Spectral Analysis of Complex Laplacian Matrices

Richard C. Wilson and Edwin R. Hancock

Department of Computer Science
University of York, York Y01 5DD, UK

Abstract. This paper explores how to extend the spectral analysis of graphs to the case where the nodes and edges are attributed. To do this we introduce a complex Hermitian variant of the Laplacian matrix. Our spectral representation is based on the eigendecomposition of the resulting Hermitian property matrix. The eigenvalues of the matrix are real while the eigenvectors are complex. We show how to use symmetric polynomials to construct permutation invariants from the elements of the resulting complex spectral matrix. We construct pattern vectors from the resulting invariants, and use them to embed the graphs in a low dimensional pattern space using a number of well-known techniques including principal components analysis, linear discriminant analysis and multidimensional scaling.

1 Introduction

Spectral graph theory is concerned with understanding how the structural properties of graphs can be characterised using the eigenvectors of the adjacency matrix or the closely related Laplacian matrix (the degree matrix minus the adjacency matrix). There is a good introductory text on the subject by Biggs [5], and comprehensive reviews of recent progress in the field can be found in the research monograph of Chung [1], and the survey paper of Mohar [4], Although spectral methods have been extensively used to address the segmentation, or grouping [8] and correspondence matching [2] problems, there has been less work on using spectral characteristics to perform pattern analysis on sets of graphs and trees. Recently, however, there has been some work aimed at filling this gap in the literature. First, it has been shown how eigenvalues can be used to index shock trees [7]. Second, adjacency matrix eigenvectors can be used to construct simple structural attributes for graphs [3].

However, exsiting spectral methods are confined to the case of graphs with weighted nodes and edges, and do not easily extend to the case of attributed graphs. To overcome this problem in this paper, we explore the use of a richer property matrix representation which can be used with attributed graphs. Conventional spectral methods make use of the eigenvalues of the Laplacian matrix (i.e. the degree matrix minus the weight matrix). This allows only very limited information concerning the properties of the graph to be encoded. Our property matrix, on the other hand, allows more information concerning graphs to be encoded by allowing complex entries, rather than the purely real entries in

A. Fred et al. (Eds.): SSPR&SPR 2004, LNCS 3138, pp. 57–65, 2004.

the conventional Laplacian. To compute this matrix we multiply the off-diagonal elements of the Laplacian (i.e. the negative edge weights) by a complex number that encodes the edge attributes. The node attributes are encoded in the diagonal elements.

The property matrix is Hermitian, and hence it has real eigenvalues and complex eigenvectors. To characterise the properties of the graphs, we construct permutation invariants by applying symmetric polynomials to the real and imaginary components of the complex eigenvectors. The invariants are used as the components of pattern vectors for the shock graphs. Sets of shock graphs can be visualised and clustered by applying simple pattern analysis techniques to the pattern vectors. Here we investigate the use of principal components analysis, linear discriminant analysis and multidimensional scaling. We demonstrate the utility of the new method in the clustering of line-patterns and shock graphs.

2 Representation

Consider the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ with node-set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, edge-set $\mathcal{E} = \{e_1, e_2, \ldots, e_m\} \subset \mathcal{V} \times \mathcal{V}$ and weight function $\mathcal{W} : \mathcal{E} :\rightarrow [0, 1]$. The adjacency matrix $\mathbf{A}$ for the graph $\mathcal{G}$ is the $n \times n$ symmetric matrix with elements

$$A_{ab} = \begin{cases} 1 & \text{if } (v_a, v_b) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

In other words, the matrix represents the edge structure of the graph. Clearly if the graph is undirected, the matrix $\mathbf{A}$ is symmetric. The corresponding weighted adjacency matrix is defined to be

$$A_{ab} = \begin{cases} \mathcal{W}(v_a, v_b) & \text{if } (v_a, v_b) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

The Laplacian of the graph is given by $\mathbf{L} = \mathbf{D} - \mathbf{A}$. where $D_{a,b} = \sum_{b=1}^{n} A_{ab}$ is the diagonal node degree matrix whose elements are the number of edges which exit the node. The Laplacian is more suitable for spectral analysis than the adjacency matrix since it is positive semi-definite.

Here we would like to extend the Laplacian to weighted and attributed graphs. The attributes or weights may be unary (i.e. assigned to the nodes) or binary (i.e. assigned to the edges) in nature. To accommodate such structures, we need to augment the representation to accommodate measurement vectors on the nodes and edges. We doe this be encoding the weights and attributes using complex numbers.

A Hermitian matrix $\mathbf{H}$ is a square matrix with complex elements that remains unchanged under the operations of transposition and complex conjugation of the elements (denoted by the dagger operator †), i.e. $\mathbf{H}^{\dagger} = \mathbf{H}$. Hermitian matrices can be viewed as the counterpart of the symmetric matrix for complex numbers. Each off-diagonal element is a complex number which has two components, and can therefore represent a 2-dimensional measurement vector. The on-diagonal elements are necessarily real quantities, so the node measurements are limited to a single quantity.

There are some constraints on how the measurements must be represented in order to produce a positive semi-definite Hermitian matrix. Let $\{x_1, x_2, \ldots, x_n\}$ be a set of unary measurements for the node-set $\mathcal{V}$. Further, let $\{y_{1,2}, y_{1,3}, \ldots, y_{n,n}\}$ be the set of binary measurements associated with the edges of the graph. Each edge then has a pair of measurements $(\mathcal{W}_{a,b}, y_{a,b})$ associated with it. There are a number of ways in which the complex number $H_{a,b}$ could represent this information, for example with the real part as $\mathcal{W}$ and the imaginary part as y. However, the off-diagonal elements of $\mathbf{H}$ are chosen to be $H_{a,b} = -\mathcal{W}_{a,b}e^{iy_{ab}}$. In other words, the connection weights are encoded by the magnitude of the complex number $H_{a,b}$ and the binary measurement by its phase. By using this encoding, the magnitude of the numbers is the same as in the original symmetric matrix.

The measurements must satisfy the conditions $-\pi \leq y_{a,b} < \pi$ and $y_{a,b} = -y_{b,a}$ to produce a Hermitian matrix. To ensure a positive definite matrix, we require $H_{aa} > -\sum_{b\neq a} |H_{ab}|$. This condition is satisfied if $H_{aa} = x_a + \sum_{b\neq a} \mathcal{W}_{a,b}$ where $x_a \geq 0$. When defined in this way the matrix is the weighted Laplacian for the graph.

For a Hermitian matrix there is an orthogonal complete basis set of eigenvectors and eigenvalues obeying the eigenvalue equation $\mathbf{He} = \lambda\mathbf{e}$. In the Hermitian case, the eigenvalues λ are real and the components of the eigenvectors $\mathbf{e}$ are complex. There is a potential ambiguity in the eigenvectors, in that any multiple of an eigenvector is also a solution, i.e. $\mathbf{H}\alpha\mathbf{e} = \lambda\alpha\mathbf{e}$. In the real case, we choose α such that $\mathbf{e}$ is of unit length. In the complex case, α itself may be complex, and needs to determined by two constraints. We set the vector length to $|\mathbf{e}_i| = 1$ and in addition we impose $\arg \sum_{i=1}^{n} \mathbf{e}_i = 0$, which specifies both real and imaginary parts.

When the eigenvectors are constructed in this way the spectral matrix is found by performing the eigenvector expansion $\mathbf{H} = \sum_{i=1}^{n} \lambda_i \mathbf{e}_i \mathbf{e}_i^{\dagger}$, where λ_i and $\mathbf{e}_i$ are the n eigenvectors and eigenvalues of the Hermitian matrix $\mathbf{H}$. We construct the complex spectral matrix for the graph $\mathcal{G}$ using the eigenvectors as columns, i.e. $\mathbf{\Phi} = \left(\sqrt{\lambda_1}\mathbf{e}_1, \sqrt{\lambda_2}\mathbf{e}_2, \ldots \sqrt{\lambda_n}\mathbf{e}_n\right)$. The matrix $\mathbf{\Phi}$ is a complete representation of the graph in the sense that we can use it to reconstruct the original Hermitian property matrix using the relation $\mathbf{H} = \mathbf{\Phi}\mathbf{\Phi}^{\dagger}$.

3 Node Permutations and Invariants

The topology of a graph is invariant under permutations of the node labels. However, the adjacency matrix, and hence the Laplacian matrix, is modified by the node order since the rows and columns are indexed by the node order. If we relabel the nodes, the Laplacian matrix undergoes a permutation of both rows and columns, and the corresponding spectral matrix undergoes a permutation of columns only. In previous work, we showed how the spectral matrix can be characterised in a permutation invariant way using sets of symmetric polynomials. If the vector $\boldsymbol{\phi}_i = (\phi_{1,i}, \phi_{2,i}, ..., \phi_{i,n})^T$ represents a column of Φ, i.e. a spectral mode, then the elementary symmetric polynomials for the mode are given by

$$
\begin{aligned}
S_1(\boldsymbol{\phi}_i) &= \sum_{j=1}^{n} \phi_{j,i} \\
S_2(\boldsymbol{\phi}_i) &= \sum_{j=1}^{n} \sum_{k=i+1}^{n} \phi_{j,i}\phi_{k,i} \\
&\vdots \\
S_r(\boldsymbol{\phi}_i) &= \sum_{j_1<j_2<\ldots<j_r} \phi_{j_1,i}\phi_{j_2,i}\ldots\phi_{j_r,i} \\
&\vdots \\
S_n(\boldsymbol{\phi}_i) &= \prod_{j=1}^{n} \phi_{j,i}
\end{aligned}
$$

Since the components of the eigenvectors are complex numbers, and therefore each $\boldsymbol{\phi}_i$ is complex. the symmetric polynomials must be evaluated with complex arithmetic and also evaluate to complex numbers. Each S_r therefore has both real and complex components. The real and complex components of the symmetric polynomials are interleaved stacked to form a feature vector $\mathbf{B}_k$ for the graph.

In order to accommodate graph of different sizes, we need to be able to compare representations of different sizes. This is achieved by expanding the representation. Consider two graphs of size m and n, $m < n$. If we add $n - m$ nodes with no connections to the first graph, we obtain two graphs of the same size. The edit cost in terms of edge insertions and deletions between these two graphs is identical to the original pair. The effect on the spectral representation is merely to add trailing zeros to each eigenvector and also additional zero eigenmodes. As a consequence, the first m elementary symmetric polynomials are unchanged, and the subsequent $n - m$ are zero. The new representation in the symmetric polynomials S can therefore be easily calculated.

4 Graph Embedding Methods

We explore three different methods for embedding the graph feature vectors in a pattern space, namely principal components analysis (PCA), multidimensional scaling (MDS) and linear discriminant analysis (LDA). In this paper we are concerned with the set of graphs $\mathcal{G}_1, \mathcal{G}_2, .., \mathcal{G}_k, ..., \mathcal{G}_N$. The kth graph is denoted by $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k)$ and the associated vector of symmetric polynomials is denoted by $\mathbf{B}_k$.

4.1 Principal Components Analysis

Principal component analysis commences by constructing the matrix $\mathbf{S} = [\mathbf{B}_1 |\mathbf{B}_2| \ldots |\mathbf{B}_k| \ldots |\mathbf{B}_N]$. with the graph feature vectors as columns. Next, we compute the covariance matrix for the elements of the feature vectors by taking the matrix product $\mathbf{C} = \mathbf{S}\mathbf{S}^T$.. We extract the principal components directions by

performing the eigendecomposition $\mathbf{C} = \sum_{i=1}^{N} l_i \mathbf{u}_i \mathbf{u}_i^T$ on the covariance matrix $\mathbf{C}$, where the l_i are the eigenvalues and the $\mathbf{u}_i$ are the eigenvectors. We use the first s leading eigenvectors (2 or 3 in practice for visualisation purposes) to represent the graphs extracted from the images. The co-ordinate system of the eigenspace is spanned by the s orthogonal vectors $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, .., \mathbf{u}_s)$. The individual graphs are represented by the long vectors $\mathbf{B}_k, k = 1, 2, \ldots, N$ can be projected onto this eigenspace using the formula $\mathbf{x}_k = \mathbf{U}^T \mathbf{B}_k$.. Hence each graph G_k is represented by an s-component vector $\mathbf{x}_k$ in the eigenspace.

4.2 Multidimensional Scaling

Multidimensional scaling(MDS) is a procedure which allows data specified in terms of a matrix of pairwise distances to be embedded in a Euclidean space. 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, weighted by the variance of each feature. For the graphs indexed i_1 and i_2, the distance is $d_{i_1,i_2} = (\mathbf{B}_{i_1} - \mathbf{B}_{i_2})^T \Sigma_D^{-1} (\mathbf{B}_{i_1} - \mathbf{B}_{i_2})$ where Σ_D is a diagonal matrix with the feature variances on the diagonal. The pairwise similarities d_{i_1,i_2} are used as the elements of an $N \times N$ dissimilarity matrix $\mathbf{S}$.

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

We subject the matrix $\mathbf{T}$ to an eigenvector analysis to obtain a matrix of embedding co-ordinates $\mathbf{X}$. If the rank of $\mathbf{T}$ 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. $l_1 \geq l_2 \geq \ldots \geq l_k > 0$. The corresponding ordered eigenvectors are denoted by $\mathbf{u}_i$ where l_i is the ith eigenvalue. The embedding co-ordinate system for the graphs obtained from different views is $\mathbf{X} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_s]$, where $\mathbf{f}_i = \sqrt{l_i}\mathbf{u}_i$ are the scaled eigenvectors. For the graph indexed i, the embedded vector of co-ordinates is $\mathbf{x}_i = (X_{i,1}, X_{i,2}, ..., X_{i,s})^T$..

4.3 Linear Discriminant Analysis

Linear discriminant analysis is closely connected to PCA. We commence by constructing separate data matrices $S_1, S_2, \ldots$ for each class. These may be used to compute the corresponding class covariance matrices $C_i = S_i S_i^T$. The average class covariance matrix $C = \frac{1}{n}\sum_{i=1}^{n} C_i$ is the found. This matrix is used as a sphering transform by computing the eigendecomposition $C = U \Lambda U^T$ and using the transform $S' = \Lambda^{-\frac{1}{2}} U^T S$. Standard PCA is then applied to the resulting

data matrix S'. The purpose of this technique is to find a linear projection which describes the class differences rather than the overall variance of the data. Since we have a limited number of data samples, we use only the first twenty dimensions of the sphering transform.

5 Line Patterns

This first experimental example involves a database of the letters A-Z with rotations in 5 degree increments and hence 72 examples per character. Here we have used the method of Huet and Hancock [6] to compute pairwise attributes from the relative angles and lengths of the line-segments defining the characters. In the left-hand panel of Figure 1 shows the 11 closest retreivals of the letter "V" based on the Euclidean distance of the spectral feature vectors. The middlt panel of Figure 1 shows the result of performing MDS on the matrix of distances for the "A"s, "E"s and "Z"s in the database. The characters form well defined clusters.

To take this study on synthetic data one step further, we have performed a classification experiment. We have generated 100 graphs of 25 nodes each. For each graph the edge-structure is randomly generated. Associated with each edge is a weight randomly and uniformly drawn from the interval $[0, 1]$. We have investigated the effect of adding random noise to the edge-weights. The weight noise is drawn from a Gaussian distribution of zero mean and known standard deviation.

We have investigated the effect of this noise on three different vector representations of the attributed graphs. The first of these is a vector with the first four polynomial features as components. The second is a vector whose components are the bin-contents of the normalised edge-weight histogram. Here the edge weights are allocated to 8 uniformly spaced bins. The final vector has the leading 4 eigenvalues of the Laplacian matrix as components.

We have computed the distances between the feature vectors for the uncorrupted and noise corrupted graphs. To compute a classification error rate, we have recorded the fraction of times that the uncorrupted graphs do not have the smallest distance to the corresponding noise corrupted graph. The right-hand panel of Figure 1 shows the error-rate as a function of the edge-weight noise standard deviation. The main feature to note from this plot are that the lowest error rate is returned by the polynomial features and the highest error rate results from the use of the edge-weight histogram.

6 Shock Graphs

The second example of the use of the complex property matrix representation is furnished by shock trees, which are an abstraction of the skeleton structure of 2D or 3D shape silhouettes. The skeleton is the locus of the centre of the bitangent circle to the object boundary, and is hence related to the medial axis transform (which seeks points which are equidistant from pairs of points on the

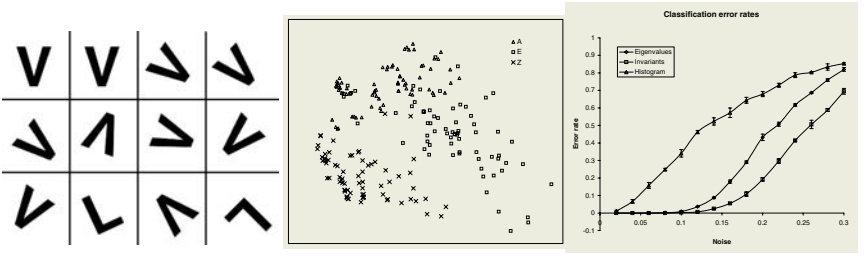

Fig. 1. Retreival results (left), MDS (middle) and classification error rate (right) for character data.

shape boundary). In practice searching for the skeleton corresponds to finding ridges in the distance transform of the object boundary. The medial axis has a natural tree structure which makes it suitable for the tree representation of shapes.

The edges of the tree represent the existence of a connecting skeletal branch between pairs of junctions. The nodes of the tree are characterised using the radius $r(a)$ of the smallest bitangent circle from the junction to the boundary. Hence, for the node a, $x_a = r(a)$. The edges are characterised by two measurements. For the edge (a, b) the first of these, $y_{a,b}$ is the angle between the nodes a and b, i.e. $y_{a,b} = \theta(a, b)$. Since most skeleton branches are relatively straight, this is an approximation to the angle of the corresponding skeletal branch. Furthermore, since $-\pi \leq \theta(a, b) < \pi$ and $\theta(a, b) = -\theta(b, a)$, the measurement is already suitable for use in the Hermitian Laplacian matrix.

In order to compute edge weights, we note that the importance of a section of the skeleton may be determined by the rate of change of boundary length with skeleton length [9], which we denote by dl/ds. This quantity is related to the rate of change of the bitangent circle radius along the skeleton, i.e. dr/ds, by the formula $\frac{dl}{ds} = \sqrt{1 - \left(\frac{dr}{ds}\right)^2}$. The edge weight $\mathcal{W}_{a,b}$ is given by the average value of dl/ds along the relevant skeletal branch.

Our experiments are performed using a database of 42 binary shapes. Each binary shape is extracted from a 2D view of a 3D object. There are 3 classes in the database, and for each object there are a number of views acquired from different viewing directions and a number of different examples of the class . We extract the skeleton from each binary shape and attribute the resulting tree in the manner outlined in Section 4.

We commence by showing some results for the three shapes shown in Figure 1. The objects studied are a hand, some puppies and some cars. The dog and car shapes consist of a number of different objects and different views of each object. The hand category contains different hand configurations. We apply the three embedding strategies outlined in Section 5 to the vectors of permutation invariants. We commence in the left-hand panel of Figure 2 by showing the result of applying MDS procedure to the three shape categories. The 'hand' shapes form a compact cluster in the MDS space. There are other local clusters consisting

of three or four members of the remaining two classes. This reflects the fact that while the hand shapes have very similar shock graphs, the remaining two categories have rather variable shock graphs because of the different objects.

The middle panel of Figure 2 shows the result of using PCA. Here the distributions of shapes are much less compact. While a distinct cluster of hand shapes still occurs, they are generally more dispersed over the feature space. There are some distinct clusters of the car shape, but the distributions overlap more in the PCA projection when compared to the MDS space.

Fig. 2. MDS (left), PCA (centre) and LDA (right) applied to the shock graphs.

In the LDA projection, we introduce information about the class designations of the shape trees. The right-hand panel of Figure 2 shows the result of the LDA procedure on the dataset. The result is a much better class separation than the PCA or MDS methods.

Based on the analysis of the different embedding methods, it appears that LDA gives the best results. One of the motivations for the work presented here was the potential ambiguities that are encountered when using the spectral features of trees. To demonstrate the effect of using attributed trees rather than simply weighting the edges, we have compared the LDA projections using both types of data. Figure 3 illustrates the result of this comparison. The right-hand plot shows the result obtained using the symmetric polynomials from the eigenvectors of the Laplacian matrix $L = D - W$, associated with the edge weight matrix. The left-hand plot shows the result of using the using the Hermitian property matrix. The Hermitian property matrix for the attributed trees clearly produces a better class separation than the Laplacian matrix for the weighted trees.

7 Conclusions

In this paper we have described a complex property matrix that can be used to encode the structure of attributed graphs. We have shown how to construct permutation invariants from the complex components of the eigenvectors of the Hermitian property matrix. The invariants are used as the components of a real-valued pattern vector, which can be embedded in a pattern space, suitable for clustering the graphs. There are clearly a number of ways in which the work presented in this paper can be developed. For instance, since the representation

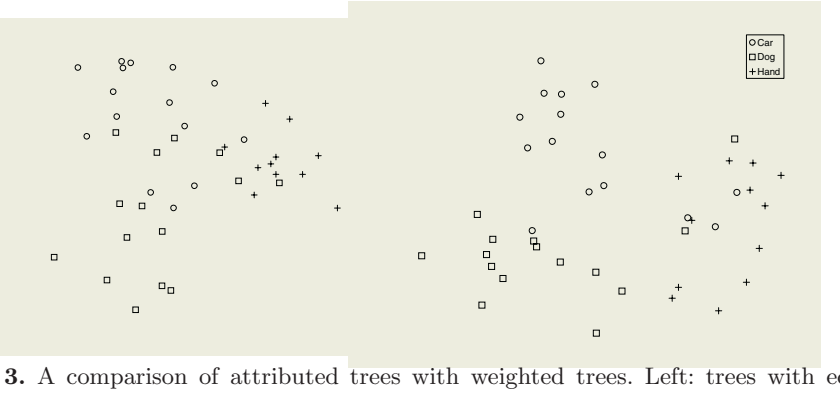

Fig. 3. A comparison of attributed trees with weighted trees. Left: trees with edge weights based on boundary lengths. Right: Attributed trees with additional edge angle information.

based on the symmetric polynomials is complete, they may provide the means by which a generative model of variations in graph structure can be developed. This model could be learned in the space spanned by the permutation invariants, and the mean graph and its modes of variation reconstructed by inverting the system of equations associated with the symmetric polynomials.

References

1. F.R.K. Chung. *Spectral Graph Theory.* American Mathmatical Society Ed., CBMS series 92, 1997.
2. S. Umeyama. An eigen decomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:695–703, 1988.
3. B. Luo, R. C. Wilson and E. R. Hancock, "Spectral Embedding of Graphs", *Pattern Recognition* 36(10), pp. 2213–2230, 2003
4. B.Mohar,"Laplace Eigenvalues of Graphs - A survey", *Discrete Mathematics*, **109**, pp. 171-183, 1992.
5. N. Biggs, "Algebraic Graph Theory", CUP.
6. B. Huet and E. R Hancock, "Line Pattern Retrieval Using Relational Histograms", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**, pp. 1363-1370, 1999.
7. K. Siddiqi, A. Shokoufandeh, S. J. Dickinson and S. W. Zucker "Shock Graphs and Shape Matching", International Journal of Computer Vision, 35(1), pp13–32, 1999.
8. J. Shi and J. Malik, "Normalized Cuts and Image Segmentation" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8) pp888–905, 2000
9. A. Torsello and E. R. Hancock, "A skeletal measure of 2D shape similarity", *Lecture notes on Computer Science*, 2059, pp594–605, 2001