

# Spectral Feature Vectors for Graph Clustering

Bin Luo<sup>1,2</sup>, Richard C. Wilson<sup>1</sup>, and Edwin R. Hancock<sup>1</sup>

<sup>1</sup> Department of Computer Science, University of York  
York YO1 5DD, UK

<sup>2</sup> Anhui University, P.R. China  
{luo,wilson,erh}@cs.york.ac.uk

**Abstract.** This paper investigates whether vectors of graph-spectral features can be used for the purposes of graph-clustering. We commence from the eigenvalues and eigenvectors of the adjacency matrix. Each of the leading eigenmodes represents a cluster of nodes and is mapped to a component of a feature vector. The spectral features used as components of the vectors are the eigenvalues, the cluster volume, the cluster perimeter, the cluster Cheeger constant, the inter-cluster edge distance, and the shared perimeter length. We explore whether these vectors can be used for the purposes of graph-clustering. Here we investigate the use of both central and pairwise clustering methods. On a data-base of view-graphs, the vectors of eigenvalues and shared perimeter lengths provide the best clusters.

## 1 Introduction

Graph clustering is an important yet relatively under-researched topic in machine learning [9,10,4]. The importance of the topic stems from the fact that it is a key tool for learning the class-structure of data abstracted in terms of relational graphs. Problems of this sort are posed by a multitude of unsupervised learning tasks in knowledge engineering, pattern recognition and computer vision. The process can be used to structure large data-bases of relational models [11] or to learn equivalence classes. One of the reasons for limited progress in the area has been the lack of algorithms suitable for clustering relational structures. In particular, the problem has proved elusive to conventional central clustering techniques. The reason for this is that it has proved difficult to define what is meant by the mean or representative graph for each cluster. However, Munger, Bunke and Jiang [1] have recently taken some important steps in this direction by developing a genetic algorithm for searching for median graphs.

Generally speaking, there are two different approaches to graph clustering. The first of these is pairwise clustering[7]. This requires only that a set of pairwise distances between graphs be supplied. The clusters are located by identifying sets of graphs that have strong mutual pairwise affinities. There is therefore no need to explicitly identify an representative (mean, mode or median) graph for each cluster. Unfortunately, the literature on pairwise clustering is much less developed than that on central clustering. The second approach is to embed

graphs in a pattern space[8]. Although the pattern spaces generated in this way are well organised, there are two obstacles to the practical implementation of the method. Firstly, it is difficult to deal with graphs with different numbers of nodes. Secondly, the node and edge correspondences must be known so that the nodes and edges can be mapped in a consistent way to a vector of fixed length.

In this paper, we attempt to overcome these two problems by using graph spectral methods to extract feature vectors from symbolic graphs [2]. Spectral graph theory is a branch of mathematics that aims to characterise the structural properties of graphs using the eigenvalues and eigenvectors of the adjacency matrix, or of the closely related Laplacian matrix. There are a number of well-known results. For instance, the degree of bijectivity of a graph is gauged by the difference between the first and second eigenvalues (this property has been widely exploited in the computer vision literature to develop grouping and segmentation algorithms). In routing theory, on the other hand, considerable use is made of the fact that the leading eigenvector of the adjacency matrix gives the steady-state random walk on the graph. Here we adopt a different approach. Our aim is to use the leading eigenvectors of the adjacency matrix to define clusters of nodes. From the clusters, we extract structural features and using the eigenvalue order to index the components we construct feature-vectors. The length of the vectors are determined by the number of leading eigenvalues. The graph spectral features explored include the eigenvalue spectrum, cluster volume, cluster perimeter, cluster Cheeger constant, shared perimeter and cluster distances. The specific technical goals in this paper are two-fold. First, we aim to investigate whether the independent or principal components of the spectral feature vectors can be used to embed graphs in a pattern space suitable for clustering. Second, we investigate which of the spectral features results in the best clusters.

## 2 Graph Spectra

In this paper we are concerned with the set of graphs  $G_1, G_2, \dots, G_k, \dots, G_N$ . The  $k$ th graph is denoted by  $G_k = (V_k, E_k)$ , where  $V_k$  is the set of nodes and  $E_k \subseteq V_k \times V_k$  is the edge-set. Our approach in this paper is a graph-spectral one. For each graph  $G_k$  we compute the adjacency matrix  $A_k$ . This is a  $|V_k| \times |V_k|$  matrix whose element with row index  $i$  and column index  $j$  is

$$A_k(i, j) = \begin{cases} 1 & \text{if } (i, j) \in E_k \\ 0 & \text{otherwise} \end{cases}. \quad (1)$$

From the adjacency matrices  $A_k, k = 1 \dots N$ , we can calculate the eigenvalues  $\lambda_k$  by solving the equation  $|A_k - \lambda_k I| = 0$  and the associated eigenvectors  $\phi_k^\omega$  by solving the system of equations  $A_k \phi_k^\omega = \lambda_k^\omega \phi_k^\omega$ . We order the eigenvectors according to the decreasing magnitude of the eigenvalues, i.e.  $|\lambda_k^1| > |\lambda_k^2| > \dots > |\lambda_k^{|V_k|}|$ . The eigenvectors are stacked in order to construct the modal matrix  $\Phi_k = (\phi_k^1 | \phi_k^2 | \dots | \phi_k^{|V_k|})$ .

We use only the first  $n$  eigenmodes of the modal matrix to define spectral clusters for each graph. The components of the eigenvectors are used to compute the probabilities that nodes belong to clusters. The probability that the node indexed  $i \in V_k$  in graph  $k$  belongs to the cluster with eigenvalue order  $\omega$  is

$$s_{i,\omega}^k = \frac{|\Phi_k(i, \omega)|}{\sum_{\omega'=1}^n |\Phi_k(i, \omega')|}. \quad (2)$$

### 3 Spectral Features

Our aim is to use spectral features for the modal clusters of the graphs under study to construct feature-vectors. To overcome the correspondence problem, we use the order of the eigenvalues to establish the order of the components of the feature-vectors. We study a number of features suggested by spectral graph theory.

#### 3.1 Unary Features

We commence by considering unary features for the arrangement of modal clusters. The features studied are listed below:

**Leading Eigenvalues:** Our first vector of spectral features is constructed from the ordered eigenvalues of the adjacency matrix. For the graph indexed  $k$ , the vector is  $B_k = (\lambda_k^1, \lambda_k^2, \dots, \lambda_k^n)^T$ .

**Cluster Volume:** The volume  $Vol(S)$  of a subgraph  $S$  of a graph  $G$  is defined to be the sum of the degrees of the nodes belonging to the subgraph, i.e.  $Vol(S) = \sum_{i \in S} deg(i)$ , where  $deg(i)$  is the degree of node  $i$ . By analogy, for the modal clusters, we define the volume of the cluster indexed  $\omega$  in the graph-indexed  $k$  to be

$$Vol_k^\omega = \frac{\sum_{i \in V_k} s_{i\omega}^k deg(i)}{\sum_{\omega=1}^n \sum_{i \in V_k} s_{i\omega}^k deg(i)}. \quad (3)$$

The feature-vector for the graph-indexed  $k$  is  $B_k = (Vol_k^1, Vol_k^2, \dots, Vol_k^n)^T$ .

**Cluster Perimeter:** For a subgraph  $S$  the set of perimeter nodes is  $\Delta(S) = \{(u, v) | (u, v) \in E \wedge u \in S \wedge v \notin S\}$ . The perimeter length of the subgraph is defined to be the number of edges in the perimeter set, i.e.  $\Gamma(S) = |\Delta(S)|$ . Again, by analogy, the perimeter length of the modal cluster indexed  $\omega$  is

$$\Gamma_k^\omega = \frac{\sum_{i \in V_k} \sum_{j \in V_k} s_{i\omega}^k (1 - s_{j\omega}^k) A_k(i, j)}{\sum_{\omega=1}^n \sum_{i \in V_k} \sum_{j \in V_k} s_{i\omega}^k (1 - s_{j\omega}^k) A_k(i, j)}. \quad (4)$$

The perimeter values are ordered according to the modal index of the relevant cluster to form the graph feature vector  $B_k = (\Gamma_k^1, \Gamma_k^2, \dots, \Gamma_k^n)^T$ .

**Cheeger Constant:** The Cheeger constant for the subgraph  $S$  is defined as follows. Suppose that  $\hat{S} = V - S$  is the complement of the subgraph  $S$ . Further let  $E(S, \hat{S}) = \{(u, v) | u \in S \wedge v \in \hat{S}\}$  be the set of edges that connect  $S$  to  $\hat{S}$ . The Cheeger constant for the subgraph  $S$  is

$$H(S) = \frac{|E(S, \hat{S})|}{\min[\text{vol}(S), \text{vol}(\hat{S})]}. \tag{5}$$

The cluster analogue of the Cheeger constant is

$$H_k^\omega = \frac{\Gamma_k^\omega}{\min[\text{Vol}_k^\omega, \text{Vol}_k^{\hat{\omega}}]}, \tag{6}$$

where

$$\text{Vol}_k^{\hat{\omega}} = \sum_{\omega=1}^n \sum_{i \in V_k} s_{i,\omega}^k \text{deg}(i) - \text{Vol}_k^\omega. \tag{7}$$

is the volume of the complement of the cluster indexed  $\omega$ . Again, the cluster Cheeger numbers are ordered to form a spectral feature-vector  $B_k = (H_k^1, H_k^2, \dots, H_k^n)^T$ .

### 3.2 Binary Features

In addition to the unary cluster features, we have studied pairwise cluster attributes.

**Shared Perimeter:** The first pairwise cluster attribute studied is the shared perimeter of each pair of clusters. For the pair subgraphs  $S$  and  $T$  the perimeter is the set of nodes belong to the set  $P(S, T) = \{(u, v) | u \in S \wedge v \in T\}$ . Hence, our cluster-based measure of shared perimeter for the clusters is

$$U_k(u, v) = \frac{\sum_{(i,j) \in E_k} s_{i,u}^k s_{j,v}^k A_k(i, j)}{\sum_{(i',j') \in E_k} s_{i',u}^k s_{j',v}^k}. \tag{8}$$

Each graph is represented by a shared perimeter matrix  $U_k$ . We convert these matrices into long vectors. This is obtained by stacking the columns of the matrix  $U_k$  in eigenvalue order. The resulting vector is  $B_k = (U_k(1, 1), U_k(1, 2), \dots, U_k(1, n), U_k(2, 1), \dots, U_k(2, n), \dots, U_k(n, n))^T$  Each entry in the long-vector corresponds to a different pair of spectral clusters.

**Cluster Distances:** The between cluster distance is defined as the path length, i.e. the minimum number of edges, between the most significant nodes in a pair of clusters. The most significant node in a cluster is the one having the largest co-efficient in the eigenvector associated with the cluster. For the cluster indexed  $u$  in the graph indexed  $k$ , the most significant node is  $i_u^k = \arg \max_i s_{iu}^k$ . To compute the distance, we note that if we multiply the adjacency matrix  $A_k$  by

itself  $l$  times, then the matrix  $(A_k)^l$  represents the distribution of paths of length  $l$  in the graph  $G_k$ . In particular, the element  $(A_k)^l(i, j)$  is the number of paths of length  $l$  edges between the nodes  $i$  and  $j$ . Hence the minimum distance between the most significant nodes of the clusters  $u$  and  $v$  is  $d_{u,v} = \arg \min_l (A_k)^l(i_u^k, i_v^k)$ .

If we only use the first  $n$  leading eigenvectors to describe the graphs, the between cluster distances for each graph can be written as a  $n$  by  $n$  matrix which can be converted to a  $n \times n$  long-vector  $B_k = (d_{1,1}, d_{1,2}, \dots, d_{1,n}, d_{2,1}, \dots, d_{n,n})^T$ .

## 4 Embedding the Spectral Vectors in a Pattern Space

In this section we describe two methods for embedding graphs in eigenspaces. The first of these involves performing principal components analysis on the covariance matrices for the spectral pattern-vectors. The second method involves performing multidimensional scaling on a set of pairwise distance between vectors.

### 4.1 Eigendecomposition of the Graph Representation Matrices

Our first method makes use principal components analysis and follows the parametric eigenspace idea of Murase and Nayar [8]. The relational data for each graph is vectorised in the way outlined in Section 3. The  $N$  different graph vectors are arranged in view order as the columns of the matrix  $S = [B_1 | B_2 | \dots | B_k | \dots | B_N]$ .

Next, we compute the covariance matrix for the elements in the different rows of the matrix  $S$ . This is found by taking the matrix product  $C = SS^T$ . We extract the principal components directions for the relational data by performing an eigendecomposition on the covariance matrix  $C$ . The eigenvalues  $\lambda_i$  are found by solving the eigenvalue equation  $|C - \lambda I| = 0$  and the corresponding eigenvectors  $e_i$  are found by solving the eigenvector equation  $Ce_i = \lambda_i e_i$ .

We use the first 3 leading eigenvectors to represent the graphs extracted from the images. The co-ordinate system of the eigenspace is spanned by the three orthogonal vectors by  $E = (e_1, e_2, e_3)$ . The individual graphs represented by the long vectors  $B_k, k = 1, 2, \dots, N$  can be projected onto this eigenspace using the formula  $x_k = e^T B_k$ . Hence each graph  $G_k$  is represented by a 3-component vector  $x_k$  in the eigenspace.

### 4.2 Multidimensional Scaling

Multidimensional scaling(MDS)[3] is a procedure which allows data specified in terms of a matrix of pairwise distances to be embedded in a Euclidean space. The classical multidimensional scaling method was proposed by Torgenson[12] and Gower[6]. Shepard and Kruskal developed a different scaling technique called ordinal scaling[5]. Here we intend to use the method to embed the graphs extracted from different viewpoints in a low-dimensional space.

To commence we require pairwise distances between graphs. We do this by computing the L2 norms between the spectral pattern vectors for the graphs. For the graphs indexed  $i1$  and  $i2$ , the distance is

$$d_{i1,i2} = \sum_{\alpha=1}^K \left[ B_{i1}(\alpha) - B_{i2}(\alpha) \right]^2. \quad (9)$$

The pairwise similarities  $d_{i1,i2}$  are used as the elements of an  $N \times N$  dissimilarity matrix  $D$ , whose elements are defined as follows

$$D_{i1,i2} = \begin{cases} d_{i1,i2} & \text{if } i1 \neq i2 \\ 0 & \text{if } i1 = i2 \end{cases}. \quad (10)$$

In this paper, we use the classical multidimensional scaling method to embed our the view-graphs in a Euclidean space using the matrix of pairwise dissimilarities  $D$ . The first step of MDS is to calculate a matrix  $T$  whose element with row  $r$  and column  $c$  is given by  $T_{rc} = -\frac{1}{2}[d_{rc}^2 - \hat{d}_r^2 - \hat{d}_c^2 + \hat{d}_\cdot^2]$ , where  $\hat{d}_r = \frac{1}{N} \sum_{c=1}^N d_{rc}$  is the average dissimilarity value over the  $r$ th row,  $\hat{d}_c$  is the similarly defined average value over the  $c$ th column and  $\hat{d}_\cdot = \frac{1}{N^2} \sum_{r=1}^N \sum_{c=1}^N d_{r,c}$  is the average similarity value over all rows and columns of the similarity matrix  $T$ .

We subject the matrix  $T$  to an eigenvector analysis to obtain a matrix of embedding co-ordinates  $X$ . If the rank of  $T$  is  $k, k \leq N$ , then we will have  $k$  non-zero eigenvalues. We arrange these  $k$  non-zero eigenvalues in descending order, i.e.  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k > 0$ . The corresponding ordered eigenvectors are denoted by  $\mathbf{e}_i$  where  $\lambda_i$  is the  $i$ th eigenvalue. The embedding co-ordinate system for the graphs obtained from different views is  $X = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_k]$ , where  $\mathbf{f}_i = \sqrt{\lambda_i} \mathbf{e}_i$  are the scaled eigenvectors. For the graph indexed  $i$ , the embedded vector of co-ordinates is  $\mathbf{x}_i = (X_{i,1}, X_{i,2}, X_{i,3})^T$ .

## 5 Experiments

Our experiments have been conducted with 2D image sequences for 3D objects which undergo slowly varying changes in viewer angle. The image sequences for three different model houses are shown in Figure 1. For each object in the view sequence, we extract corner features. From the extracted corner points we construct Delaunay graphs. The sequences of extracted graphs are shown in Figure 2. Hence for each object we have 10 different graphs. In table 1 we list the number of feature points in each of the views. From inspection of the graphs in Figure 2 and the number of feature points in Table 1 it is clear that the different graphs for the same object undergo significant changes in structure as the viewing direction changes. Hence, this data presents a challenging graph clustering problem.

Our aim is to investigate which combination of spectral feature-vector and embedding strategy gives the best set of graph-clusters. In other words, we aim to see which method gives the best definition of clusters for the different objects.

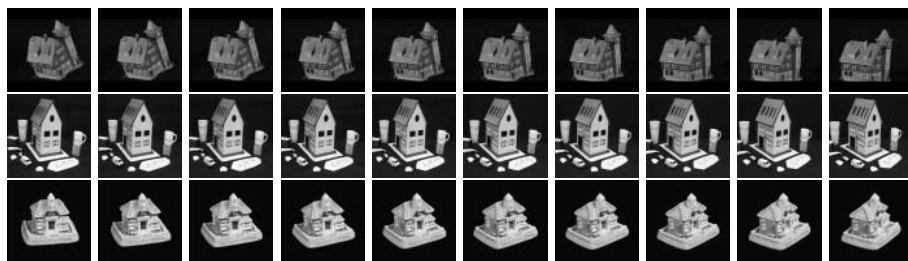
**Table 1.** Number of feature points extracted from the three image sequences

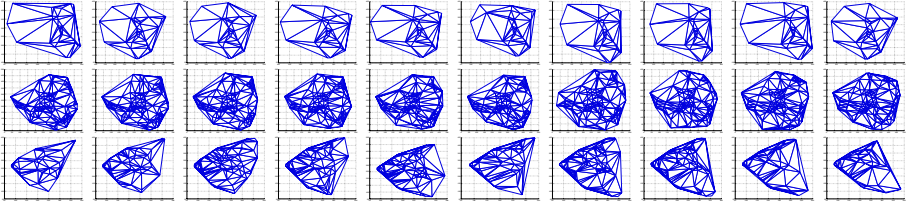
Image Number	1	2	3	4	5	6	7	8	9	10
CMU	30	32	32	30	30	32	30	30	30	31
MOVI	140	134	130	136	137	131	139	141	133	136
Chalet	40	57	92	78	90	64	113	100	67	59

In Figure 3 we compare the results obtained with the different spectral feature vectors. In the centre column of the figure, we show the matrix of pairwise Euclidean distances between the feature-vectors for the different graphs (this is best viewed in colour). The matrix has 30 rows and columns (i.e. one for each of the images in the three sequences with the three sequences concatenated), and the images are ordered according to the position in the sequence. From top-to-bottom, the different rows show the results obtained when the feature-vectors are constructed using the eigenvalues of the adjacency matrix, the cluster volumes, the cluster perimeters, the cluster Cheeger constants, the shared perimeter length and the inter-cluster edge distance. From the pattern of pairwise distances, it is clear that the eigenvalues and the shared perimeter length give the best block structure in the matrix. Hence these two attributes may be expected to result in the best clusters.

To test this assertion, in the left-most and right-most columns of Figure 3 we show the leading eigenvectors of the embedding spaces for the spectral feature-vectors. The left-hand column shows the results obtained with principal components analysis. The right-hand column shows the results obtained with multidimensional scaling. From the plots, it is clear that the best clusters are obtained when MDS is applied to the vectors of eigenvalues and shared perimeter length. Principal components analysis, on the other hand, does not give a space in which there is a clear cluster-structure.

We now embark on a more quantitative analysis of the different spectral representations. To do this we plot the normalised squared eigenvalues  $\hat{\lambda}_i^2 =$

**Fig. 1.** Image sequences



**Fig. 2.** Graph representation of the sequences

$\frac{\lambda_i^2}{\sum_{i=1}^n \lambda_i^2}$  against the eigenvalue magnitude order  $i$ . In the case of the parametric eigenspace, these represent the fraction of the total data variance residing in the direction of the relevant eigenvector. In the case of multidimensional scaling, the normalised squared eigenvalues represent the variance of the inter-graph distances in the directions of the eigenvectors of the similarity matrix. The first two plots are for the case of the parametric eigenspaces. The left-hand side plot of Figure 4 is for the unary attribute of eigenvalues, while the middle plot is for the pairwise attribute of shared perimeters. The main feature to note is that of the binary features the vector of adjacency matrix eigenvalues has the fastest rate of decay, i.e. the eigenspace has a lower latent dimensionality, while the vector of Cheeger constants has the slowest rate of decay, i.e. the eigenspace has greater dimensionality. In the case of the binary attributes, the shared perimeter results in the eigenspace of lower dimensionality.

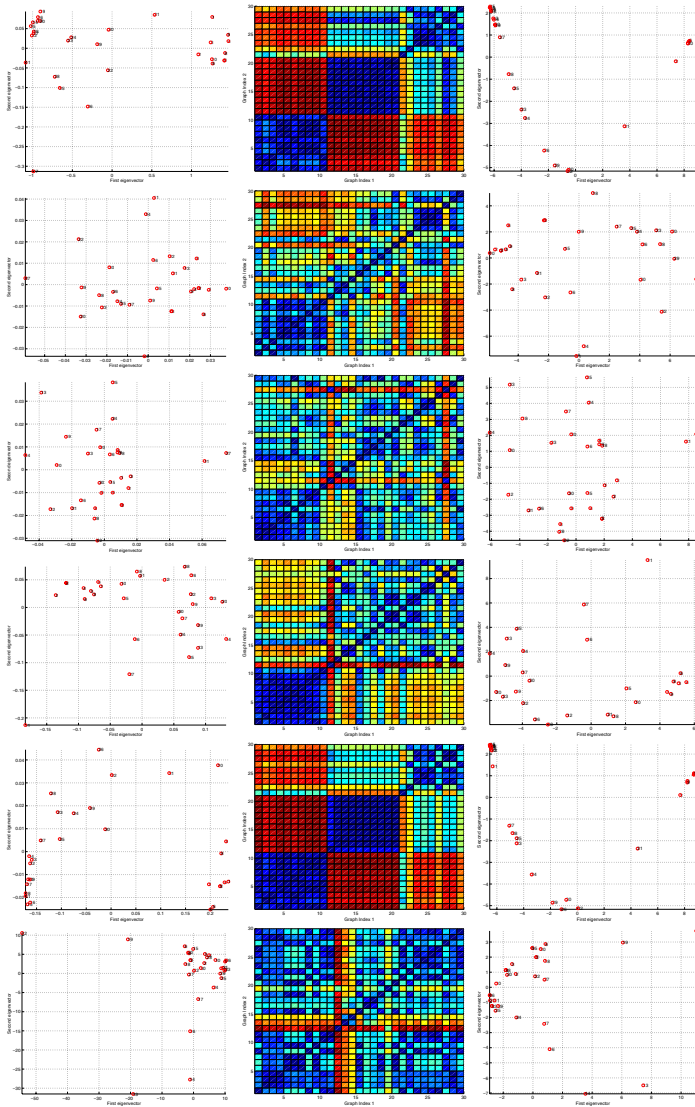
In the last plot of Figure 4 we show the eigenvalues of the graph similarity matrix. We repeat the sequence of plots for the three house data-sets, but merge the curves for the unary and binary attributes into a single plot. Again the vector of adjacency matrix eigenvalues gives the space of lower dimensionality, while the vector of inter-cluster distances gives the space of greatest dimensionality.

Finally, we compare the performances of the graph embedding methods using measures of their classification accuracy. Each of the six graph spectral features mentioned above are used. We have assigned the graphs to classes using the the K-means classifier. The classifier has been applied to the raw Euclidean distances, and to the distances in the reduced dimension feature-spaces obtained using PCA and MDS. In Table 2 we list the number of correctly classified graphs. From the table, it is clear that the eigenvalues and the shared perimeters are the best features since they return higher correct classification rates. Cluster distance is the worst feature for clustering graphs. We note also that classification in the feature-space produced by PCA is better than in the original feature vector spaces. However, the best results come from the MDS embedded class spaces.

## 6 Conclusions

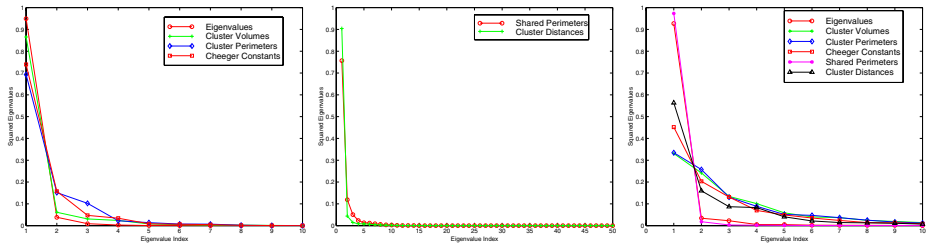
In this paper we have investigated how vectors of graph-spectral attributes can be used for the purposes of clustering graphs. The attributes studied are the





**Fig. 3.** Eigenspace and MDS space embedding using the spectral features of binary adjacency graph spectra, cluster volumes, cluster perimeters, cluster Cheeger constants, shared perimeters and cluster distances

leading eigenvalues, and, the volumes, perimeters, shared perimeters and Cheeger numbers for modal clusters. The best clusters emerge when we apply MDS to the vectors of leading eigenvalues. The best clusters result when we use cluster volume or shared perimeter.



**Fig. 4.** Comparison of graph spectral features for eigenspaces. The left plot is for the unary features in eigenspace, the middle plot is for the binary features in eigenspace and the right plot is for all the spectral features in MDS space

**Table 2.** Correct classifications

Features	Eigenvalues	Volumes	Perimeters	Cheeger constants	Shared Perimeters	Distances
Raw vector	29	26	26	13	25	12
PCA	29	27	26	17	25	12
MDS	29	28	27	16	29	17

Hence, we have shown how to cluster purely symbolic graphs using simple spectral attributes. The graphs studied in our analysis are of different size, and we do not need to locate correspondences. Our future plans involve studying in more detail the structure of the pattern-spaces resulting from our spectral features. Here we intend to investigate the use of ICA as an alternative to PCA as a means of embedding the graphs in a pattern-space. We also intend to study how support vector machines and the EM algorithm can be used to learn the structure of the pattern spaces. Finally, we intend to investigate whether the spectral attributes studied here can be used for the purposes of organising large image data-bases.

## References

1. H. Bunke. Error correcting graph matching: On the influence of the underlying cost function. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:917–922, 1999. 83
2. F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society Ed., CBMS series 92, 1997. 84
3. Chatfield C. and Collins A. J. *Introduction to multivariate analysis*. Chapman & Hall, 1980. 87
4. R. Englert and R. Glantz. Towards the clustering of graphs. In *2nd IAPR-TC-15 Workshop on Graph-Based Representation*, 1999. 83
5. Kruskal J. B. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29:115–129, 1964. 87

6. Gower J. C. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53:325–328, 1966. 87
7. B. Luo, A. Robles-Kelly, A. Torsello, R. C. Wilson, and E. R. Hancock. Clustering shock trees. In *Proceedings of GbR*, pages 217–228, 2001. 83
8. H. Murase and S. K. Nayar. Illumination planning for object recognition using parametric eigenspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1219–1227, 1994. 84, 87
9. S. Rizzi. Genetic operators for hierarchical graph clustering. *Pattern Recognition Letters*, 19:1293–1300, 1998. 83
10. J. Segen. Learning graph models of shape. In *Proceedings of the Fifth International Conference on Machine Learning*, pages 25–29, 1988. 83
11. K. Sengupta and K. L. Boyer. Organizing large structural modelbases. *PAMI*, 17(4):321–332, April 1995. 83
12. Torgerson W. S. Multidimensional scaling. i. theory and method. *Psychometrika*, 17:401–419, 1952. 87