

# Characterising Graphs using the Heat Kernel

Bai Xiao, Richard C. Wilson and Edwin R. Hancock,  
Department of Computer Science, University of York,  
York YO1 5DD, UK.

## Abstract

The heat-kernel of a graph is computed by exponentiating the Laplacian eigen-system with time. In this paper, we study the heat kernel mapping of the nodes of a graph into a vector-space. Specifically, we investigate whether the resulting point distribution can be used for the purposes of graph-clustering. Our characterisation is based on the covariance matrix of the point distribution. We explore the relationship between the covariance matrix and the heat kernel, and demonstrate the eigenvalues of the covariance matrix are found by exponentiating the Laplacian eigenvalues with time. We apply the technique to images from the COIL database, and demonstrate that it leads to well defined graph clusters.

## 1 Introduction

One of the problems that arises in the manipulation of large amounts of graph data is that of characterising the topological structure of individual graphs. One of the most elegant ways of doing this is to use the spectrum of the Laplacian matrix [6]. For instance Shokoufandeh *et al* [10] have used topological spectra to index tree structures and Luo *et al* [3] have used the the spectrum of the adjacency matrix to construct pattern spaces for graphs. One way of viewing these methods is that of constructing a low-dimensional feature-space that captures the topological structure of the graphs under study.

An interesting alternative to using topological information to characterise graphs is to embed the nodes of a graph in a vector space, and to study the distribution of points in this space. Broadly speaking there are three ways in which the problem has been addressed. First, the graph can be interpolated by a surface whose genus is determined by the number of nodes, edges and faces of the graph. Second, the graph can be interpolated by a hyperbolic surface which has the same pattern of geodesic (internode) distances as the graph [1]. Third, a manifold can be constructed whose triangulation is the simplicial complex of the graph [2].

In the pattern analysis community, there has recently been renewed interest in the use of embedding methods motivated by graph theory. One of the best known of these is ISOMAP [8]. Here a neighborhood ball is used to convert data-points into a graph, and Dijkstra's algorithm is used to compute the shortest (geodesic) distances between nodes. By applying multidimensional scaling (MDS) to the matrix of geodesic distances the manifold is reconstructed. The resulting algorithm has been demonstrated to locate well-formed manifolds for a number of complex data-sets. Related algorithms include locally linear embedding which is a variant of PCA that restricts the complexity of the input data using a nearest neighbor graph [11], and the Laplacian eigenmap that constructs an

adjacency weight matrix for the data-points and projects the data onto the principal eigenvectors of the associated Laplacian matrix (the degree matrix minus the weight matrix) [4]. Collectively, these methods are sometimes referred to as manifold learning theory.

Recently, Lebanon and Lafferty [9] have taken the study one step further by using the heat-kernel to construct statistical manifolds that can be used for inference and learning tasks. The heat kernel is found by exponentiating the Laplacian eigen-system with time. There are a number of different invariants that can be computed from the heat-kernel. Asymptotically for small time, the trace of the heat kernel [6] (or the sum of the Laplacian eigenvalues exponentiated with time) can be expanded as a rational polynomial in time, and the co-efficients of the leading terms in the series are directly related to the geometry of the manifold. For instance, the leading co-efficient is the volume of the manifold, the second co-efficient is related to the Euler characteristic, and the third co-efficient to the Ricci curvature. The zeta-function (i.e. the sum of exponentials found by raising the eigenvalues to a non-integer power) for the Laplacian also contains geometric information. For instance its derivative at the origin is related to the torsion tensor for the manifold. Finally, Colin de Verdiere has shown how to compute geodesic invariants from the Laplacian spectrum [5].

The aim in this paper is to investigate whether the heat kernel can be used for the purposes of embedding the nodes of a graph in a vector space. We use the heat kernel to map nodes of the graph to points in the vector space. In other words, we perform kernel PCA on the graph heat-kernel. We provide an analysis which shows how the eigenvalues and eigenvectors of the covariance matrix for the point distribution resulting from the kernel mapping are related to those of the Laplacian. Based on this analysis we explore how the covariance matrix eigenvalues can be used for the purposes of characterising and clustering the graphs.

## 2 Heat Kernels on Graphs

In this section, we develop a method for approximating the geodesic distance between nodes by exploiting the properties of the heat kernel. To commence, suppose that the graph under study is denoted by  $G = (V, E)$  where  $V$  is the set of nodes and  $E \subseteq V \times V$  is the set of edges. Since we wish to adopt a graph-spectral approach we introduce the adjacency matrix  $A$  for the graph where the elements are

$$A(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

We also construct the diagonal degree matrix  $D$ , whose elements are given by  $D(u, u) = \sum_{v \in V} A(u, v)$ . From the degree matrix and the adjacency matrix we construct the Laplacian matrix  $L = D - A$ , i.e. the degree matrix minus the adjacency matrix. The normalised Laplacian is given by  $\hat{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ . The spectral decomposition of the normalised Laplacian matrix is  $\hat{L} = \Phi \Lambda \Phi^T$ , where  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{|V|})$  is the diagonal matrix with the ordered eigenvalues as elements and  $\Phi = (\phi_1 | \phi_2 | \dots | \phi_{|V|})$  is the matrix with the ordered eigenvectors as columns. Since  $\hat{L}$  is symmetric and positive semi-definite, the eigenvalues of the normalised Laplacian are all positive. The eigenvector associated with the smallest non-zero eigenvalue is referred to as the Fiedler-vector. We are interested in the heat equation associated with the Laplacian, i.e.  $\frac{\partial h_t}{\partial t} = -\hat{L} h_t$ , where  $h_t$  is the heat

kernel and  $t$  is time. The heat kernel can hence be viewed as describing the flow of information across the edges of the graph with time. The rate of flow is determined by the Laplacian of the graph. The solution to the heat equation is found by exponentiating the Laplacian eigen-spectrum, i.e.  $h_t = \Phi \exp[-t\Lambda] \Phi^T$ . When  $t$  tends to zero, then  $h_t \simeq I - \hat{L}t$ , i.e. the kernel depends on the local connectivity structure or topology of the graph. If, on the other hand,  $t$  is large, then  $h_t \simeq \exp[-t\lambda_m] \phi_m \phi_m^T$ , where  $\lambda_m$  is the smallest non-zero eigenvalue and  $\phi_m$  is the associated eigenvector, i.e. the Fiedler vector. Hence, the large time behavior is governed by the global structure of the graph.

## 2.1 Heat Kernel Embedding

We use the heat kernel to map the nodes of the graph into a vector-space. Let  $Y$  be the  $|V| \times |V|$  matrix with the vectors of co-ordinates as columns. The vector of co-ordinates for the node indexed  $u$  is hence the  $u^{\text{th}}$  column of  $Y$ . The co-ordinate matrix is found by performing the Young-Householder decomposition  $h_t = Y^T Y$  on the heat-kernel. Since  $h_t = \Phi \exp[-\Lambda t] \Phi^T$ ,  $Y = \exp[-\frac{1}{2}\Lambda t] \Phi^T$ . Hence, the co-ordinate vector for the node indexed  $u$  is

$$y_u = (\exp[-\frac{1}{2}\lambda_1 t] \phi_1(u), \exp[-\frac{1}{2}\lambda_2 t] \phi_2(u), \dots, \exp[-\frac{1}{2}\lambda_{|V|} t] \phi_{|V|}(u))^T \quad (2)$$

The kernel mapping  $\mathcal{M} : V \rightarrow \mathcal{R}^{|V|}$ , embeds each node on the graph in a vector space  $\mathcal{R}^{|V|}$ . The heat kernel  $h_t = Y^T Y$  can also be viewed as a Gram matrix, i.e. its elements are scalar products of the embedding co-ordinates. Consequently, the kernel mapping of the nodes of the graph is an isometry. The squared Euclidean distance between the nodes  $u$  and  $v$  is given by

$$\begin{aligned} d_E(u, v)^2 &= (y_u - y_v)^T (y_u - y_v) = \sum_{i=1}^{|V|} \exp[-\lambda_i t] (\phi_i(u) - \phi_i(v))^2 & (3) \\ &= h_t(u, u) + h_t(v, v) - 2h_t(u, v) & (4) \end{aligned}$$

## 2.2 Kernel PCA

One very simple way to characterise the heat-kernel mapping is to study the properties of the covariance matrix of the point-set generated by the kernel mapping. To construct the covariance-matrix for the kernel mapping, we commence by computing the mean co-ordinate vector. The components of the mean co-ordinate vector are found by averaging the elements in the rows of  $Y$ . The mean co-ordinate vector is given by

$$\hat{y} = \frac{1}{|V|} Y e = \frac{1}{|V|} \exp[-\frac{1}{2}\Lambda t] \Phi^T e$$

where  $e = (1, 1, \dots, 1)^T$  is the all ones vector of length  $|V|$ . Subtracting the mean from the kernel mapping co-ordinates, the matrix of centred co-ordinates is

$$Y_C = Y - \frac{1}{|V|} Y e e^T = \exp[-\frac{1}{2}\Lambda t] \Phi^T (I - \frac{1}{|V|} e e^T) = \exp[-\frac{1}{2}\Lambda t] \Phi^T M^T,$$

where  $M^T = (I - \frac{1}{|V|}ee^T)$  is the data centering matrix. The covariance matrix is

$$\Sigma = \frac{1}{|V|}Y_C Y_C^T = \frac{1}{|V|} \exp[-\frac{1}{2}\Lambda t] \Phi^T M^T M \Phi \exp[-\frac{1}{2}\Lambda t].$$

Hence, we can write  $\Sigma = \frac{1}{|V|}C^T C$  where  $C = M\Phi \exp[-\frac{1}{2}\Lambda t]$ . To compute the eigenvectors of  $\Sigma$  we first construct the matrix,  $CC^T = M\Phi \exp[-\Lambda t] \Phi^T M^T = Mh_t M^T$ , i.e.  $CC^T$  has eigenvalue matrix  $\Lambda_h = \exp[-\Lambda t]$  and un-normalised eigenvector matrix  $U = M\Phi$ . As a result the matrix  $C^T C$  has normalised eigenvalue matrix  $\hat{U} = C^T U \Lambda_h^{-\frac{1}{2}}$  and eigenvalue matrix  $\Lambda_h$ . To see this note that

$$(C^T U \Lambda_h^{-\frac{1}{2}}) \Lambda_h (C^T U \Lambda_h^{-\frac{1}{2}})^T = C^T U U^T C = CC^T.$$

Hence  $C^T C$  has eigenvector matrix  $\Lambda_h = \exp[-\frac{1}{2}\Lambda t]$  and normalised eigenvector matrix

$$\hat{U} = \left( M\Phi \exp[-\frac{1}{2}\Lambda t] \right)^T M\Phi \left( \exp[-\Lambda t] \right)^{-\frac{1}{2}} = \exp[-\frac{1}{2}\Lambda t] \Phi^T M^T M \Phi \exp[\frac{1}{2}\Lambda t].$$

Finally, it is interesting to note that the projection of the centred co-ordinates onto the eigen-vectors of the covariance matrix is  $Y_P = \hat{U}^T Y_C = \exp[-\frac{1}{2}\Lambda t] \Phi^T M^T = Y_C$ . To characterise the distribution of mapped points, we use the vector  $B_h = \exp[-\frac{1}{2}\Lambda t]e$ , which has the eigen-values of the kernel mapping covariance matrix as elements. In our experiments, we will compare the result of using this representation of the data with the use of the vector of Laplacian eigenvalues  $B_L = \Lambda e$ .

### 3 Sectional Curvature

An interesting property of the embedding is the difference between the geodesic and Euclidean distances. This difference is related to the sectional curvature of the path connecting nodes on a manifold that results from the kernel mapping. In this section, we explore this relationship. We commence by showing how geodesic distance can be computed from the spectrum of the kernel, and then show how this can be used to compute sectional curvature.

#### 3.1 Geodesic Distance

The heat kernel can also be used to compute the path length distribution on the graph. To show this, consider the normalised adjacency matrix  $P = D^{-\frac{1}{2}}(I - \hat{L})D^{-\frac{1}{2}}$ , where  $I$  is the identity matrix. The heat kernel can be rewritten as  $h_t = e^{-t(I-P)}$ . We can perform a McLaurin expansion on the heat-kernel to re-express it as a polynomial in  $t$ . The result of this expansion is

$$h_t = e^{-t} \left( I + tP + \frac{(tP)^2}{2!} + \frac{(tP)^3}{3!} + \dots \right) = e^{-t} \sum_{k=0}^{\infty} P^k \frac{t^k}{k!} \quad (5)$$

The matrix  $P$  has elements

$$P(u, v) = \begin{cases} 1 & \text{if } u = v \\ \frac{1}{\sqrt{\deg(u)\deg(v)}} & \text{if } u \neq v \text{ and } (u, v) \in E \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

As a result, we have that

$$P^k(u, v) = \sum_{S_k} \prod_{i=1}^k \frac{1}{\sqrt{\deg(u_i)\deg(u_{i+1})}} \quad (7)$$

where the walk  $S_k$  is a sequence of vertices  $u_0, \dots, u_k$  of length  $k$  such that  $u_i = u_{i+1}$  or  $(u_i, u_{i+1}) \in E$ . Hence,  $P^k(u, v)$  is the sum of weights of all walks of length  $k$  joining nodes  $u$  and  $v$ . In terms of this quantity, the elements of the heat kernel are given by

$$h_t(u, v) = \exp[-t] \sum_{k=0}^{|V|^2} P^k(u, v) \frac{t^k}{k!} \quad (8)$$

We can find a spectral expression for the matrix  $P^k$  using the eigen-decomposition of the normalised Laplacian. Writing  $P^k = (I - \hat{L})^k = \Phi(I - \Lambda)^k \Phi^T$ , the element associated with the nodes  $u$  and  $v$  is

$$P^k(u, v) = \sum_{i=1}^{|V|} (1 - \lambda_i)^k \phi_i(u) \phi_i(v) \quad (9)$$

The geodesic distance between nodes, i.e. the length of the walk on the graph with the smallest number of connecting edges, can be found by searching for the smallest value of  $k$  for which  $P^k(u, v)$  is non zero, i.e.  $d_G(u, v) = \text{floor}_k P^k(u, v)$ .

### 3.2 Sectional Curvature

An interesting way to characterise the non-linearity of the kernel mapping is to measure the difference between geodesic and Euclidean distances. As we will show, in this section this quantity is related to the sectional curvature associated with paths between nodes on the manifold that results from the kernel mapping. The sectional curvature is determined by the degree to which the geodesic bends away from the Euclidean chord. Hence for a geodesic on the manifold, the sectional curvature can be estimated easily if the Euclidean and geodesic distances are known. Suppose that the geodesic can be locally approximated by a circle. Let the geodesic distance between the pair of points  $u$  and  $v$  be  $d_G(u, v)$  and the corresponding Euclidean distance be  $d_E(u, v)$ . Further let the radius of curvature of the approximating circle be  $r_s(u, v)$  and suppose that the tangent-vector to the manifold undergoes a change in direction of  $2\theta_{u,v}$  as we move along a connecting circle between the two points. In terms of the angle  $\theta_{u,v}$ , the geodesic distance, i.e. the distance traversed along the circular arc, is  $d_G(u, v) = 2r_s(u, v)\theta_{u,v}$ , and as a result we have that  $\theta_{u,v} = d_G(u, v)/2r_s(u, v)$ . The Euclidean distance, on the other hand, is given by  $d_E(u, v) = 2r_s(u, v) \sin \theta_{u,v}$ , and can be approximated using the McClaurin series

$$d_E(u, v) = 2r_s(u, v) \left( \theta_{u,v} - \frac{1}{6} \theta_{u,v}^3 + \dots \right)$$

Substituting for  $\theta_{u,v}$  obtained from the geodesic distance, we have

$$d_E(u,v) = d_g(u,v) - \frac{d_g(u,v)^3}{24r_s^2(u,v)}$$

Solving the above equation for the radius of curvature, the sectional curvature of the geodesic connecting the nodes  $u$  and  $v$  is approximately

$$k_s(u,v) = \frac{1}{r_s(u,v)} = \frac{2\sqrt{6}(d_G(u,v) - d_E(u,v))^{\frac{1}{2}}}{d_G(u,v)^{\frac{3}{2}}} \quad (10)$$

To characterise the geometry of the graph embedding we construct a histogram of sectional curvatures. The sectional curvatures are assigned to  $m$  bins and the normalised contents of the  $j$ th bin is denoted by  $H(j)$ . The feature vector for the graph is constructed from the normalised bin-contents and  $B_C = (H(1), H(2), \dots, H(m))^T$ .

## 4 Experiments

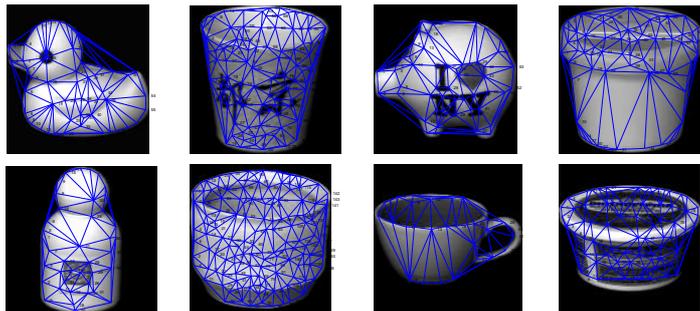


Figure 1: Eight objects with their Delaunay graphs overlaid.

We have applied our geometric technique to images from the COIL data-base. The data-base contains views of 3D objects under controlled viewer and lighting conditions. For each object in the data-base there are 72 equally spaced views, which are obtained as the camera circumscribes the object. We study the images from eight example objects. A sample view of each object is shown in Figure 1. For each image of each object we extract feature points using the method of [7]. We have extracted graphs from the images by computing the Voronoi tessellations of the feature-points, and constructing the region adjacency graph, i.e. the Delaunay triangulation, of the Voronoi regions. Our technique has been applied to the resulting graph structures.

In Figure 2 we show the result of performing PCA on the vector of heat kernel eigenvalues  $B_C$ . Here we have projected the vectors onto the three leading principal components directions. From left-to-right and top-to-bottom, the different subplots show the results as the parameter  $t$  varies from 0.03 to 3000. The different colours in the plot distinguish the different objects. The object clusters evolve in an interesting way as  $t$  increases.

Initially, they are very compact but elongated, and distinct. At large values of  $t$  they are more dispersed and overlapped.

For comparison Figure 3 shows the result if the PCA procedure is repeated on the vectors of Laplacian eigenvalues  $B_L$ . Here the clusters are more dispersed and overlapped than when the small  $t$  heat kernel mapping is used, and are more similar to the result obtained with the large  $t$  mapping.

In Figures 4 and 5 we show the results of using sectional curvature. In Figure 4 we show example histograms for the twenty views for four of the objects in the COIL database. Here the histograms are stacked behind each other, and are ordered by increasing view number. There are a number of conclusions that can be drawn from the histograms. First, the histograms for the same object are relatively stable with view number. Second, the histograms for the different objects have different shapes. The plots shown were obtained with  $t = 0.003$ .

The results of applying PCA to the vectors of histogram bin contents are shown in Figure 5 for different values of  $t$ . We obtain reasonably well defined clusters of objects. In purely qualitative terms they appear poorer than those obtained using the kernel mapping, but better than those obtained using the Laplacian eigenvalues.

To investigate the behavior of the methods in a more quantitative way, we have plotted the Rand index for the different objects. The Rand index is defined as  $R_I = \frac{A}{A+E}$  where  $A$  is the number of "agreements" and  $E$  is the number of "disagreements" in cluster assignment. The index is hence the fraction of views of a particular class that are closer to an object of the same class than to one of another class. The results of the comparison are shown in Figure 6. The blue curve in the plot shows the Rand index as a function of  $t$  for heat kernel mapping. For low values of  $t$  the performance is around 90%, but this drops rapidly once a critical value of  $t$  is exceeded. This shoulder of the curve corresponds falls between the second and third panel in Figure 2. This appears to be associated with the point at which the clusters cease to be compact but elongated and become more extended. The red curve shows the result of the sectional curvature histograms are used. Here the method gives slightly poorer performance at low values of  $t$  (around 0.87), but the performance drop-off is at a larger value of  $t$ . Finally, when the spectrum of normalised Laplacian eigenvalues is used then the Rand-index is 0.57, which is considerably lower than the result obtained with the heat-kernel embedding.

## 5 Conclusion and Future Work

In this paper we have explored how the use of the heat kernel can lead to a useful characterisation of the structure of a graph. Our method is based on performing kernel PCA on the heat kernel. This allows graph nodes to be mapped to points in a high dimensional space. We use the eigenvalues of the covariance matrix of the mapped points to characterise the structure of the graphs. We demonstrate that the methods leads to well formed graph clusters.

There are a number of ways in which we intend to extend this work. First, we will explore the use of the method to construct a generative model that can be used to account for the structure of graphs. Second, we will use the framework to develop a discriminative model that can be used to accurately classify graphs.

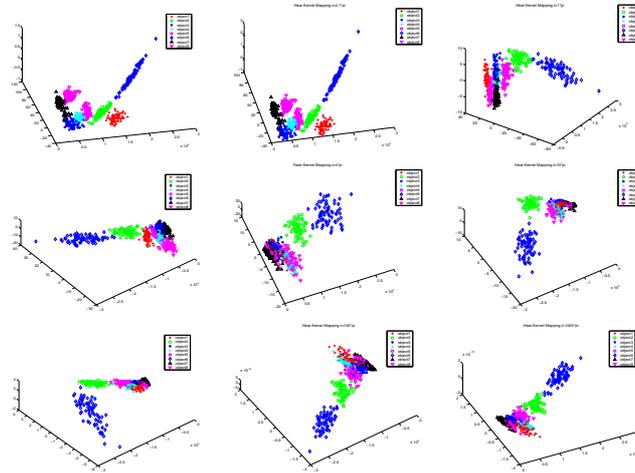


Figure 2: Distributions of graphs obtained by changing the  $t$  variable.

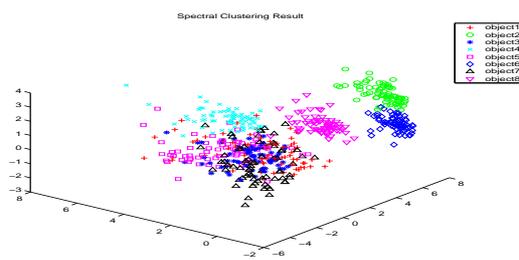


Figure 3: Result of applying PCA to the leading Laplacian eigenvalues.

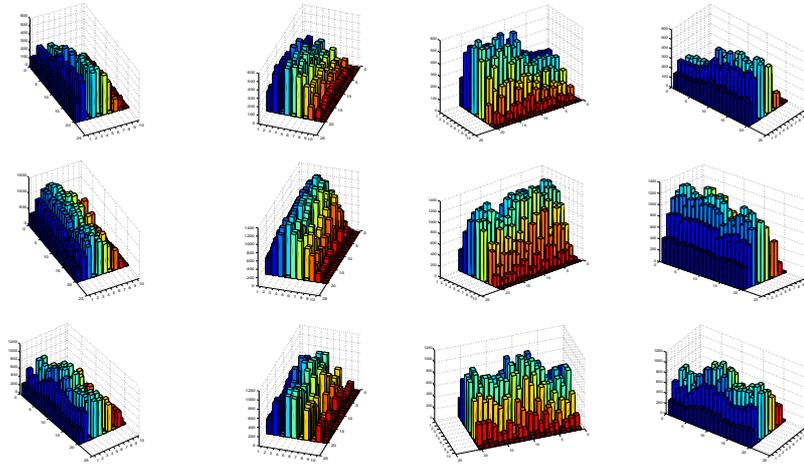


Figure 4: 3-D views of the histograms of the sectional curvature.

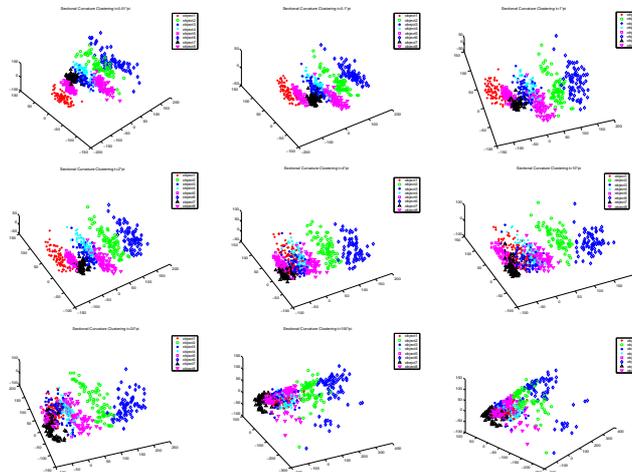


Figure 5: Distributions of graphs obtained from sectional curvature by varying the  $t$  parameter

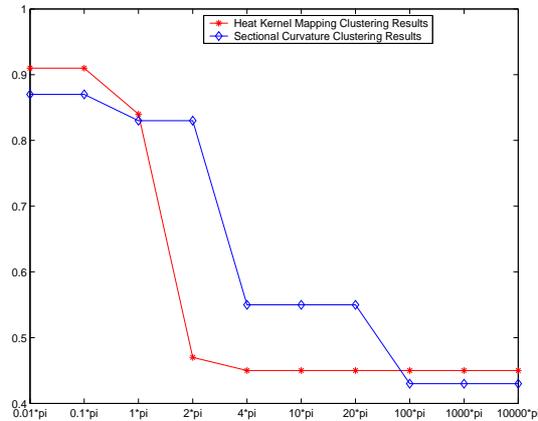


Figure 6: Rand index for the different clustering methods

## References

- [1] A.D.Alexandrov and V.A.Zalgaller. Intrinsic geometry of surfaces. *Transl.Math.Monographs*, 15, 1967.
- [2] A.Ranicki. Algebraic l-theory and topological manifolds. *Cambridge University Press*, 1992.
- [3] R. C. Wilson B. Luo and E.R. Hancock. Spectral embedding of graphs. *Pattern Recognition*, 36:2213–2223, 2003.
- [4] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Neural Information Processing Systems*, 14:634–640, 2002.
- [5] Colin de Verdiere. Spectra of graphs. *Math of France*, 4, 1998.
- [6] F.R.K.Chung. Spectral graph theory. *American Mathematical Society*, 1997.
- [7] C.G. Harris and M.J. Stephens. A combined corner and edge detector. *Fourth Alvey Vision Conference*, pages 147–151, 1994.
- [8] J.B.Tenenbaum, V.D.Silva, and J.C.Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:586–591, 2000.
- [9] J. Lafferty and G. Lebanon. Diffusion kernels on statistical manifolds. *Technical Report CMU-CS-04-101*, 2004.
- [10] A. Shokoufandeh, S. Dickinson, K. Siddiqi, and S. Zucker. Indexing using a spectral encoding of topological structure. *CVPR*, pages 491–497, 1999.
- [11] S.T.Roweis and L.K.Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.