

Mixing spectral representations of graphs

David White and Richard C. Wilson
Department of Computer Science
University of York
Heslington, York, UK
wilson@cs.york.ac.uk

Abstract

Generative models are well known in the domain of statistical pattern recognition. Typically, they describe the probability distribution of patterns in a vector space. The individual patterns are defined by vectors and so the individual features of the pattern are well defined. In contrast, very little has been done with generative models of graphs. Graphs are not naturally represented in a vector space since there is no natural labelling of the vertices of the graphs - different labellings lead to different representations of the graph structure. Because of this, simple statistical quantities such as mean and variance are difficult to define for a group of graphs. While we can define statistical quantities of individual edges, it is not so straightforward to define how sets of edges in graphs are related. The spectral decomposition of a graph can be used to extract information about the relationship of edges and parts in a graph. In this paper we look at the problem of mixing graphs by using the spectral representation of a graph as an intermediate step. The spectral representation allows us to mix different structural features from each of the graphs to create new combinations. We can also define an averaging process on the spectral representations which generates a graph close to the graph median.

1 Introduction

Generative models are well known in the domain of statistical pattern recognition. Typically, they describe the probability distribution of patterns in a vector space. The individual patterns are defined by vectors and so the individual features of the pattern are well defined. Graphs are not naturally represented in a vector space since there is no natural labelling of the vertices of the graphs - different labellings lead to different representations of the graph structure. Because of this, simple statistical quantities such as mean and variance are difficult to define for a group of

graphs.

The key problem in utilising graph representations lies in measuring their structural similarity. This is a difficult problem because there is no explicit labelling of the parts, and typically correspondences must be established before similarity can be assessed. As an example, Sanfeliu and Fu[8] employed the concept of graph edit distance, giving separate edit costs for relabeling, insertion and deletion on both nodes and edges. A search is necessary to locate the set of operations which have minimal cost. More recently, Bunke[1, 2] has established a relationship between the minimum graph edit distance and the size of the maximum common subgraph. The graph edit distance provides a well defined way of measuring the similarity of two graphs.

Spectral graph theory provides another approach to the problem of graph similarity[3]. Eigenvector methods have been used for grouping via pairwise clustering. Examples include Shi and Malik's [10] iterative normalised cut method which uses the Fiedler (i.e. second) eigenvector for image segmentation and Sarkar and Boyer's use of the leading eigenvector of the weighted adjacency matrix [9]. Graph spectral methods have also been used to correspondence analysis. For example, Umeyamas method[12] allows the matching of two graphs of equal size by using the eigendecompositions of the adjacency matrices. Kosinov and Caelli[7] have used properties of the spectral decomposition to represent graphs and Shokoufandeh et al[11] has used eigenvalues of shock graphs to index shapes. Wilson and Hancock have shown[13] how permutation invariant polynomials can be used to derive features which describe graphs and make full use of the available spectral information. Ferrer et al[4] have used the spectral decomposition to match graphs in an intermediate step to finding the graph median.

The problem of representing a set of graphs via a representative has been addressed by Jiang et al[6, 5], who define the median of a set of graphs to be the graph which has the minimum sum-of-distances to all the graphs in the set. This definition is equivalent in a sense to the mean of a set of

points. The computation of such median graphs is computationally expensive and Jiang et al employ a genetic algorithm to locate median graphs. Ferrer et al[4] use a spectral method to compute correspondences between the eigendecompositions of graphs, and then update the median graph using these correspondences. The updates are performed sequentially.

The spectral decomposition itself provides a partial ordering of the graph representation through the order of eigenvalues. By completing the ordering process, we can identify which parts of the graph correspond to each other and so define the statistical quantities such as mean and variance which are required for a generative model. The other key issue in a generative model, the ability to reconstruct patterns from points in sample space, is also addressed partially by using the spectral representation, since it is possible to generate candidate graphs from their spectral representations. The spectral decomposition also represents principle parts of the graphs, so we can use the representation in order to mix different parts of graphs.

The outline of the paper is as follows; In section 2 we describe various matrix representations of graphs, which are key to the reconstruction process. In section 3, we describe how we order the spectral representations and mix them together. In section 4, the reconstruction process is detailed. Finally, we provide some experimental analysis.

2 Spectral representations

The graphs under consideration here are undirected graphs. The methods employed here also apply to weighted graphs, but the case of a weighted graph we need not ultimately concern ourselves with recovering discrete edges. We denote a graph by $G = (V, E)$ where V is the set of nodes and $E \subseteq V \times V$ is the set of edges. The degree of a vertex u is the number of edges incident on the vertex u and is denoted d_u . A *matrix representation* of the graph is a $|V|$ by $|V|$ matrix \mathbf{X} , such that an element X_{ij} of this matrix represents some property of the pair of vertices i and j . Diagonal elements X_{ii} encode information about the vertex i only. A simple example is the adjacency matrix A , where A_{ij} is 1 when there is an edge between i and j , and zero otherwise. There are a number of alternative representations which we discuss below.

The *spectral representation* of the graph is obtained from the matrix representation using the eigendecomposition. Let \mathbf{X} be the matrix representation in question. Then the eigendecomposition is $\mathbf{X} = \Phi\Lambda\Phi^T$ 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. The spectrum is the set of eigenvalues

$$s = \{\lambda_1, \lambda_2, \dots, \lambda_{|V|}\}$$

The spectral representation is the pair $\{\Phi, \Lambda\}$ which completely describe the graph, since the original adjacency matrix can be constructed from the spectral representation. If \mathbf{X} is positive semi-definite, then we can combine these matrices to define the spectral representation by a single matrix $\Phi\Lambda^{\frac{1}{2}}$.

The spectral representation is not however a unique description of the graph. If we reorder the vertices of G , then although the graph is unchanged, we obtain a different matrix and therefore a different spectral representation. If \mathbf{P} is the permutation matrix which re-orders the vertices, then

$$\mathbf{X}' = \mathbf{P}\mathbf{X}\mathbf{P}^T$$

represents the same graph as \mathbf{X} . The spectral representation $\{\mathbf{P}\Phi, \Lambda\}$ therefore represents the same graph for any permutation matrix \mathbf{P} . In addition if there are repeated eigenvalues in the spectrum, then the eigendecomposition is not uniquely defined.

2.1 Standard Graph Representations

The most basic matrix representation of a graph is using the *adjacency matrix* A for the graph. This matrix is given by

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

Clearly if the graph is undirected, the matrix A is symmetric. As a consequence, the eigenvalues of A are real. These eigenvalues may be positive, negative or zero and the sum of the eigenvalues is zero. The eigenvalues may be ordered by their magnitude and collected into a vector which describes the graph spectrum.

In some applications, it is useful to have a positive semidefinite matrix representation of the graph. This may be achieved by using the *Laplacian*. We first construct the diagonal degree matrix \mathbf{D} , whose diagonal elements are given by the node degrees $D(u, u) = d_u$. From the degree matrix and the adjacency matrix we then can construct the standard combinatorial Laplacian matrix

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \quad (2)$$

i.e. the degree matrix minus the adjacency matrix. The Laplacian has at least one zero eigenvalue, and the number of such eigenvalues is equal to the number of disjoint parts in the graph.

The *signless Laplacian* has all entries greater than zero and is defined to be

$$|\mathbf{L}| = \mathbf{D} + \mathbf{A} \quad (3)$$

3 Mixing spectral representations

Because graphs are not vectorial quantities, many operations which are straightforward with vectors are difficult to do with graphs. For example, finding the mean of a set of graphs is non-trivial task. An number of authors have looked at the problem of averaging graphs. Jiang et al[6] defined the median graph and proposed a genetic algorithm to find the median. Ferrer et al[4] suggested using spectral alignment of the graphs to speed up the process of finding median graphs. In contrast to these approaches which look for median adjacency matrices, we propose the direct mixing or averaging of spectral modes.

The spectral representation is an interesting one in terms of mixing graphs for a number of reasons. Firstly, part of the correspondence problem is solved in the spectral representation; the columns of Φ are ordered by the eigenvector magnitude which is not affected by the vertex labelling. However, the rows are still permuted when we change the graph indexing. Secondly, in the Laplacian and related matrices, structures of different scales in the graph are associated with eigenvalues of different magnitudes. It is well known that the Fiedler vector of the normalised Laplacian can be used to partition the graph into parts, whereas the principle eigenvector represents global structure in the graph. As a result, it is possible to mix different scales separately using the spectral representation.

3.1 Spectral alignment and mixing

Before mixing the spectral representations of two graphs, we must first align the rows of Φ so that they are in the same order. This may be done using a spectral graph matching method such as Umeyama's method[12] or a variant such as that of Ferrer et al[4]. In order for the process of mixing spectral modes to be effective, the eigensystems of the graphs must be relatively similar. These spectral methods of alignment should be effective on such graphs. In the experiments detailed below, the graphs are already correctly aligned.

It is well known that the eigenvectors of the decomposition of a matrix are sign-ambiguous. In other words, the eigenvectors are recovered up to a sign factor of ± 1 . It is necessary to determine these factors if we are to correctly mix the corresponding eigenmodes. Our method is based on identifying the largest component of an eigenvector and correcting the sign based on that coordinate. Given a set of spectral matrices $\{\Phi_1, \Phi_2, \dots, \Phi_m\}$, let ϕ_{ij} be the j th eigenvector (mode) from Φ_i . The k th component of this eigenvector can then be denoted ϕ_{ijk} . We find the largest magnitude component for mode j from

$$l_j = \arg \max_k \sum_i |\phi_{ijk}|$$

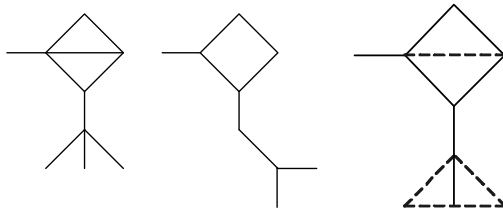


Figure 1. Example graphs and the spectral average graph. Dotted edges have weights of close to 1/2.

We then correct the sign of the eigenvectors by ensuring that component l_j is positive for mode j in all the spectral matrices.

Once aligned, the spectral matrices may be merged by simply taking the average of the matrices, i.e.

$$\Phi_m = \frac{1}{2}(\hat{\Phi}_1 + \hat{\Phi}_2)$$

$$\Lambda_m = \frac{1}{2}(\hat{\Lambda}_1 + \hat{\Lambda}_2)$$

which will give a combination of two graphs. Figure 1 demonstrates an example of this mixing process. Figure 1 shows the original graphs and the result of reconstruction from the mixed spectral representation using the method detailed in the next section.

The spectral representation is a particularly flexible one for mixing graphs since there is a separation of different scales of the graph in the ordering of the eigenvalues. It is therefore possible to mix graphs by selecting different parts of the spectrum from each of the graphs being mixed. This enables the selection of different parts of the structure from the different graphs. In order to achieve this, we define a mixing matrix thus:

$$M_i = \begin{pmatrix} f_{i,1} & 0 & \dots & 0 \\ 0 & f_{i,2} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & f_{i,n} \end{pmatrix}, \sum_i M_i = I$$

The diagonal elements f_{ij} define the fraction of mode j which is selected from the spectrum of graph i . The mixed spectral matrices are then defined as

$$\Phi_m = \sum_i \Phi_i M_i \quad (4)$$

$$\Lambda_m = \sum_i \Lambda_i M_i \quad (5)$$

Figure 2 shows the results of mixing the two example graphs from Figure 1 in different proportions. On the left, the modal proportions are $[1, 0.5, 0, 0, 0, 0, 0]$ and

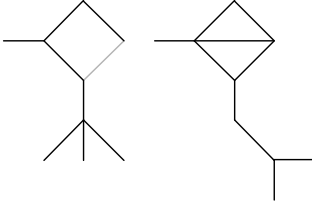


Figure 2. Mixing spectral modes in different proportions from two graphs

$[0, 0.5, 1, 1, 1, 1, 1, 1]$ from the first and second graphs respectively. On the right, the proportions are reversed, i.e. $[0, 0.5, 1, 1, 1, 1, 1, 1]$ and $[1, 0.5, 0, 0, 0, 0, 0, 0]$. By selecting the proportions we are able to mix key structures from each of the two graphs.

4 Reconstruction

With the merged spectral representations to hand, we can reconstruct a graph using the reverse of the eigendecomposition:

$$X_m = \Phi_m \Lambda_m \Phi_m^T$$

In general, the averaging process will lead to a matrix X which does not have the required properties. Firstly, the averaged spectral matrix Φ_m will not be orthonormal. In order to correct this we apply the Gram-Schmidt orthogonalisation procedure to Φ_m to obtain an orthonormal spectral matrix.

Secondly, the reconstructed matrix representation may not be consistent with the chosen representation. For example, if we are operating with Laplacians, the diagonal elements will not be the vertex degrees. In addition, the edges will be weighted. There is therefore a need to project X_m onto the nearest graph.

It is tempting to interpret the weights as edge probabilities. In fact we could select edges in the final conditioned Laplacian with a probability equal to their weights in X_m . This approach would generate sets of random graphs which are close to the original graph. On the other hand, this ignores the co-occurrence of edges in the original graphs. A simple and satisfactory solution is to use a threshold for edges and non-edges. An element of X_m is considered an edge if it is greater than θ , an adjustable threshold.

5 Experimental results

In the first set of experiments, we take a graph set and determine the spectral average graph by the method described above. The graphs have eight vertices with a common core graph consisting of four vertices. This is linked to another graph of four vertices which contains random edges for each

of the graphs in the set. The goal is to find a graph which is similar to all graphs of the set. In order to visualise the results, we have produced an MDS plot based on the edit distances between the graphs. For comparison, we have found the median graph as defined by Jiang et al[5], which is defined by

$$G_m = \arg \min_G \sum d(G_i, G)$$

where $d(\cdot)$ is the edit distance. We can also use $e = \sum d(G_i, G)$ as a measure of the quality of the average graph.

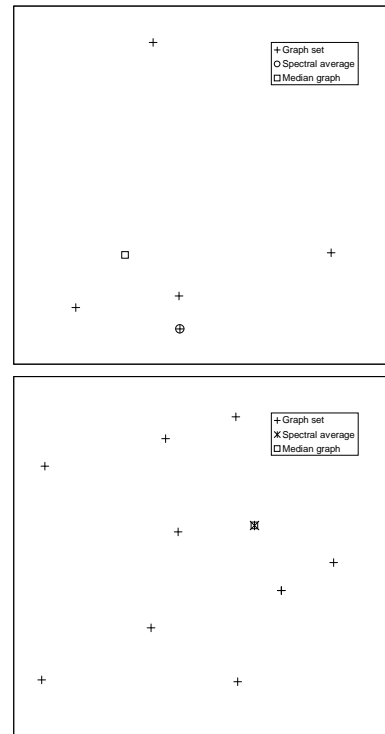


Figure 3. MDS projection of graph sets with spectral average and median graphs: Top - a 5 graph set, bottom - a 10 graph set

Figure 3 shows the results for two graph sets. The set on the left has five graphs. The spectral average graph has a total distance of $e = 10$ and differs by one edge from the median graph which has $e = 9$. On the right, ten graphs are used in the set. Here the median and spectral average graphs are identical and have $e = 22$, less than any graphs from the original set.

One of the key benefits of mixing graphs in spectral representations is the ability of the representation to separate different structural parts of the graph. In order to examine the effectiveness of this approach, we have constructed a second experiment. In this experiment we generate two graph classes. Both classes are generated by joining two core structures at a common vertex. Let the core graphs

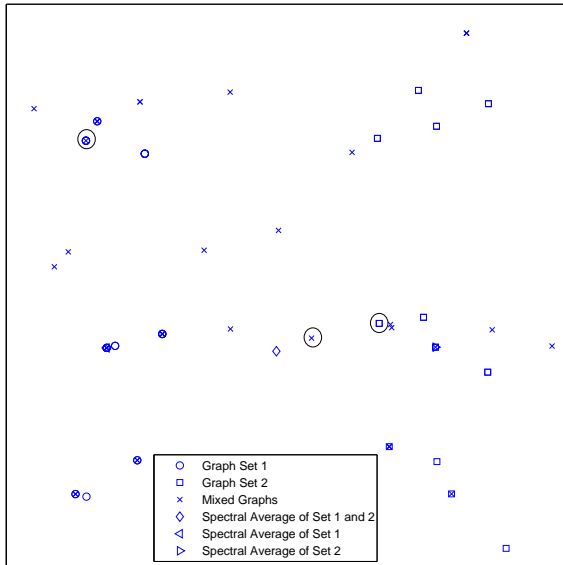


Figure 4. Mixing the spectral modes of different graph sets. The circled points consist of examples from set 1, set 2 and a mixed graph.

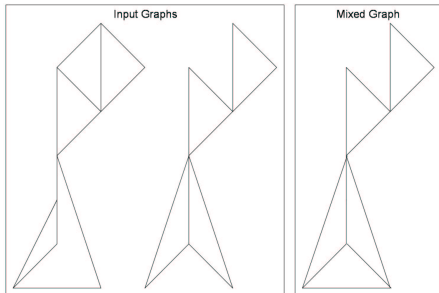


Figure 5. The mixed graphs circled in Figure 4.

be C_1, C_2, C_3 . Then set one is constructed by joining C_1 and C_2 , and set two by joining C_1 and C_3 . In the results below, the core graphs are 5 vertices. Before constructing each example, the core graph is perturbed by a random edge edit operation. As a result we obtain two sets which are internally very similar and between sets slightly less than half the graph is similar. By spectrally mixing a pair from each set, we hope to obtain graphs which have parts from both sets and so lies between them. The results are shown in Figure 4. Figure 5 shows the example graphs circled in Figure 4. The two graph sets form their own individual groups, and we have successfully generated example which lie over the whole space spanned by the two sets and in between.

References

- [1] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18:689–694, 1997.
- [2] 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.
- [3] F. R. K. Chung. *Spectral Graph Theory*. AMS, 1997.
- [4] M. Ferrer, F. Serratoso, and A. Sanfeliu. Synthesis of median spectral graph. In *IbPRIA*, pages 139–146, 2005.
- [5] X. Jiang, A. Munger, and H. Bunke. Computing the generalized median of a set of graphs. In *Proc. 2nd IAPR workshop on Graph Based representations*, pages 115–124, 1999.
- [6] X. Jiang, A. Munger, and H. Bunke. On median graphs: Properties, algorithms and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1144–1151, 2001.
- [7] S. Kosinov and T. Caelli. Inexact multisubgraph matching using graph eigenspace and clustering models. *Structural, Syntactic and Statistical Pattern Recognition, LNCS*, 2396:133–142, 2002.
- [8] A. Sanfeliu and K. S. Fu. A distance measure between attributed relational graphs for pattern-recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 13(3):353–362, 1983.
- [9] S. Sarkar and K. L. Boyer. Preceptual organization in computer vision. *IEEE Trans. Systems, Man and Cybernetics*, 23:382–399, 1993.
- [10] J. Shi and J. Malik. Normalized cuts and image segmentation. *CVPR*, pages 731–737, 1997.
- [11] A. Shokoufandeh, S. Dickinson, K. Siddiqi, and S. Zucker. Indexing using a spectral coding of topological structure. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, number 491–497, 1999.
- [12] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):695–703, 1988.
- [13] R. C. Wilson, E. R. Hancock, and B. Luo. Pattern vectors from algebraic graph theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1112–1125, 2005.