SPECTRAL METHOD FOR LEARNING STRUCTURAL VARIATIONS IN GRAPHS

Bin Luo[1,2]

[1] The Key Laboratory of IC&SP,
Anhui University, Anhui, Hefei 230039, China

Richard Wilson[2], *Edwin Hancock*[2]

[2] University of York, York YO10 5DD, UK

ABSTRACT

This paper investigates the use of graph-spectral methods for learning the modes of structural variation in sets of graphs. Our approach is as follows. First, we vectorise the adjacency matrices of the graphs. Using a graph-matching method we establish correspondences between the components of the vectors. Using the correspondences we cluster the graphs using a Gaussian mixture model. For each cluster we compute the mean and covariance matrix for the vectorised adjacency matrices. We allow the graphs to undergo structural deformation by linearly perturbing the mean adjacency matrix in the direction of the modes of the covariance matrix. We demonstrate the method on sets of corner Delaunay graphs for 3D objects viewed from varying directions.

1. INTRODUCTION

Many shape-analysis problems in computer vision can be abstracted using relational graphs. Examples include the use of shock graphs[7] to represent the differential structure of boundary contours and view graphs. The main advantage of the graph-representation is that it captures the structural variation of shape in a parsimonious way. However, there are a number of limitations to the use of graph-representations. First, they are notoriously susceptible to the effects of noise and clutter. Hence, the addition or loss of a few nodes and edges can result in graphs of significantly different structure. Second, it is difficult to characterise and hence learn the distribution of structural variations in sets of graphs. As a result it is not possible to construct pattern-spaces associated with the modes of structural variations for graphs.

The ability to learn the modes of variation of the adjacency matrix is an important one. The reason for this is that it allows the statistical significance of changes in the edge-structure of graphs to be assessed. This is crucial capability in measuring the similarity of graphs, or in matching them to one-another. There have been several previous attempts to solve this problem. One of the earliest of these was to extend the concept of string edit distance to graphs. Here Fu and his co-workers[5] introduced edit costs associated with the re-labelling, insertion and deletion of edges. However, the costs were selected on an ad-hoc basis and there was no method for learning the edit costs. Moreover, the theory underlying graph-edit distance lacks the formality and rigour of that for strings. However, some steps have recently been taken by Bunke[1], who has shown the relationship between edit distance and the size of the maximum common subgraph. Christmas, Kittler and Petrou[2] have taken a probabilistic approach to the problem. They develop a Gaussian model for edge structure, which is used to compute compatibility function for relaxation labelling. Again, there is no methodology for learning the model from data.

The aim in this paper is to overcome this problem by developing statistical methods for analysing the modes of structural variation in sets of graphs. We pose the problem as that of estimating a covariance matrix for the edge-sets of the graphs. To do this we require a means of vectorising the graphs. We do this using correspondence matches to permute the adjacency matrices onto a standard reference order. We estimate the required correspondences using a recently reported EM algorithm[4], in which the correspondences are located using a singular value decomposition in the maximisation step. The standardised adjacency matrices are vectorised by stacking the columns to form long-vectors. We model the pattern-space for the standardised long-vectors using a Gaussian mixture model. The EM algorithm is used to make maximum likelihood estimates for the mean-vectors and covariance matrices for the mixture components.

From the covariance matrices for the standardised vectors, there are a number of ways in which to construct pattern-spaces. The simplest of these is to construct an eigenspace by projecting the standardised adjacency matrix long-vectors onto the leading eigenvectors of the covariance matrix. The distribution of graphs so produced can be further simplified by fitting a manifold or a mixture model. However, here we use the eigenvectors of the covariance matrix to construct a linear model for variations in the adjacency matrices. To do this we borrow ideas from point distribution models. Here Cootes and Taylor[3] have shown how to construct a linear shape-space for sets of landmark points for 2D shapes. We use a variant of this idea to model variations in the long-vectors for the standardised covariance matrices. We com-

mence by computing the leading eigenvectors for the cluster covariance matrices. The graphs deformed by displacing the mean adjacency matrix long-vectors in the directions of the leading eigenvectors of the covariance matrix. Our method allows the pattern of edge-deformations to be learned and applied at the global level. In principal edge edit costs can be obtained from our model via a process of averaging the deformations. In this way we construct a generative model of graph-structure. This model may be both fitted to data and sampled.

2. BACKGROUND

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

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

To construct our generative model of variations in graph structure, we will convert the adjacency matrices into long-vectors where the entries have a standardised order. To do this we need to permute the order of the rows and columns of the adjacency matrices. We represent the set of correspondences between the nodes in pairs of graphs using a correspondence matrix. For the graphs indexed k and l, the correspondence matrix is denoted by $S_{k,l}$. The elements of the matrix convey the following meaning

$$S_{k,l}(i,j) = \begin{cases} 1 & \text{if } i \in V_k \text{ and } j \in V_l \text{ are in correspondence} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

To recover the correspondence matrices, we use the EM algorithm recently reported by Luo and Hancock [4]. This algorithm commences from a Bernoulli model for the correspondences indicators which are treated as missing data, From this distribution an expected log-likelihood function for the missing correspondence indicators is developed. In the maximisation step a singular value decomposition method is used to recover the correspondence matrix which satisfy the condition

$$S_{k,l}(i,j) = \arg\max_{S} Tr[A_k^T S A_l S^T] \quad (3)$$

In other words, the maximum likelihood correspondence matrices are those that maximise the correlation of the two adjacency matrices.

2.1. Preclustering

To establish initial clusters, we perform pairwise clustering using the pairwise distance between spectral feature-vectors extracted from the graphs. 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 ψ_k^ω by solving the system of equations $A_k \psi_k^\omega = \lambda_k^\omega \psi_k^\omega$. We order the eigenvectors according to the decreasing magnitude of the eigenvalues, i.e. $|\lambda_k^1| > |\lambda_k^2| > \ldots |\lambda_k^{|V_k|}|$. With the eigenvalues and eigenvectors of the adjacency matrix to hand, the spectral decomposition for the adjacency matrix of the graph indexed k is

$$A_k = \sum_{i=1}^{|V_k|} \lambda_k^i \psi_k^i (\psi_k^i)^T \quad (4)$$

Our vector of spectral features is constructed from the ordered eigenvalues of the adjacency matrix. For the graph indexed k, the vector is $B_k = (\lambda_k^1, \lambda_k^2, ..., \lambda_k^n)^T$. For the pair of graphs indexed k and l, the squared Euclidean distance between the spectral feature vectors is $d_{k,l}^2 = (B_k - B_l)^T (B_l - B_l)$. From the set of distance, we can construct a pairwise representation of the affinity of different graphs. The affinity is captured using an $N \times N$ matrix whose element with row k and column l is $W_{k,l}^{(0)} = \exp[-\mu d_{k,l}^2]$.

We apply a pairwise clustering method to the matrix W. The initial set of clusters are defined by the eigenmodes of the link-weight matrix W. Here we follow Sarkar and Boyer [6] who have shown how the positive eigenvectors of the matrix of link-weights can be used to assign objects to perceptual clusters. Using the Rayleigh-Ritz theorem, they observe that the scalar quantity $\mathbf{x}^t W \mathbf{x}$ is maximised when $\mathbf{x}$ is the leading eigenvector of W. Moreover, each of the subdominant eigenvectors corresponds to a disjoint pairwise cluster. They confine their attention to the same-sign positive eigenvectors (i.e. those whose corresponding eigenvalues are real and positive, and whose components are either all positive or are all negative in sign). If a component of a positive same-sign eigenvector is non-zero, then the corresponding object belongs to the associated cluster of segmental entities. The eigenvalues $\lambda_1, \lambda_2 ....$ of W are the solutions of the equation $|W - \lambda I| = 0$ where I is the $|V| \times |V|$ identity matrix. The corresponding eigenvectors $\mathbf{x}_{\lambda_1}, \mathbf{x}_{\lambda_2}, ...$ are found by solving the equation $W \mathbf{x}_{\lambda_i} = \lambda_i \mathbf{x}_{\lambda_i}$. With this notation, the set of positive same-sign eigenvectors is represented by $\Omega = \{\omega | \lambda_\omega > 0 \wedge [(\mathbf{x}_\omega^*(i) > 0 \forall i) \vee \mathbf{x}_\omega^*(i) < 0 \forall i])\}$.

To develop our pairwise clustering method further, we require a cluster membership indicator $s_{i\omega}$ which convey the following meaning

$$s_{i\omega} = \begin{cases} 1 & \text{if node } i \text{ belongs to cluster } \omega \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

We use the iterative maximum likelihood algorithm for pairwise clustering recently developed by Robles-Kelly and Hancock [8]. The resulting estimated weight matrix $\hat{W}$ can be used to calculate the leading eigenvectors and eigenvalues.

We initialise the cluster memberships using the components of the same-sign positive eigenvectors and set

$$s_{iw} = \frac{|\mathbf{x}_\omega^*(i)|}{\sum_{i \in V_\omega} |\mathbf{x}_\omega^*(i)|} \quad (6)$$

3. GRAPH CLUSTERING

We use the pairwise clusters to seed our EM algorithm. For each cluster, we identify the modal graph. This is the graph for which $l_\omega = \arg\max_i s_{i\omega}$. The modal graphs for the different clusters are used to establish a reference order for the nodes. We use the correspondence matrices to permute the node-order of the graphs into the reference order for the different clusters. For the graph indexed k, the permuted adjacency matrix relevant to the cluster ω is $M_k^\omega = S_{k,l_\omega}^T A_k S_{k,l_\omega}$.

Once the adjacency matrices have been permuted, then we can convert them into pattern-vectors. We do this by stacking the columns of the adjacency matrix to form a long-vector. For the graph-indexed k and the cluster ω the long-vector is , $V_k^\omega = (M_k^\omega(1,1), M_i^\omega(1,2), ..., M_k^\omega(n,n))$. Using the pairwise cluster-membership indicators we can compute the mean long-vectors and the covariance matrices for each cluster. These will be used to seed our EM algorithm. For the cluster indexed ω, the mean-vector is

$$U_\omega^{(0)} = \sum_{k=1}^{N} s_{k\omega} V_k^\omega \quad (7)$$

while the covariance matrix is

$$\Sigma_\omega^{(0)} = \sum_{k=1}^{N} s_{k\omega} (V_k^\omega - U_\omega)(V_k^\omega - U_\omega)^T \quad (8)$$

With this initialisation to hand, we use a Gaussian mixture model to iteratively recover improved estimates of the mean long-vector and the associated covariance matrix. We commence by assuming that the long-vectors are drawn from the Gaussian distribution

$$p(V_k^\omega | U_\omega, \Sigma_\omega) = \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{|\Sigma|}} \exp\left[-\frac{1}{2}(V_k^\omega - U_\omega)^T \Sigma^{-1} (V_k^\omega - U_\omega)\right] \quad (9)$$

In the maximisation step of the algorithm, the mean long-vector and the covariance matrices for each of the clusters are updated. The updated mean is

$$U_\omega^{(n+1)} = \sum_{k=1}^{N} P(k \in \omega | V_k^\omega, U_\omega^{(n)}, \Sigma_\omega^{(n)}) V_k^\omega \quad (10)$$

While the updated estimate of the covariance matrix is

$$\Sigma_\omega^{(n+1)} = \sum_{k=1}^{T} P(k \in \omega | U_\omega^{(n)}, \Sigma_\omega^{(n)}, V_k^\omega)(V_k^\omega - U_\omega^{(n)})(V_k^\omega - U_\omega^{(n)})^T \quad (11)$$

In the expectation step the *a posteriori* cluster membership probabilities are updated using the Bayes rule. The update equation is

$$P(k \in \omega | U_\omega^{(n)}, \Sigma_\omega^{(n)}, V_k^\omega) = \frac{\alpha_{k\omega} p(V_k^\omega | U_\omega^{(n+1)}, \Sigma_\omega^{(n)})}{\sum_{\omega \in \Omega} \alpha_{k\omega} p(V_k^\omega | U_\omega^{(n+1)}, \Sigma_\omega^{(n)})} \quad (12)$$

where

$$\alpha_{k\omega} = \frac{1}{N} \sum_{k=1}^{N} P(k \in \omega | U_\omega^{(n)}, \Sigma_\omega^{(n)}, V_k^\omega) \quad (13)$$

4. MODAL ANALYSIS

Our aim is to use the covariance matrix delivered by the EM algorithm to analyse the modes of variation for the long-vectors representing the adjacency matrices of the graphs. To do this we use a simple linaer model which has been used to great effect in represent the modes of variations for sets of point patterns.

We commence by computing the eigenvalues and eigenvectors for the covariance matrix Σ_ω. The eigenvalues $\lambda_1, \lambda_2, ...$ are found by solving the polynomial equation $|\Sigma_\omega - \lambda I| = 0$, where I is the identity matrix. The assoiciated eigenvectors $\phi_1^\omega, \phi_2^\omega, ...$ are found by solving the linear eigenvector equation $\Sigma_\omega \phi_l^\omega = \lambda_l \phi_l^\omega$ From the eigenvectors we construct a modal matrix. The eigenvectors are ordered in decreasing eigenvalue order to form the columns of the modal matrix $\Phi_\omega = (\phi_1^\omega | \phi_2^\omega | .... | \phi_n^\omega)$. The linear deformation model allows the components of the adjacency matrix long-vectors to undergo displacement in the directions of the eigenvectors of the covariance matrix. For the long-vector of the graph G_k and the cluster indexed ω, the displaced vector is given by

$$\tilde{V}_k^\omega = U_\omega + \Phi_\omega b_{k\omega}^* \quad (14)$$

The degree of displacement for the different vector components is controlled by the vector of parameters $b_{k\omega}$.

The linear deformation model may be fitted to data. This is done by searching for the least squares parameter vector. Suppose that the model is to be fitted to the graph with standardised adjacency matrix V_k. The least-squares parameter vector satisfies the condition

$$b_{k\omega}^* = \arg\min_b (V_k^\omega - U_\omega - \Phi_\omega b)^T (V_k^\omega - U_\omega - \Phi_\omega b) \quad (15)$$

The solution is

$$b_{k\omega}^* = \frac{1}{2}\left[\Phi_\omega^T \Phi_\omega\right]^{-1} \Phi_\omega^T \left\{V_k^\omega - U_\omega\right\} \quad (16)$$

5. EXPERIMENT

We present experiments on graphs extracted from 2D images of 3D objects which undergo gradual changes in viewing direction. The graphs are constructed by first detecting corners in the images, and then triangulating them using a Delaunay graph. The images used in our study are taken from a turntable sequence of a model house in the CMU/VASC database.

In figure 1, we show example images from the CMU /VASC sequence, which contains 10 frames in total. In Figure 2, we show the graphs extracted from the images. We use 9 of the 10 graphs in the sequence for the purposes of training the linear model. The $8th$ graph in the sequence is retained for testing (recognition) and is matched to the model. The top-left panel of Figure 3 shows the modal graph. This graph is the one which has the largest cluster membership probability in the pre-clustering step. The top-right panel shows the test graph which is generated from the $8th$ image in the sequence. The mean graph is displayed in bottom-left panel. This graph is generated using the method described in section 4. The model fitting result is shown in the bottom-right panel. In the lower two panels of the figure, the darkness of the edges is proportional to the magnitude of the corresponding element of the adjacency matrix. In the case of the mean-graph, this quantity is proportional to the number of times the corresponding edge appears in the training data. It is interesting to note the similarities and differences in the structure of the mean and modal graphs. In the first instance, all the strong edges in the mean graph are common with the modal graph. Second, in the mean graph edges are pruned away from the high degree nodes. Hence, the learning process would appear to locate common salient structure, but remove ephemeral detail.

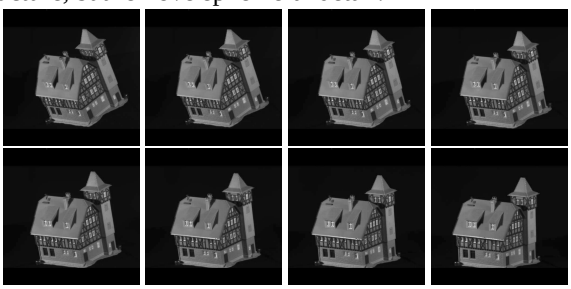

Fig. 1. CMU sequence

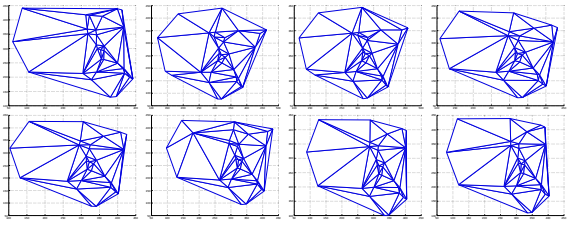

Fig. 2. Training graphs for the CMU sequence

6. CONCLUSIONS

In this paper, we have presented a framework for leaning a linear model of the modes of structural variation in sets of graphs. We commence by locating correspondences between the nodes in different graphs and using the correspondence order to vectorise the adjacency matrices. We

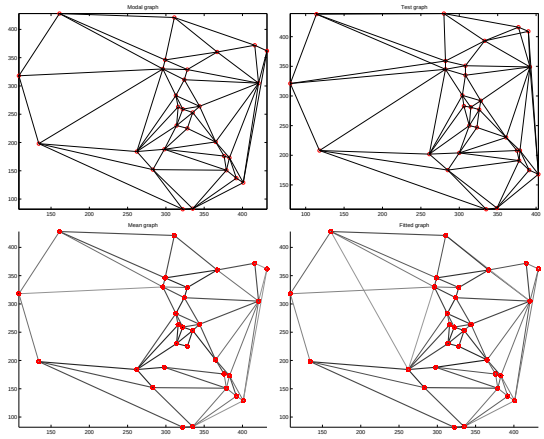

Fig. 3. Modal graph, test graph, mean graph and fitted graph for the CMU sequence

cluster the vectors using the EM algorithm. From the eigenmodes of the cluster covariance matrices we construct a linear model of the modes of structural variation in the graphs.

There are a number of ways in which we intend to develop this work. First, we aim to integrate the correspondence and clustering steps into a single process. Second, we aim to use the cluster covariance matrices to construct piecewise subspace models for the graphs.

7. REFERENCES

[1] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19:255–259, 1998.

[2] W.J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):749–764, 1995.

[3] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[4] B. Luo and E.R. Hancock. Structural graph matching using the em algorithm and singular value decomposition. *IEEE PAMI*, 23(10):1120–1136, 2001.

[5] A. Sanfeliu and K.S. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions Systems, Man and Cybernetics*, 13(3):353–362, May 1983.

[6] S. Sarkar and K.L. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. *CVIU*, 71(1):110–136, July 1998.

[7] K. Siddiqi, A. Shokoufandeh, S.J. Dickinson, and S.W. Zucker. Shock graphs and shape matching. *IJCV*, 35(1):13–32, November 1999.

[8] A. Robles-Kelly, E.R. Hancock. Pairwise Clustering with Matrix Factorisation and the EM Algorithm *ECCV* II: 63 ff. 2002