabilizes the local structure.

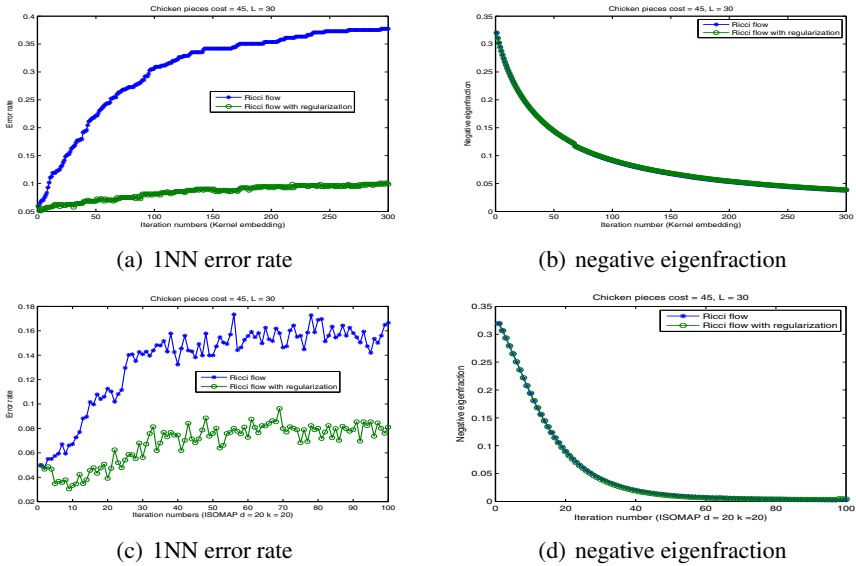


Fig. 3. (a) is the 1NN error rate with and without the regularization step using the kernel embedding during iteration. (b) is the negative eigenfraction with and without the regularization step using the kernel embedding during iteration. (c) is the 1NN error rate with and without the regularization step using the Isomap embedding during iteration. (d) is the negative eigenfraction with and without the regularization step using the Isomap embedding during iteration.

Next, we turn our attention to the effect of regularisation and the choice of embedding on the results of classification. The classification results were obtained with the 1-NN classifier and 10-fold cross validation. In Figure 3 we compare the 1-NN error rates and the negative eigenfraction obtained with regularised and unregularised versions of Ricci flow on the two embedding schemes. The first point to note is that for both the kernel embedding and Isomap, we obtain better classification results when heat kernel regularisation is used. However, in each case the application of the Ricci flow scheme causes the classification error to increase with iteration number. However, in the case of the regularised kernel embedding, the effect is smallest. Finally, the choice of embedding scheme strongly affects the rate of decrease of the negative eigenfraction, with

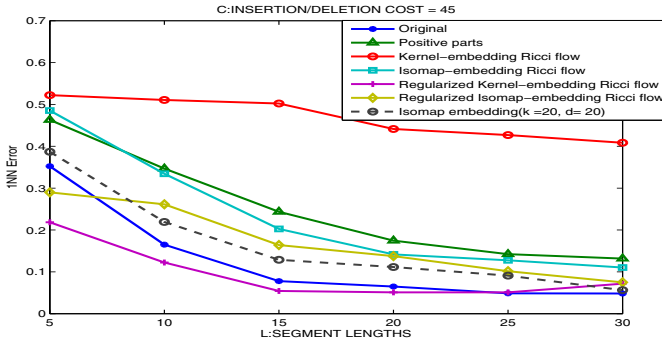


Fig. 4. Error rate from 1NN

Isomap giving a faster rate of decrease with iteration number than the kernel embedding. However, for both embedding schemes the use of regularisation has little effect on the rate of decrease.

Finally, we have compared our results with the known manifold embedding technique Isomap and those obtained using some alternative non-Euclidean distance rectification procedures. The methods explored were a) using the original distances, b) projecting onto the positive subspace and taking the distance here, unregularised Ricci flow on c) the kernel embedding and d) the Isomap embedding, regularised Ricci flow on e) the kernel embedding and f) the Isomap embedding. Figure 4 shows the 1-NN error rate as function of the shape parameter L (the segment length). The best results are obtained with Ricci flow on the regularised kernel embedding. All of the remaining methods give poorer results than applying the classifier to the original distance data.

8 Conclusion

In this paper we have explored how to evolve a non-Euclidean dissimilarity measure into a Euclidean one using Ricci flow. We commence by representing the dissimilarity data using a weighted graph, where the nodes represent objects and the edge weights dissimilarities between objects. We embed the graph onto a manifold so that the geodesic distance between nodes is equal to the dissimilarity on the edges. Under the embedding the edges acquire a curvature determined by the difference between geodesic distance (dissimilarity) and Euclidean distance. The Ricci flow, modifies the Gaussian curvatures on the edges, so as to flatten the manifold. We explore in depth the effect of stabilising this process by using heat-kernel regularisation to smooth the Gaussian curvatures prior to evolving the manifold.

We apply our method to the Chicken Pieces data. When applied without regularisation, although the distance measures can be transformed into a Euclidean space there is some loss of discriminating power and the classifier performance degrades. The loss of information is attributable to the effect of the Ricci evolution process which acts independently on each edge and ignores the local structure of the manifold. When heat

kernel regularization is used the ranking of distance measures is preserved, and better performance is achieved. Although the method degraded the error obtained with a 1NN classifier, it does deliver data in a form where geometric classification methods can be applied to the data. The Ricci flow evolution minimise the curvatures, when the curvatures reach zero, then the geodesic and the Euclidean distances are equal and the negative eigenfraction is zero.

As the embedding methods affects the magnitude of curvatures a lot, one way to develop our work is to reduce the reliance on the embedding methods by using spherical embedding and tangent space projection. Another direction is to develop incremental learning, in which new points can be mapped on the manifold, as our current method is performed in a batch mode, i.e., all training points are processed simultaneously.

Acknowledgements

This work was supported by the EU FET project SIMBAD (213250). Edwin Hancock was supported by a Royal Society Wolfson Research Merit Award.

References

1. Pekalska, E., Duin, R.P.W.: Beyond traditional kernels: classification in two dissimilarity-based representation spaces. *IEEE Transactions on Systems Man and Cybernetics-Part C* 38(6) (November 2008)
2. Sanfeliu, A., Fu, K.S.: A distance measure between attributed relational graphs for pattern recognition. *IEEE transactions on systems, man, and cybernetics* 13(3), 353–362 (1983)
3. Bunke, H.: A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters* 19(3-4), 255–259 (1998)
4. Borg, I., Groenen, P.: *Modern multidimensional scaling: Theory and applications*. Springer, Heidelberg (2005)
5. Tenenbaum, J., Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 2319–2323 (2000)
6. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding (2000)
7. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems* 1, 585–592 (2002)
8. Duin, R.P.W., Pekalska, E., Harol, A., Lee, W.J., Bunke, H.: On euclidean corrections for non-euclidean dissimilarities. In: *SSPR/SPR*, pp. 551–561 (2008)
9. Pekalska, E., Duin, R., Gunter, S., Bunke, H.: On not making dissimilarities euclidean. *Lecture notes in computer science*, pp. 1145–1154 (2004)
10. Xu, W., Hancock, E.R.W.: Rectifying non-euclidean similarity data using ricci flow embedding. In: *To appear ICPR 2010* (August 2010)
11. Pekalska, E., Harol, A., Duin, R., Spillmann, B., Bunke, H.: Non-euclidean or non-metric measures can be informative, pp. 871–880 (2006)
12. Chow, B., Luo, F.: Combinatorial Ricci flows on surfaces. *J. Differential Geom.* 63(1), 97–129 (2003)
13. ElGhawalby, H., Hancock, E.R.: Measuring graph similarity using spectral geometry. In: *ICIAR*, pp. 517–526 (2008)
14. Lindman, H., Caelli, T.: Constant curvature riemannian scaling. *Journal of Mathematical Psychology* 17, 89–109 (1978)
15. Kondor, R., Lafferty, J.: Diffusion kernels on graphs and other discrete structures. In: *Proceedings of the ICML*, pp. 315–322 (2002)