The Independent and Principal Component of Graph Spectra

Bin Luo[1,2] Richard C. Wilson[1] Edwin R. Hancock [1]
[1]Department of Computer Science, University of York,York YO1 5DD, UK.
[2]Anhui University, P.R. China
Tel. +44-1904-433374 E-mail: erh@cs.york.ac.uk

Abstract

In this paper, we demonstrate how PCA and ICA can be used for embedding graphs in pattern-spaces. Graph spectral feature vectors are calculated from the leading eigenvalues and eigenvectors of the unweighted graph adjacency matrix. The vectors are then embedded in a lower dimensional pattern space using both the PCA and ICA decomposition methods. Synthetic and real sequences are tested using the proposed graph clustering algorithm. The preliminary results show that generally speaking the ICA is better than PCA for clustering graphs. The best choice of graph spectral feature for clustering is the cluster shared perimeters.

1 Introduction

Graph clustering is an important yet relatively underresearched topic in machine learning [7, 8, 2]. 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[9] 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[3]. 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[4]. 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. To overcome these two problems, we have used graph spectral methods to extract feature vectors from symbolic graphs[3]. Graph spectral features are used to generate feature vectors. The length of the vectors are determined by the number of leading eigenvalues, and the order of the components of the vectors is the order of the eigenvectors. The graph spectral features explored include the eigenvalue spectrum, cluster volume, cluster perimeter, cluster Cheeger constant, shared perimeter and cluster distances. The aim 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 $G_1, G_2, .., G_k, ..., G_N$. The kth 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 $A_k, k = 1...N$ at hand, we can calculate the eigenvalues λ_k by solving the equation $|A_k - \lambda_k I| = 0$ and the associated eigenvectors ϕ_k^ω by

1051-4651/02 $17.00 © 2002 IEEE

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. $|\lambda_k^1| > |\lambda_k^2| > \ldots |\lambda_k^{|V_k|}|$. The eigenvectors are stacked in order to construct the modal matrix $\Phi_k = (\phi_k^1|\phi_k^2|\ldots|\phi_k^{|V_k|})$.

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 ω is

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

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 $B_k = (\lambda_k^1, \lambda_k^2, ..., \lambda_k^n)^T$.

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 ω in the graph-indexed k to be

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

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

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 ω is

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

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

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

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

The cluster analogue of the Cheeger constant is

$$H_k^\omega = \frac{\Gamma_k^\omega}{min[Vol_k^\omega, Vol_k^{\bar{\omega}}]}, \tag{6}$$

where

$$Vol_k^{\bar{\omega}} = \sum_{\omega=1}^{n}\sum_{i\in V_k} s_{i,\omega}^k deg(i) - Vol_k^\omega \tag{7}$$

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

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

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

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 $B_k = (U_k(1,1), U_k(1,2), ..., U_k(1,n), U_k(2,1)....., U_k(2,n,),$ $...U_k(n,n))^T$ 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 $B_k = (d_{1,1}, d_{1,2}, ...d_{1,n}, d_{2,1}.....d_{n,n})^T$.

3.3 Pattern space embedding by PCA

Our first method makes use principal components analysis and follows the parametric eigenspace idea of Murase and Nayar [4, 5, 6]. The relational data for each image is vectorised in the way outlined in Section 3. The N different image vectors are arranged in view order as the columns of the matrix $S = [B_1|B_2|\ldots|B_i|\ldots|B_N]$.

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 λ_i are found by solving the eigenvalue equation $|C - \lambda I| = 0$ and the corresponding eigenvalues $\vec{e}_i$ are found by solving the eigenvector equation $C\vec{e}_i = \lambda_i\vec{e}_i$.

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 $E = (\vec{e}_1, \vec{e}_2, \vec{e}_3)$. The individual graphs represented by the long vectors $B_i, i = 1, 2, \ldots, N$ can be projected onto this eigenspace using the formula $\vec{x}_i = \vec{e}^T B_i$ Hence each graph G_i is represented by a 3-component vector $\vec{x}_i$ in the eigenspace.

3.4 Pattern space embedding by ICA

Our second approach uses Independent component analysis(ICA) to embed the graphs in a pattern space. We explore how to decompose a set of graphs into significantly different independent components. These can then be used for graph clustering by projecting the original graphs into the pattern space spanned by the independent components.

The ICA algorithm used in this paper is Cardoso and Souloumiac's JADE algorithm[3]. JADE is a statistically based algorithm. The main features of the algorithm are as follows. As with other ICA algorithms, the first step is data whitening or sphering. The aim is to eliminate correlations from the data. This can be achieved by removing the mean of the data and using PCA on the data covariance matrix. As a result, the whitened vector set is $Z = WB$, where W is the estimated whitening matrix. The second step of JADE is estimate the 4th-order cumulants Q_z. In the noiseless case, Q_z can be calculated as follows,

$$Q_Z(I_n) = E\{|Z|^2 ZZ^T\} - (n+1)I_n, \qquad (9)$$

where I_n is the n-order identity matrix and $E(.)$ is the expectation operator. Next, a joint diagonalization is performed to find a matrix $\hat{V}$ to minimise the non-diagonal entries of the cumulants matrices,

$$\hat{V} = \arg\min \Sigma_i Off(V^T Q_Z V). \qquad (10)$$

Again we use the first 3 most significant independent components to represent the graphs extracted from the images. The co-ordinate system of the patternspace is spanned by the three independent components by $\hat{e} = (\hat{V}_1, \hat{V}_2, \hat{V}_3)$. The individual graphs represented by the long vectors Z_k, $k = 1, 2, ..., N$ can be projected onto this pattern space using the formula $\vec{x}_k = \vec{e}^T Z_k$. Hence each graph G_k is represented by a 3-component vector $\vec{x}_k$ in the pattern space.

4 Experiments

The aim in this paper is to compare the utility of PCA and ICA for clustering graphs which are represented by vectors of spectral features. To achieve this, we need several groups of graphs which have similar within-group structures and significantly different between-group structures. As shown in Figure 1, sample images are taken from four different sequences of images of houses taken from slowly varying viewpoints. These include a synthetic model sequence, the CMU house image sequence, the INRIA MOV1 house sequence and a Swiss chalet sequence which were captured in out laboratory. For each image, we extract corner features and then generate Delaunay graphs on the corner features. From the Delaunay graphs in Figure 2, we can see that the graphs of the synthetic and the CNW images are significantly different from other groups of graphs and are sirnilar with the ones in the same group. More diverse graph structures appear in the MOV1 graphs and in the Swiss chalet graphs. We use the leading eigenvectors in the case of PCA approach and leading independent components in the ICA approach to define clusters of nodes.

The first two columns of Figure 3 are obtained using PCA analysis and the second two columns are obtained using ICA decomposition. In both cases, the left column shows the resulting pattern space and the second column shows the Euclidean distance maps. The graphs are embedded in the different pattern spaces. From the first row to the fifth row, we show the experimental results obtained when the spectral feature vectors are constructed from ordered eigenvalues, the cluster volumes, the cluster perimeters, the cluster Cheeger constants, the shared perimeters and the cluster distances.

From Figure 3 we observe that although some cluster structures present in the pattern spaces generated by PCA, the clusters are far more clear when ICA is used. Since we know there are four different classes of graphs, we expect that our clustering method should return four distinct clusters. From row 1 and column 4, we see that by using the vector of ordered eigenvalues, the 10 synthetic model graphs are separated from other graphs very well. The cluster of CMU houses is obvious although there exists three subclusters which can be observed from the sub-block structure of the distance map along the diagonal. The clusters of the MOV1 and the chalet graphs are barely distinguishable. The best result comes from the ICA embedding of the shared perimeter vectors. This can be seen from row 5 and column

4 of Figure 3. The first three clusters are fairly clear. Only the chalet sequence is not well clustered. This approves the observation made earlier that the graph structures of the chalet sequence are not stable.

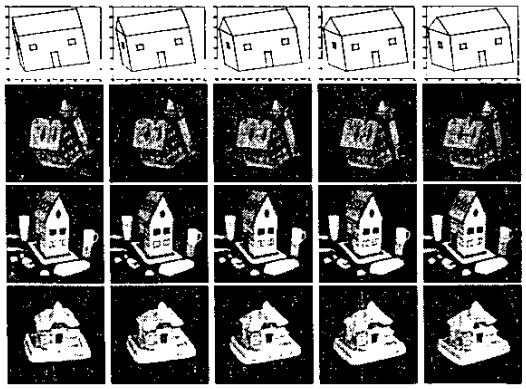

Figure 1. The synthetic, CMU, MOV1 and the chalet sequences

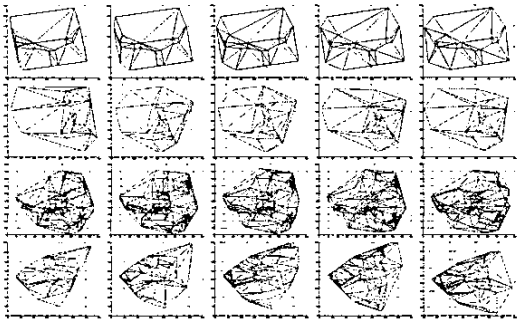

Figure 2. Graph representation of the synthetic, CMU, MOVI and the chalet sequences

5 Conclusions

In this paper, we explored that ICA and PCA can be used for embedding graphs in a pattern space suitable for clustering. Generally speaking, the ICA embedding is better than the PCA embedding in the sense of separating groups of graphs. Of the spectral features used in our experiments, the eigenvalues of the adjacency matrix and the shared perimeters result in the pattern spaces with the best cluster structure.

Our future plans involve studying in more detail the structure of the patten-spaces resulting from our spectral features. We 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.

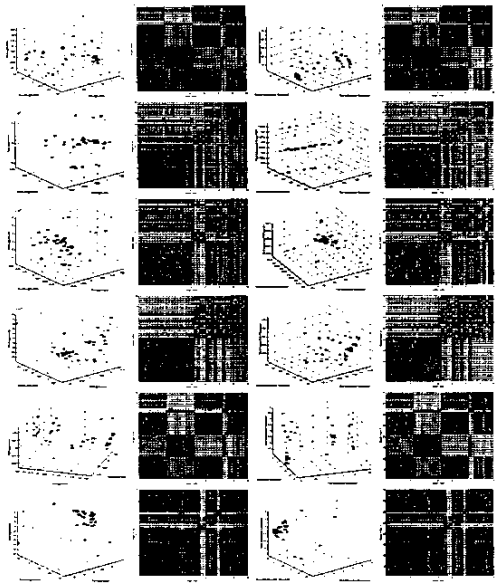

Figure 3. PCA and ICA clustering of the sequences

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.

[2] R. Englert and R. Glantz. Towards the clustering of graphs. In *2nd IAPR-TC-15 Workshop on Graph-Based Representation*, pages 213–221,1999.

[3] B. Luo, A. Robles-Kelly, A. Torsello, R.C. Wilson, and E.R. Hancock. A probabilistic framework for graph clustering. In *Proceedings of CVPR*, pages I:912–919, 2001.

[4] 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.

[5] H. Murase and S.K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.

[6] S.K. Nayar, S.A. Nene, and H. Murase. Subspace methods for robot vision. *IEEE T-RA*, 12(5):750–758, October 1996.

[7] S. Rizzi. Genetic operators for hierarchical graph clustering. *Pattern Recognition Letters*, 19:1293–1300, 1998.

[8] J. Segen. Learning graph models of shape. In *Proceedings of the Fifth International Conference on Machine Learning*, pages 25–29, 1988.

[9] K. Sengupta and K.L. Boyer. Organizing large structural modelbases. *PAMI*, 17(4):321–332, April 1995.