Inexact Multisubgraph matching using Graph Eigenspace and Clustering Models *

Serhiy Kosinov and Terry Caelli

Department of Computing Science, Research Institute for Multimedia Systems (RIMS),The University of Alberta, Edmonton, Alberta, CANADA T6G 2H1

Abstract. In this paper we show how inexact multisubgraph matching can be solved using methods based on the projections of vertices (and their connections) into the eigenspaces of graphs - and associated clustering methods. Our analysis points to deficiencies of recent eigenspectra methods though demonstrates just how powerful full eigenspace methods can be for providing filters for such computationally intense problems. Also presented are some applications of the proposed method to shape matching, information retrieval and natural language processing.

Keywords: Multisubgraph matching, eigenspaces, clustering.

1 Introduction

Inexact graph matching is a fundamental task in a variety of application domains including shape matching, handwritten character recognition, natural language processing, to name a few. Naturally, there exist numerous both general and application-specific approaches for solving the problem of inexact graph matching. However, the task still presents a substantial challenge, and there still is room for improvement in some of the existing methods. Our work attempts to demonstrate the power of combining eigenspace graph decomposition models with clustering techniques to solve this problem. But before providing a detailed description of the proposed method, it is beneficial to put our work briefly into the context of previously developed solutions.

A rather generalized view point adopted by Bunke[1] poses the task of inexact graph matching as a problem of structural pattern recognition. In this work, the author has studied error-tolerant graph matching using *graph edit distance*, a concept that provides a measure of dissimilarity of two given entities and has its origins in the domain of strings. Here, a pair of graphs is compared by finding a sequence of edit operations, such as edge/vertex deletion, insertion or substitution, that transforms one graph into the other, whereas the dissimilarity, or distance, of the two graphs is said to be the minimum possible cost of such a transformation. Other important notions developed by Bunke are the weighted mean and generalized median of a pair of graphs[5], which allow a

* This project was funded by a grant from the NSERC Canada.

range of well-established techniques from statistical pattern recognition, such as clustering with self-organizing maps, to be applied in the domain of graphs. In a way similar to the work of Bunke is the effort of Tirthapura et al.[14], who successfully deployed the classical Levenshtein distance in matching shock graphs that represent 2D shapes.

Another elegant and theoretically well-grounded approach to subgraph matching is that developed by Hancock et al.[6], who, instead of going further with goal-directed search, adopt a probabilistic framework and use optimization methods to solve the graph matching problem. That is, by modelling the correspondence errors encountered during graph matching with the aid of the Bernoulli probability distribution, the authors are able to devise a graph matching likelihood function that allows one to estimate the conditional likelihood of one graph given the other and recover the best possible graph node correspondence by means of Expectation-Maximization (EM) and Singular Value Decomposition (SVD).

There also exists a whole family of graph matching techniques, generally known as spectral methods, that seek to represent and distinguish structural properties of graphs using eigenvalues and eigenvectors of graph adjacency matrices. The most valuable characteristics of such methods include being invariant to edge/vertex reordering, ability to map a graph's structural information into lower-dimensional spaces and stability under minor perturbations. On top of that, the eigendecomposition technique itself is far less computationally expensive as compared to the advanced combinatorial search procedures. Among recent developments in this field are the Umeyama's[15] formulation for same-size graph matching that derives the minimum difference permutation matrix via eigendecomposition techniques, Shapiro and Brady's[10] method for comparing graphs according to the corresponding values of the rearranged eigenvectors of graph adjacency matrices, and the work of Dickinson et al.[11] on indexing hierarchical structures with topological signature vectors obtained from the sums of adjacency matrix eigenvalues.

Similarly to the above contributions, our work borrows heavily from graph eigendecompositions. The proposed model is based upon the fundamental idea that graph matching need not be posed as a combinatorial matching problem but, rather, as one of *clustering common local relational structures* between different graphs. This results in a natural grouping between vertices of quite different graphs which share similar relational properties. We show how to do this using projection principles as used in SVD where vertex vectors from different graphs can be projected into common eigenvector subspaces.

2 Graph Eigenspace methods

2.1 Eigenspectra and eigenvectors of graphs

As mentioned above, the basic technique deployed in the majority of spectral methods is eigendecomposition. In general, for undirected graphs, it is expressed as follows:

$$A = VDV^T \tag{1}$$

where A is the square symmetric adjacency matrix of a graph, whose entry a_{ij} at the place (i,j) is equal to one if there exists an edge that connects vertex i with vertex j, and zero otherwise; V is an orthogonal matrix whose columns are normalized eigenvectors of A, and D is a diagonal matrix containing the eigenvalues λ_i of matrix A. The set of the eigenvalues found on the diagonal of matrix D is called the spectrum of A, and hence the common name for the family of methods.

One of the most well-known properties of eigendecomposition, and the one that has attracted researchers' attention for the purpose of solving inexact graph matching task in the first place, is that an eigenvalue spectrum of a matrix is invariant with respect to similarity transformations, i.e. for any non-singular matrix P, the product matrix PAP^{-1} has the same eigenvalues as A. From the view point of the graph matching problem, this means that the derived spectrum of a graph represented by its adjacency matrix is not affected by any arbitrary vertex reorderings, whose influence, or rather lack thereof, is in essence captured by the above vertex permutation matrix P.

Still, regardless of however elegant the possible graph matching problem solutions seemed at first in terms of graph eigenspectra, it was proven early on that the spectra of graphs are not unique. An obvious example that dates back to as far as 1957 was discovered by Collatz and Sinogowitz[2], and is shown in Figure 1.

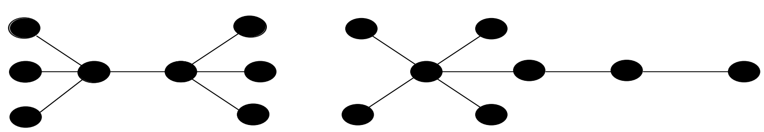


Fig. 1. two different graphs with identical eigenspectra

The above figure depicts two non-isomorphic graphs, that are nevertheless co-spectral, i.e. the sets of eigenvalues of their adjacency matrices are identical, and therefore the two graphs cannot be distinguished by relying exclusively on their spectra. Furthermore, Schwenk[9] demonstrated that as the number of vertices gets large, the probability of occurrence of a non-isomorphic co-spectral subgraph pair in any two graphs being compared asymptotically approaches unity. This means that pure spectral methods based solely on eigenvalues are generally not rich enough to fully represent graph structure variability.

Naturally, the above arguments do not add support for spectral methods. However, it is not so difficult to see that this lack of uniqueness can be easily overcome by using graph spectra together with the set of associated eigenvectors, or even by relying on the eigenvectors alone (see Equation 1). Another drawback usually attributed to the spectral methods is that they are not extendible to matching graphs of different sizes. For example, the method developed by Umeyama[15] applies only for graphs of the same size. Nevertheless, these shortcomings can be eliminated by applying normalization and projection operations - the topic of the following section.

2.2 Normalizations and Projections

Subspace projection methods, in the principal component analysis (PCA) literature, are conventionally used to reduce the dimensionality of data, while minimizing the information loss due to the decreased number of dimensions. It is performed in the following way. The dataset covariance matrix Σ is first decomposed into the familiar eigenvalue/eigenvector matrix product (see Eq. 1):

$$\Sigma = U \Lambda U^T \tag{2}$$

where U is a matrix of eigenvectors ("principal components" of the data), and Λ is a diagonal matrix of eigenvalues. The original data is then projected onto a smaller number of the most important (i.e., associated with the largest eigenvalues) principal components as specified in the below equation (and thus, the data's dimensionality is reduced):

$$\hat{x} = U_k^T x \tag{3}$$

Here, $\hat{x}$ is the computed projection, U_k^T is the matrix of k principal components in a transposed form, and x is an item from the original data.

Taking the very same approach, we can project vertex connectivity data from a graph adjacency matrix onto a smaller set of its most important eigenvectors. The projection coordinates obtained in this way would then represent the relational properties of individual vertices relative to the others in the lower-dimensional eigenspace of a given graph. In this eigenvector subspace, structurally similar vertices or vertex groups would be located close to each other, which can be utilized for approximate comparison and matching of graphs.

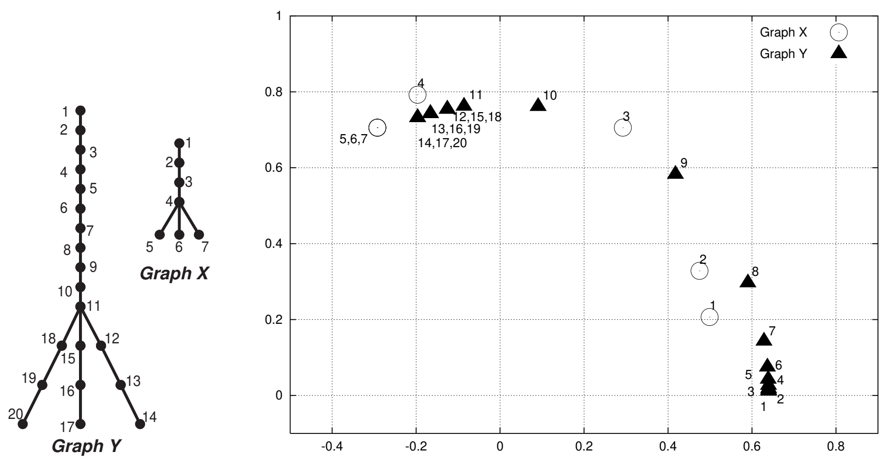
However, in order to be able to use the outlined above projection method for graph matching, it is necessary to resolve the following issues: first, how many dimensions to choose for vertex eigenspace projections? Second, how to ensure the comparability of the derived projections for graphs with a different number of vertices?

The first is answered by the relative sizes of the eigenvalues associated with each dimension or eigenvector with non-zero eigenvalue signalling the redundancy of the associated subspaces. That is, for a given pair of graphs one should choose the k most important eigenvectors as the projection components, where k is the smaller value of the ranks of adjacency matrices of the two graphs being compared[1] , i.e. $k = min(rank(A_{Graph_1}), rank(A_{Graph_2}))$.

As for the second question, the empirical evidence suggests that an extra step of renormalization of the projections may suffice. Here, the idea is that for

[1] However, in order to make the following examples more illustrative, without a loss of generality in the further discussion we will use only 2-dimensional projections, which can be easily depicted in the 2D plane.

the purpose of comparing two arbitrary graphs we need not consider the values of the projections as such, but instead should look at how they are positioned and oriented *relative to each other* in their eigenvector subspace. That is, if projections are themselves viewed as vectors, we disregard their magnitudes, while only paying attention to their direction and orientation. And this is exactly what projection coordinate renormalization helps us to do: in the end all of the projections are unit-length vectors that can only be distinguished by their orienation, and not by their length. In addition to that, we also carry out a dominant sign correction of the projection coordinates of either of the two graphs being matched so as to align one set of graph vertex projections against the other. This corresponds to setting the direction of the axes in such a way to result in the most compatible alignment between the vertex data using the dominant sign test.



(a) Graphs X, Y.

(b) Projections of graphs X and Y into 2D eigenvector subspace.

Fig. 2. Example 1: graphs and their projections

In order to provide an illustration for the described above propositions, let us consider an example with two graphs X and Y depicted in Figure 2(a). Although different in size, the two graphs are nevertheless quite similar to each other. In fact, one may see graph Y as an enlarged version of graph X. The result of projecting the two graphs into the normalized 2D eigenvector subspace shown in Figure 2(b) demonstrates the following two important features of the proposed method: firstly, the projections of vertices of both graphs follow a similar pattern, which means that it is possible to determine overall structural similarity of graphs with different number of vertices, and secondly, one may also see (by examining the juxtaposition of the projected vertices of both graphs) that graph vertices with similar relational properties tend to get projected into the areas that are close to each other. These properties are quite valuable, and, as such, have the potential to prove useful in solving the graph matching problem. The latter conjecture is confirmed by the experimental results which show that an overall graph similarity can be estimated by comparing the vertex projection distributions with the aid of multi-dimensional extension of Kolmogorov-Smirnov (K-S) statistical test. However, the K-S test becomes a rather computationally expensive procedure if applied to high-dimensional data. Also, it does not help us much to resolve another important issue of the graph matching problem, namely, the one of recovering structurally similar vertex correspondence in a pair of graphs being compared. To this end, we use clustering methods - as follows.

2.3 Clustering in Graph Eigenspaces and inexact solutions to subgraph matching

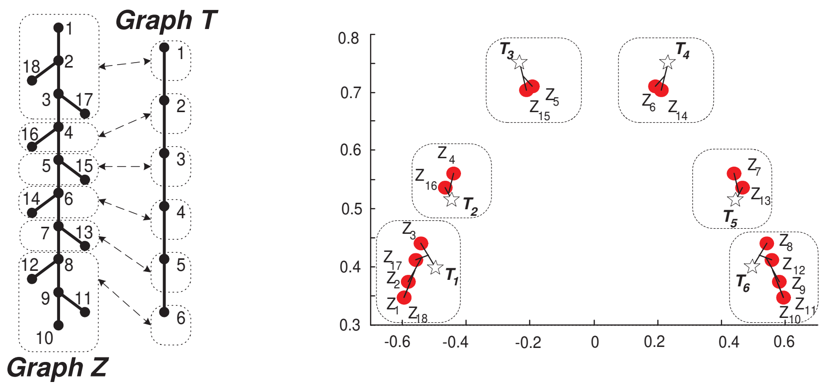
This eigenvector subspace method allows us to determine the overall similarity of a pair of graphs by the positioning of the vertex projections of both graphs relative to each other. The only remaining step for solving the graph matching problem is to find the correspondence among the vertices that have similar relational properties. The main advantage of using clustering to solve this problem is that it can equally well discover correspondence relationships of various types, i.e. it is not limited to finding the best one-to-one matches of vertices from one graph to the other, but it can also identify the whole sub-graphs and vertex groups that possess similar structural properties.[2] In order to realize this, we deploy a standard agglomerative clustering routine with only two necessary modifications: first, the algorithm gives a higher priority for clustering the candidate vertex projections that belong to different graphs, rather than the same one; second, the clustering procedure stops as soon as all of the vertex projections have been associated with a certain cluster. Once the clustering is completed, a simple customized cluster validity index that takes into account the number of obtained clusters and their quality based on the Dice[3] coefficient formula[3] is used to measure the similarity (or distance) of a pair of graphs. Figure 3 illustrates the result of vertex projection clustering (Figure 3(b)) of two sample graphs Z and T with 18 and 6 vertices respectively, that recovers a natural correspondence among the groups of vertices in these two graphs (shown in Figure 3(a)).

3 Application

For the purpose of initial testing the proposed graph matching method, two application areas were chosen: first being the matching of shapes represented by

[2] This quiality can be very important when the two graphs have substantially different number of vertices.

[3] analogous to the well-known "intersection-over-union" measure of set similarity.



(a) Graphs Z, T.

(b) Clustered projections of graphs Z and T (the formed clusters of vertex correspondence are circled).

Fig. 3. Example 2: Clustering of vertex projections of sample graphs Z and T.

shock trees, and second - information retrieval with sentence parse tree analysis. In the first application area, shock tree matching (described in detail in [13, 4,

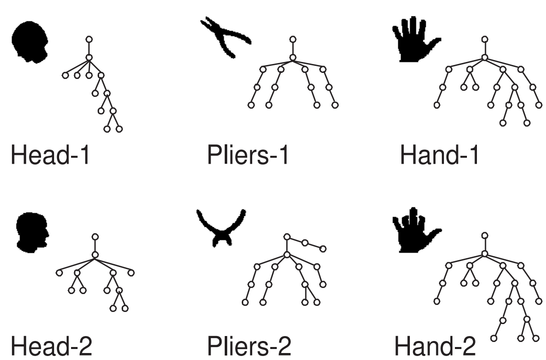


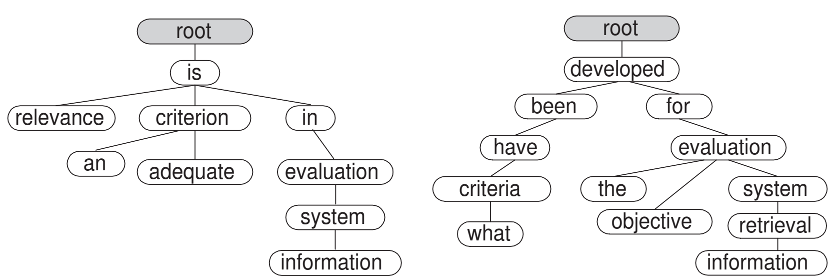
Fig. 4. The subset of shapes and their shock graph representations, from [8]

12]), a small set of shapes documented in [8] was used. The graphical representation of the dataset and the similarity matrix for the tested shapes, as calculated according to the above mentioned cluster validity index for measuring the similarity among the clustered eigenvector subspace projections, are shown in Figure 4 and Table 1 respectively (where the best matching shape similarity values are in bold font).

Table 1. The similarity matrix obtained for the subset of shapes shown in Figure 4.

	Head-1	**Head-2**	**Pliers-1**	**Pliers-2**	**Hand-1**	**Hand-2**
Head-1		**0.5536**	0.1936	0.4392	0.2000	0.1280
Head-2	**0.5536**		0.2373	0.3978	0.3133	0.1270
Pliers-1	0.1936	0.2373		**0.4857**	0.2087	0.2006
Pliers-2	0.4392	0.3978	**0.4857**		0.2126	0.1612
Hand-1	0.2000	0.3133	0.2087	0.2126		**0.3777**
Hand-2	0.1280	0.1270	0.2006	0.1612	**0.3777**	

In the second application area, a subset of queries and documents from ADI[16] text collection was parsed into a group of dependency trees. Subsequently, a standard keyword-based information retrieval system[7] was modified so as, on one hand, to restrict the keyword matching process only to words that have similar structural properties in both query and document sentence dependency trees, and, on the other hand, to allow for more flexibility in individual word comparisons by letting a direct within-cluster part of speech correspondence count as a partial match. As a result, the overall performance indicators improved, which can be illustrated by the following example.



(a) Document sentence parse tree. (b) Query sentence parse tree.

Fig. 5. Parse trees of sample sentences from document 27 and query 13.

Both query 13 and document 27 in the ADI text collection have a substantial keyword overlap[4] , however, a conventional keyword-based information retrieval system does not recognize this pair as the best match. Instead such a system ranks high some other "relevant" documents which share a lot of keywords with the query, even though these common keywords are quite inappropriate if one

[4] The sample sentences considered are: *"What criteria have been developed for the objective evaluation of information retrieval and dissemination systems?"*, and *"Is relevance an adequate criterion in retrieval system evaluation?"*.

considers their context conveyed by the sentence syntactic structure. The use of the proposed eigenvector subspace projection method allowed us to take into account the parse tree structure in addition to the keyword information, which lead to improved results. The parse trees (after conjunction expansion and prepositional post-modifier normalization) of the sample sentences from the above document and query are depicted in Figure 5; their projections, that were used to estimate syntactic structural similarity of individual keywords, are shown in Figure 6.

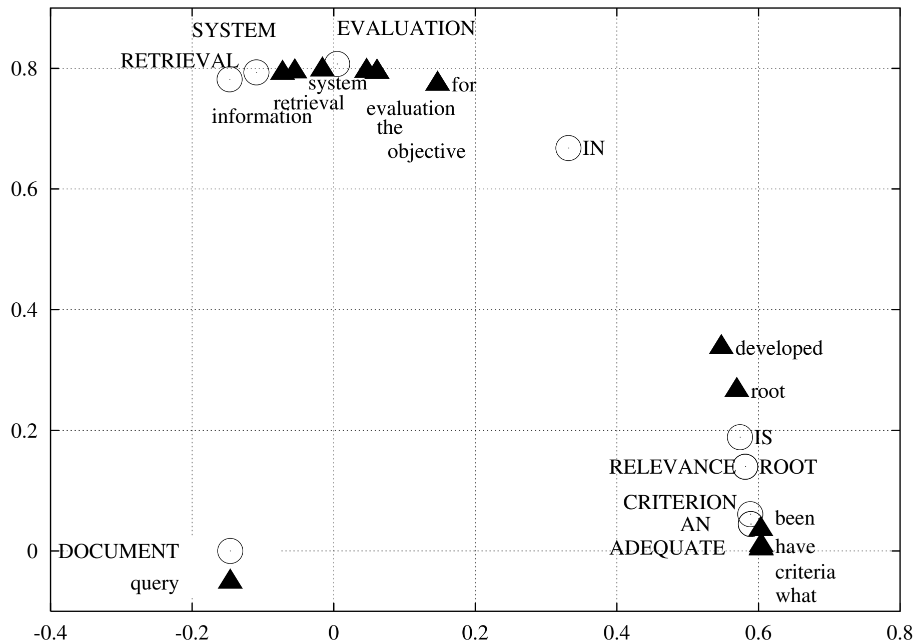


Fig. 6. Comparison of two sentence parse tree projections: an application in natural language processing.

4 Conclusion

In this paper, we have described an approach for inexact multisubgraph matching using the technique of projection of graph vertices into the eigenspaces of graphs in conjunction with standard clustering methods. The two most important properties of the proposed approach are, first, its ability to match graphs of considerably different sizes, and, second, its power to discover correspondence relationships among subgraphs and groups of vertices, in addition to the "one-to-one" type of vertex correspondence that the majority of previously developed solutions of the graph matching problem mostly focused on. In addition to that, we have also explored two potential areas for practical application for the described approach - matching of shapes represented by shock trees and natural language processing, and obtained results encouraging further research of the method.

References

1. H. Bunke. Recent advances in structural pattern recognition with application to visual form analysis. *IWVF4, LNCS*, 2059:11–23, 2001.
2. L. Collatz and U. Sinogowitz. Spektren endlicher grafen. *Abh. Math. Sem. Univ. Hamburg*, 21:63–77, 1957.
3. L. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26:297–302, 1945.
4. P. Dimitrov, C. Phillips, and K. Siddiqi. Robust and efficient skeletal graphs. *Conference on Computer Vision and Pattern Recognition*, june 2000.
5. X. Jiang, A. Munger, and H. Bunke. On median graphs: properties, algorithms, and applications. *IEEE Trans. PAMI*, 23(10):1144–1151, October 2001.
6. B. Luo and E. Hancock. Structural graph matching using the em algorithm and singular value decomposition. *IEEE Trans. PAMI*, 23(10):1120–1136, October 2001.
7. N. Maloy. Successor variety stemming: variations on a theme. 2000. project report (unpublished).
8. M. Pelillo, K. Siddiqi, and S. Zucker. Matching hierarchical structures using association graphs. *IEEE Trans. PAMI*, 21(11), November 1999.
9. A. Schwenk. *Almost all trees are cospectral.* Academic Press, New York - London, 1973.
10. L. Shapiro and J. Brady. Feature-based correspondence - an eigenvector approach. *Image and Vision Computing*, 10:268–281, 1992.
11. A. Shokoufandeh and S. Dickinson. A unified framework for indexing matching hierarchical shape structures. *IWVF4, LNCS*, 2059:67–84, 2001.
12. K. Siddiqi, S. Bouix, A. Tannebaum, and S. Zucker. Hamilton-jacobi skeletons. *To appear in International Journal of Computer Vision.*
13. K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 30:1–24, 1999.
14. S. Tirthapura, D. Sharvit, P. Klein, and B. Kimia. Indexing based on edit-distance matching of shape graphs. *Multimedia Storage and Archiving Systems III*, 3527(2):25–36, 1998.
15. S. Umeyama. An eigen decomposition approach to weighted graph matching problems. *IEEE Trans. PAMI*, 10:695–703, 1998.
16. URL. Adi, cisi, med test collections.
ftp://ftp.cs.cornell.edu/pub/smart/.