

An Inexact Graph Comparison Approach in Joint Eigenspace

Wan-Jui Lee and Robert P.W. Duin

Faculty of Electrical Engineering, Mathematics and Computer Sciences,
Delft University of Technology, The Netherlands
W.J.Lee@tudelft.nl, r.duin@ieee.org

Abstract. In graph comparison, the use of (dis)similarity measurements between graphs is an important topic. In this work, we propose an eigendecomposition based approach for measuring dissimilarities between graphs in the *joint* eigenspace (JoEig). We will compare our JoEig approach with two other eigendecomposition based methods that compare graphs in *different* eigenspaces. To calculate the dissimilarity between graphs of different sizes and perform inexact graph comparison, we further develop three different ways to resize the eigenspectra and study their performance in different situations.

1 Introduction

Graphs are a general and powerful data structure for object representation in structural pattern recognition[11,2]. The nodes in a graph can represent different parts of an object and the relations between the parts are represented by edges. Also, labels and attributes for the nodes and edges can further be used to incorporate more information in a graph representation. But for simplicity, we only consider nodes and edges without labels and attributes in this work.

In pattern recognition and other related areas, (dis)similarity among objects is an important issue. If the graphs are used for object representation, the problem turns into determining (dis)similarity between objects, which is usually referred to as graph comparison. In the study of graph comparison, there are two main directions. One is by comparing their spatial structures and the other is by comparing their spectral structures. Early approaches to graph comparison were mainly based on the first. The best known example is maximum common subgraph [2] where the similarity of two graphs is decided by how much they spatially share in common. An alternative is the graph edit distance [8]. In graph edit distance computation, one introduces a set of graph edit operations, such as deletion, insertion or substitution, to apply on nodes as well as edges. The edit distance of two graphs is defined as the shortest sequence of edit operations that transforms one graph into the other. These methods for finding spatial (dis)similarity between graphs are guaranteed to find the optimal solutions but require exponential time and space due to the NP-completeness of the problem.

In 1988, Umeyama [6] introduced the spectra of graphs for the exact graph comparison problem. This approach is based on spectral graph theory that is concerned with characterising the structural properties of graphs using the eigenvectors of the adjacency matrix or the closely related Laplacian matrix (the degree matrix minus the adjacency matrix). To compare two different graphs A and B , the idea is to transform graph A into the space of graph B . So there exists an approximated graph A' of graph A in the space of graph B . The difference between A and B is actually calculated by the difference between A' and B . There is another approach for inexact graph comparison based on graph spectra proposed by Caelli [1]. Instead of transforming graph A into the space of graph B , it projects graph A into its own eigenspace and the same is done for graph B . Because graph A and B could have different numbers of nodes, a renormalization step is introduced for eigenspace projections. Therefore, both graphs are projected into a unit hypersphere defining the eigenspaces. In this renormalized space, two graphs are close if their rescaled eigenvectors have similar angles and their eigenvalues are similar. The authors declare that the graph comparison was done in a common eigenspace of these two graphs, but actually it's done in their own eigenspaces but with the common unit hypersphere.

In Umeyama's method, the graphs are compared in the original space of one of the graphs, but this is not unique since there are two graphs. As a matter of fact, we further discover that the projection of graph A in the space of graph B is actually determined by how the eigenvalues of graph A live in the eigenspace of graph B . So if we want to have a symmetric dissimilarity measurement using Umeyama's method, we have to take an average or choose one of the results from two different eigenspaces, i.e. the eigenspace of A or B . In Caelli's approach, the dissimilarities of the graphs are compared by their projections in their own eigenspaces. For a fair comparison of two graphs, we will bring them into the same eigenspace and then compare them in this unique eigenspace. Inspired by Umeyama's approach, we actually can find an unique joint eigenspace (JoEig) for both graphs, which is a space with the eigenvectors of both graphs. However, the sizes of these two sets of eigenvectors could be different and therefore it is essential to adjust these two sets of eigenvectors for sharing the same number of dimensions in order to construct the joint eigenspace. We study three possibilities for setting the number of eigenvectors in this work. The first two choices are setting the number of the eigenvectors according to the size of the larger or the smaller graph, respectively. We study as well for a number smaller than both of the graphs to emphasize the importance of larger eigenvalues. We will compare these three settings and discuss how the dimensionality of eigenvectors effects the performance of classifiers in the experiments.

The rest of the paper is organized as follows. In Section 2, we introduce Umeyama's and Caelli's approaches for comparing graphs, respectively. Our graph dissimilarity measurement approach, JoEig, is described in Section 3. Simulation results are presented in Section 4. Finally, a conclusion is given in Section 5.

2 Previous Eigendecomposition Approaches for Graph Comparison

Before we introduce Umeyama’s and Caelli’s approaches for graph comparison, some definitions and introduction on graphs are given as in the following.

A graph is a set of nodes connected by edges in its most general form. Consider the undirected graph $G = (V, E, W)$ with the node set $V = \{v_1, v_2, \dots, v_n\}$, the edge set $E = \{e_1, e_2, \dots, e_m\} \subset V \times V$, and the weight function $W : E \rightarrow (0, 1]$. If the graph edges are weighted, the adjacency matrix A for the graph G is the $n \times n$ matrix with elements

$$A_{ij} = \begin{cases} W(v_i, v_j), & \text{if } (v_i, v_j) \in E; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Clearly since the graph is undirected, the matrix A is symmetric. The Laplacian [10] of the graph is defined by $L = D - A$, where D is the diagonal node degree matrix whose elements $D_{ii} = \sum_{k=1}^n A_{ik}$. The Laplacian matrix of G is positive semidefinite and singular, and it is more often adopted for spectral analysis than the adjacency matrix because of its properties.

2.1 An Eigendecomposition Approach for the Weighted Graph Matching Problem

In 1988, Umeyama [6] developed an eigendecomposition approach for the weighted graph matching problem. Let $G = (V_1, E_1, W_1)$, $H = (V_2, E_2, W_2)$ be weighted graphs with n nodes (these two graphs are with the same size). The weighted graph matching problem is the problem of finding a one-to-one correspondence Φ between $V_1 = \{v_1, v_2, \dots, v_n\}$ and $V_2 = \{v'_1, v'_2, \dots, v'_n\}$ which minimizes the difference between G and H . The following criterion is used as a measure of difference:

$$J(P) = \|PL_G P^T - L_H\|^2 \quad (2)$$

where the permutation matrix P presents the node correspondence Φ and $\|\cdot\|$ is the Euclidean norm ($\|L\| = (\sum_{i=1}^n \sum_{j=1}^n l_{ij}^2)^{\frac{1}{2}}$). Thus the weighted graph matching problem is reduced to the problem of finding the permutation matrix P which minimizes $J(P)$. In general, it is not easy to find the exact solution of the weighted graph matching problem since it is purely combinatorial. Thus, an efficient method which gives a "nearly" optimum solution, i.e., a permutation matrix P' whose criterion value $J(P')$ is very close to the optimum value. Since the optimum criterion value cannot be known in advance, the aim is to determine a permutation matrix of a small criterion value.

Let G and H be weighted undirected graphs and L_G and L_H be their Laplacian matrices, respectively. The eigendecomposition of L_G and L_H are performed as

$$L_G = V_G D_G V_G^T, \quad L_H = V_H D_H V_H^T, \quad (3)$$

where V_G and V_H are orthonormal matrices and D_G and D_H are diagonal matrices of the eigenvalues (in ascending order) of G and H , respectively. Now if G and H are isomorphic,

$$PV_G D_G V_G^T P^T = V_H D_H V_H^T. \quad (4)$$

It means there exists a permutation matrix P that has the one-to-one correspondence of nodes between G and H . Also, the eigenvalues of these two graphs should be the same. That is, $D_G = D_H$, and therefore $PV_G = V_H$ which leads to $P = V_H V_G^T$. As a result, the difference between two graphs can be computed as

$$\|V_H V_G^T L_G V_G V_H^T - L_H\|^2, \text{ or } \|V_G V_H^T L_H V_H V_G^T - L_G\|^2. \quad (5)$$

Even though Umeyama's approach is only designed for graphs with the same size, we extend it in this paper to compare it with the other approaches. First of all, the Laplacian matrix is used instead of adjacency matrix. Secondly, the eigenvalue and eigenvector matrices are resized because the graphs might be with different numbers of nodes. We study three different ways for computing the resized eigenvalue diagonal matrix and the eigenvector matrix, and they will be discussed further in the next section. Finally, the difference between two graphs is the difference between two approximations instead of the one between the original graph and its approximation. This is to prevent asymmetry which can easily be seen in Eq.(5). The difference between two graphs can be different if the original graph is different. Therefore we use a new criterion for Umeyama's approach to make the results symmetric, and that is

$$\|V_H V_G^T L_G V_G V_H^T - V_G V_H^T L_H V_H V_G^T\|^2 = \|V_H D_G V_H^T - V_G D_H V_G^T\|^2. \quad (6)$$

By observing Eq.(6), we can see that the approximation of one graph in Umeyama's approach is by relocating the other graph's eigenvalues in its eigenspace. This suggests that the difference between these two approximations is the difference derived from two different eigenspace.

2.2 An Eigenspace Projection Clustering Method for Inexact Graph Matching

Subspace projection methods are conventionally used to reduce the dimensionality of data by minimizing the number of dimensions and the amount of information loss. To solve the inexact graph matching problem, Caelli [1] proposed an approach to project graphs into their subspaces. Furthermore, to compare graphs in different subspaces, these subspaces should have the same dimensionality. This was done by setting a number k to fix the dimensionalities of these subspaces. In Caelli's approach, the graph Laplacian matrix, L , is first decomposed into the familiar eigenvalue and eigenvector matrix product $L = VD V^T$ as in Eq.(3). The original data is then projected onto a smaller number of k most important (i.e., the k largest eigenvalues) principal components.

To compare the similarities of two vertices from two graphs of different sizes, an additional renormalization step for such projections was introduced. Both eigenvector (V) and eigenvalue (D) matrices are truncated according to the chosen number of projection dimensions (k)-largest eigenvalues and their associated eigenvectors. The renormalized eigenvectors and eigenvalues become: $V'_k = \frac{V_k}{\|V_k\|}$, $D'_k = \frac{D_k}{\|D_k\|}$. As a result, the renormalized subspace projection of each vertex (column of L) was obtained as $L'_k = D'_k(V'_k)$. In this renormalized subspace, vertices from different graphs will be close together if their rescaled eigenvectors have similar angles and their rescaled eigenvalues are similar. Such scaling can map initially quite different dimensional vectors into similar positions in the subspace as long as their eigenspectrum components in this lower dimensional subspace share similar amounts of the matrix variance and the rescaled unit eigenvectors have similar orientations. It is easy to see that the similarities are dependent on similar truncated and rescaled eigenspectra as well as similar unit rescaled eigenvectors but they are independent of the relative sizes of the original graphs.

To be more precise, in this work, the difference between two graphs using this method was computed as

$$\|D'_G(V'_G)^T - D'_H(V'_H)^T\|^2, \quad (7)$$

where D' is the resized eigenvalue diagonal matrix while V' is the resized eigenvector matrix. The number of the k most important components are set to the size of the smaller graph.

3 JoEig: Graph Comparison in Joint Eigenspace

Unlike the previous methods, which project graphs into different eigenspaces and compare them, we project each pair of two graphs into a joint eigenspace. This joint eigenspace is expanded by both set of eigenvectors. Given Eq.(3), we can further rewrite the equations to

$$D_H = V_H^T L_H V_H, \quad D_G = V_G^T L_G V_G. \quad (8)$$

If graph G and H are isomorphic, we will have $D_G = D_H$ and therefore

$$V_G^T L_G V_G = V_H^T L_H V_H. \quad (9)$$

By multiplying V_G on the left side and V_H^T on the right side of Eq.(9), we can further derive

$$L_G V_G V_H^T = V_G V_H^T L_H. \quad (10)$$

Therefore, $V_G V_H^T$ is the joint projection vector for both graphs G and H . By introducing Eq.(3) into Eq.(10), we will have

$$V_G D_G V_G^T V_G V_H^T = V_G V_H^T V_H D_H V_H^T. \quad (11)$$

After removing $V_G^T V_G$ and $V_H^T V_H$ from Eq.(11),

$$V_G D_G V_H^T = V_G D_H V_H^T \quad (12)$$

will be obtained. Therefore, the difference between two graphs using our method can be defined as

$$\|V_G D_G V_H^T - V_G D_H V_H^T\|^2. \quad (13)$$

Unlike Umeyama’s method in which the approximation of one graph is defined by relocating the other graph’s eigenvalues in its own eigenspace, our approach approximates the graph by relocating its eigenvalues in the joint eigenspace constructed by the eigenvectors of both graphs as shown in Eq.(13).

However, the sizes of graphs might be different, and therefore it might not be possible to make a matrix product between V_G and D_H or V_H and D_G as in Eq.(13). A feasible solution is to fix the number of eigenvectors for both graphs. In this work, we study three possibilities for setting the number of eigenvectors. First, we make full use of the eigenvectors from the larger graph (by larger graph, we mean graphs with more nodes). In this case, we have to expand the eigenvectors of the smaller graph with zero vectors and also assign zero eigenvalues to them. But this will lower the influence of the smaller graph because the newly added zero eigenvalue and eigenvectors have limit impact on the results. On the other hand, we also try to make full use of the eigenvectors from the smaller graph and keep the same number of eigenvectors and eigenvalues in the larger graph as in the smaller graph by removing less important eigenvalues and eigenvectors from the larger graph. Less important eigenvectors are those with smaller eigenvalues. This, on the other hand, is in favour of the smaller graph. Moreover, we ignore the size of these two graphs and just pick a reasonable fix small number of eigenvectors and eigenvalues for both graphs. In this way, the strength of the relationship inside a graph is more important than the size of the graph, but it is not easy to choose an appropriate number of eigenvalues and eigenvectors to use.

4 Experiments

In this section, we compare the JoEig approach with other two methods, i.e. Umeyama’s and Caelli’s methods, described in Section 2 in the dissimilarity space [9]. Three classifiers, i.e., linear discriminant (ldc), quadratic discriminant (qdc) and nearest mean classifier (nmc), are adopted to have a more general understanding over the performance in the dissimilarity space. Three real-world datasets, i.e., Mutag [3], House [13] and Coil-20 [7], are used in the experiments. We use 60 representative objects to construct the dissimilarity space for the House and Coil-20 datasets and 30 representative objects for the Mutag dataset. Also, the eigenvalue diagonal and eigenvector matrices are resized in three different ways. The first is to enlarge the matrices of the smaller graph to the size of the larger graph, and it is marked as *max* in the figures. The second is by

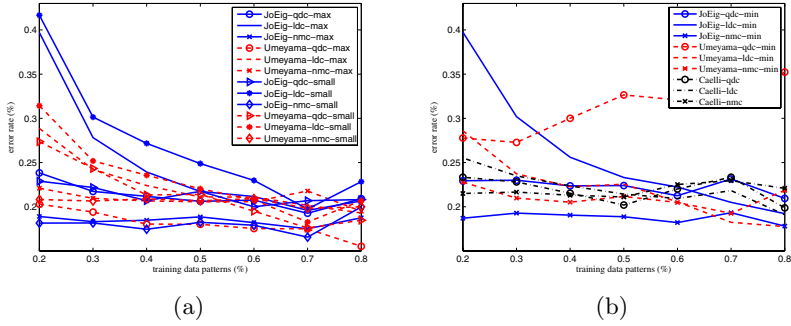


Fig. 1. Mugtag Dataset: Learning curves of (a) JoEig and Umeyama’s method with qdc, ldc, and nmc using max and small strategies and (b) all methods with qdc, ldc, and nmc using min resizing strategies

shrinking the matrices of the larger graph to the size of the smaller graph, and it is called *min*. The third one is called *small*, and it is done by reducing the matrices of both graphs to the size of the smaller graph minus 5. Moreover, all the results in the following are the average over 50 repetitions of experiments resulting in a very small standard deviation.

4.1 Experiment 1: Mutag Dataset

The Mutag dataset consists originally of 230 chemical compounds assayed for mutagenicity in *Salmonella typhimurium*. Among the 230 compounds, however, only 188 (125 positive ones, 63 negative ones) are considered to be learnable [3] and thus only 188 patterns are used in the simulations.

From Figure 1, it can be observed that under the cases of using a smaller number of eigenvalues and eigenvectors, nmc performs much better than qdc and ldc using JoEig and Umeyama’s method, especially with small sample sizes. However, by using *max* to enlarge the smaller graph, Umeyama’s method prefers qdc than nmc while nmc still performs better than qdc with JoEig. Which indicates that Umeyama’s method using the *max* strategy brings the data points into a dissimilarity space where the global distance is not good enough for separating classes and therefore requires a more complex classifier to fulfill the task.

4.2 Experiment 2: House Dataset

The house dataset contains 101, 111, and 182 image sequences from three different houses, respectively. We extract the feature points using the scale-invariant feature transform (SIFT) method [5] and then compute the Voronoi tessellations of the feature points to construct the region adjacency graph, i.e., the Delaunay triangulation, of the Voronoi regions.

As shown in Figure 2, all methods prefer ldc and qdc over nmc when the *max* strategy is used, but nmc is more preferable when the *min* strategy is adopted.

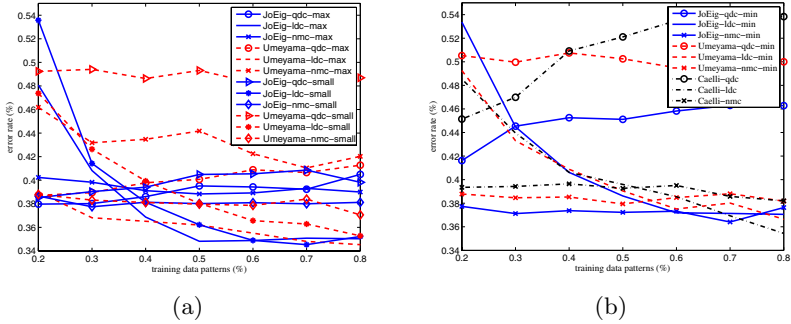


Fig. 2. House Dataset: Learning curves of (a) JoEig and Umeyama’s method with qdc, ldc, and nmc using max and small strategies and (b) all methods with qdc, ldc, and nmc using min resizing strategies

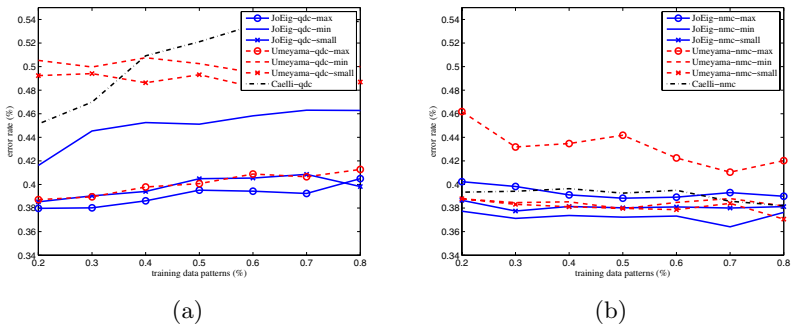


Fig. 3. House Dataset: Learning curve of (a) quadratic discriminant and (b) nearest mean classifier using JoEig, Umeyama’s, and Caelli’s methods for graph comparison

Compared to the other two datasets, the graphs in this dataset are relatively more similar and therefore their relative distances are also closer. Which makes the global distance of not so great help in separating these three classes. However, selecting a small number of eigenvalues and eigenvectors to reconstruct the graph distances in the dissimilarity space seems to enhance the distance between graphs and makes it an easier task for nmc to handle. From Figure 3, we can also observe that nmc performs better with *small* and *min* strategies and qdc is better with the *max* strategy.

4.3 Experiment 3: Coil-20 Dataset

The Coil-20 contains multiple views of the same object in different poses with respect to the camera. There are originally 20 objects (classes) in the data, but we only use 5 objects and in total 358 images to form the dataset. Graphs are derived with the same method described in Section 4.2.

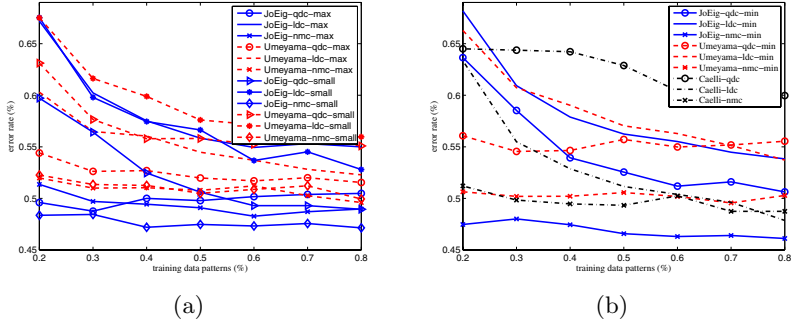


Fig. 4. Coil-20 Dataset: Learning curves of (a) JoEig and Umeyama’s method with qdc, ldc, and nmc using max and small strategies and (b) all methods with qdc, ldc, and nmc using min resizing strategies

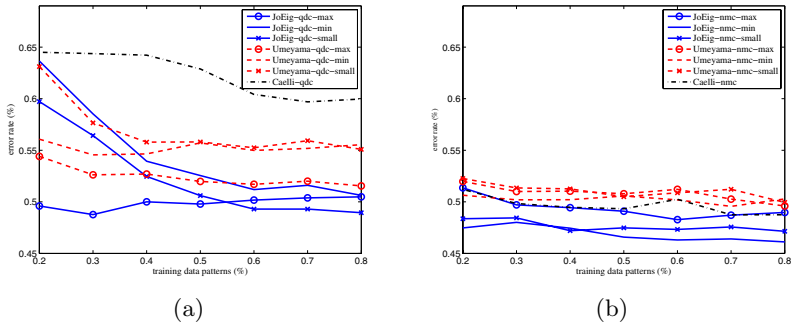


Fig. 5. Coil-20 Dataset: Learning curve of (a) quadratic discriminant and (b) nearest mean classifier using JoEig, Umeyama’s, and Caelli’s methods for graph comparison

As given in Figure 4, again we discover that nmc performs better with JoEig and Umeyama’s method using different resizing strategies despite the fact that COIL-20 dataset is with a much more complex graph structure than the Mutag dataset. From Figure 5, we derive the same conclusion as before that nmc performs better with *small* and *min* strategies while qdc is better with the *max* strategy.

5 Conclusions

We propose a graph comparison approach named JoEig in the joint eigenspace based on eigendecomposition. The main contribution of this work is introducing the joint eigenspace and comparing graphs in it. We also study three different ways for resizing the eigenvalue diagonal and eigenvector matrices to solve the inexact graph matching problem. From a series of examples presented in the experiments, we conclude that our proposed JoEig approach is in general better

than Umeyama's and Caelli's methods. Furthermore, nmc performs better with *small* and *min* strategies while qdc is better with the *max* strategy. All in all, resizing graphs with *small* or *min* strategies and then using the nmc classifier in JoEig space is probably a better solution despite the original structure of the graph dataset.

References

1. Caelli, T., Kosinov, S.: An Eigenspace Projection Clustering Method for Inexact Graph Matching. *IEEE Trans. Pattern Analysis and Machine Intelligence* 26(4), 515–519 (2004)
2. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty Years of Graph Matching in Pattern Recognition. *International Journal of Pattern Recognition and Artificial Intelligence* 18(3), 265–298 (2004)
3. Debnath, A.K., Lopez de Compadre, R.L., Debnath, G., Shusterman, A.J., Hansch, C.: Structure-Activity Relationship of Mutagenic Aromatic and Heteroaromatic Nitro Compounds. Correlation with Molecular Orbital Energies and Hydrophobicity. *Journal of Medicinal Chemistry* 34, 786–797 (1991)
4. He, P.R., Zhang, W.J., Li, Q., Wu, F.X.: A New Method for Detection of Graph Isomorphism Based on the Quadratic Form. *Journal of Mechanical Design* 125, 640–642 (2003)
5. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60(2), 91–110 (2004)
6. Umeyama, S.: An Eigendecomposition Approach to Weighted Graph Matching Problems. *IEEE Trans. Pattern Analysis and Machine Intelligence* 10(5), 695–703 (1988)
7. Nene, S.A., Nayar, S.K., Murase, H.: Columbia University Image Library (COIL-20), <http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php>
8. Neuhaus, M., Bunke, H.: Edit Distance-Based Kernel Functions for Structural Pattern Classification. *Pattern Recognition* 39, 1852–1863 (2006)
9. Pekalska, E., Duin, R.P.W.: Dissimilarity Representations Allow for Building Good Classifiers. *Pattern Recognition Letters* 23(8), 943–956 (2002)
10. Qiu, H.J., Hancock, E.R.: Spectral Simplification of Graphs. In: *Proceedings of the 8th European conference on computer vision, Czech Republic*, pp. 114–126 (2004)
11. Wilson, R.C., Hancock, E.R., Luo, B.: Pattern Vectors from Algebraic Graph Theory. *IEEE Trans. Pattern Analysis and Machine Intelligence* 27(7), 1–13 (2005)
12. Zhao, G.X., Luo, B., Ting, J., Ma, J.X.: Using Eigen-Decomposition Method for Weighted Graph Matching. In: Huang, D.-S., Heutte, L., Loog, M. (eds.) *ICIC 2007*. LNCS, vol. 4681, pp. 1283–1294. Springer, Heidelberg (2007)
13. CMU Vision and Autonomous System Center's Image Database, <http://www.ius.cs.cmu.edu/idb/>