Selecting Structural Base Classifiers for Graph-Based Multiple Classifier Systems*

Wan-Jui Lee[1], Robert P.W. Duin[1], and Horst Bunke[2]

[1] Pattern Recognition Laboratory,
Delft University of Technology, The Netherlands
W.J.Lee@tudelft.nl, r.duin@ieee.org
[2] Institute of Computer Science and Applied Mathematics,
University of Bern, Switzerland
bunke@iam.unibe.ch

Abstract. Selecting a set of good and diverse base classifiers is essential for building multiple classifier systems. However, almost all commonly used procedures for selecting such base classifiers cannot be directly applied to select structural base classifiers. The main reason is that structural data cannot be represented in a vector space.

For graph-based multiple classifier systems, only using subgraphs for building structural base classifiers has been considered so far. However, in theory, a full graph preserves more information than its subgraphs. Therefore, in this work, we propose a different procedure which can transform a labelled graph into a new set of unlabelled graphs and preserve all the linkages at the same time. By embedding the label information into edges, we can further ignore the labels. By assigning weights to the edges according to the labels of their linked nodes, the strengths of the connections are altered, but the topology of the graph as a whole is preserved.

Since it is very difficult to embed graphs into a vector space, graphs are usually classified based on pairwise graph distances. We adopt the dissimilarity representation and build the structural base classifiers based on labels in the dissimilarity space. By combining these structural base classifiers, we can solve the labelled graph classification problem with a multiple classifier system. The performance of using the subgraphs and full graphs to build multiple classifier systems is compared in a number of experiments.

1 Introduction

A multiple classifier system [6] is based on the idea to combine several classifiers such that the combined system achieves better performance than the individual ones. The base classifiers to be combined are required to be sufficiently diverse [5,6]. Data resampling and feature subset selection [5] are two common ways for

* We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250).

N. El Gayar, J. Kittler, and F. Roli (Eds.): MCS 2010, LNCS 5997, pp. 155–164, 2010.

promoting the diversity of base classifiers. Data resampling methods, e.g. bagging and boosting [15], select different training samples for different base classifiers. In feature subset selection, base classifiers are trained using different subsets of features and their solutions are usually very different. Therefore, feature subset selection methods could yield better diversity than data resampling methods. There is another reason besides diversity for feature subset selection, that is, the curse of dimensionality problem. For a dataset with very high dimensionality, it is possible that one single classifier could not find a good solution in this high dimensional vector space. By feature subset selection, the problem can be solved in lower dimensional spaces and there is a higher chance to find better solutions.

However, for structural pattern recognition problems, patterns are not represented with only numerical features and there is also no direct way to embed graphs into a vector space, especially for the structural relationships within one object. Because the space of structural data, e.g., strings, trees or graphs, has not been properly vectorized yet, there are just a few attempts for building multiple classifier systems based on structural representations [1,12,2,8]. Obviously, feature subset selection methods are not applicable to train base classifiers for structural patterns as most other methods developed for statistical pattern recognition problems. But then the question arises what is a good alternative for increasing the diversity and maybe also avoiding the problem of high dimensionality for structural multiple classifier systems?

One of the few examples for creating structural base classifiers is discussed in [14]. The idea is to generate different graph-based classifiers by randomly removing nodes and their incident edges from the training graphs until a maximum number of nodes is reached for all graphs. Because of the randomness, different graph-based classifiers can be created and each becomes a base classifier in the multiple classifier system. However, with this setting, we still need to compute similarity/dissimilarity for labelled graphs using time-consuming techniques such as the maximum common subgraph [2] or the graph edit distance [9] considering a labelled graph classification problem. Unlike graphs with unlabelled nodes, graphs with labelled nodes usually need to be processed and described with more complicated algorithms and structures. Also, classifying graphs with labelled nodes is a more difficult task than classifying graphs with unlabelled nodes. Therefore, a method was proposed in [8] to decompose labelled graphs into sets of unlabelled subgraphs based on label information, and compare the dissimilarity between all pairs of subgraphs in order to create base classifiers in the dissimilarity space [10] for different labels.

Both existing methods described above for building structural multiple classifier systems only consider subgraphs for training base classifiers. One is selecting subgraphs randomly and the other is selecting subgraphs based on label information. Does this mean that subgraph selection is the best way for increasing the diversity of the base classifiers? Does subgraph selection somehow resemble feature subset selection?

In [8], we observed a very interesting phenomenon that all the combiners reaching the lowest error rate have at least one of the global structure base

classifiers as the base classifiers. So it is clear that the global structures can improve the classification performance. The global structure means that the linkages of the given graph are fully preserved. Therefore instead of subgraphs, the full graphs are used for creating structural base classifiers. This suggests that full graphs are beneficial to the multiple classifier system. Therefore, instead of selecting subgraphs for training base classifiers, we propose a method to alter the full graphs and train base classifiers based on different versions of altered full graphs. The goal for this work is to investigate the best way for building diverse structural base classifiers, i.e., whether we should select the subgraphs, alter the full graphs or adopt both.

To derive different full graphs from the same graph and transfer a labelled graph into an unlabelled one, the label information is utilized by us to alter the graph into different forms. The alteration is done by assigning weights to the edges and the label information is also embedded in these weights. We can further ignore the labels on the nodes once the label information is embedded on the edges.

The rest of the paper is organized as follows. A multiple classifier system utilizes the label information of graphs for altering full graphs in order to build structural base classifiers is proposed in Section 2. In Section 3, we recap the JoEig approach for comparing unlabelled graphs. Simulation results are presented in Section 4. Finally, a conclusion is given in Section 5.

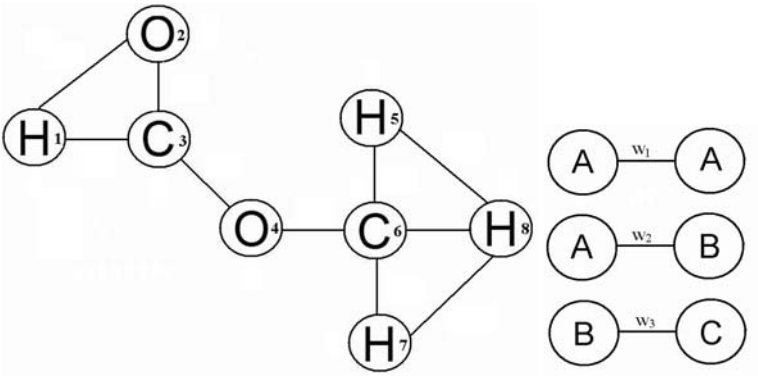
2 Building a Multiple Classifier System Using Altered Full Graphs

Before we introduce the altered full graphs for building structural base classifiers for a multiple classifier system, some definitions and an 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 $V = \{v_1, v_2, \ldots, v_n\}$, the edge set $E = \{e_1, e_2, \ldots, e_m\} \subset V \times V$, 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

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

Clearly since the graph is undirected, the matrix A is symmetric. The Laplacian 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. We use the example graph shown in Figure 1(a) through this section to explain our method. This example graph is with 8 nodes and each node is labelled with one symbol. There are no attributes on the edges and the elements of the adjacency matrix A given in Eq.(2) of this



(a) (b)

Fig. 1. Examples of (a) a labelled graph; (b) possible linkage combinations in graphs according to label A

graph are either 1 or 0 to indicate whether there is an edge between two nodes or not.

$$A = \begin{pmatrix} 0\,1\,1\,0\,0\,0\,0\,0 \\ 1\,0\,1\,0\,0\,0\,0\,0 \\ 1\,1\,0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0\,1 \\ 0\,0\,0\,1\,1\,0\,1\,1 \\ 0\,0\,0\,0\,0\,1\,0\,1 \\ 0\,0\,0\,0\,1\,1\,1\,0 \end{pmatrix}. \tag{2}$$

2.1 Full Graph Alterations

Our goal is to solve the labelled graph classification problem by altering a labelled graph into a set of unlabelled graphs that preserve all the linkage structures. The alteration is given by assigning weights($\neq 0$) to the edges. In order to assign the weights in such a way that the new set of altered full graphs are diverse, the weights are given according to the node label information. For instance, if there are three different labels, i.e., A, B and C, in a graph, we can define three kinds of connection strengths according to label A as shown in Figure 1(b). Suppose label A is considered as the master label, the edge connects two nodes with both master label A will have the weight w_1. The edge connects one node with master label A and the other node which is not A will have the weight w_2, and finally the edge that connects two nodes that are both not master label will have the weight w_3. Without loss of generality, we assume that $1 \geq w_1 \geq w_2 \geq w_3 > 0$. By selecting different labels as the master label, the strengths of the connections

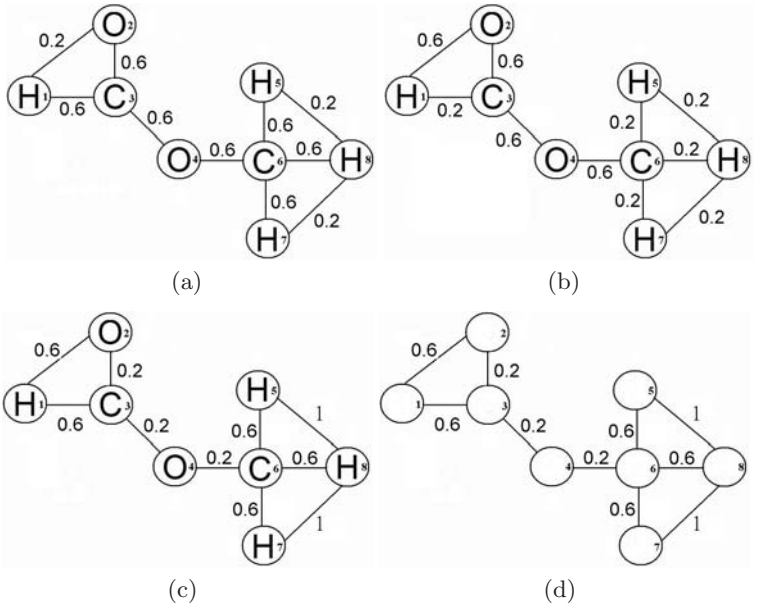


Fig. 2. Examples of altered full graphs with weights assigned based on label (a) C, (b) O and (c) H, respectively, from the graph in Figure 1(a), and (d) the unlabelled version of altered full graph based on label H

in a graph will change but the linkage structure will remain the same because the weights can not be equal to zero by assumption.

For the example in Figure 1(a), there are three different labels, i.e., C, H and O. For each label, we will assign weights to the edges according to this particular label. Let $w_1 = 1$, $w_2 = 0.6$ and $w_3 = 0.2$, Figure 2(a), Figure 2(b) and Figure 2(c) are with weights assigned according to label C, O, and H, respectively. Now that the label information is embedded to the edges with different weights, it means that we can ignore the label within the graph as in Figure 2(d) and fully describe this graph with a connection matrix (which is composed of the weights of the edges). For the example graph in Figure 2(a), its connection matrix A_C will be

$$A_C = \begin{pmatrix} 0 & 0.2 & 0.6 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0.6 & 0 & 0 & 0 & 0 & 0 \\ 0.6 & 0.6 & 0 & 0.6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.6 & 0 & 0 & 0.6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.6 & 0 & 0.2 \\ 0 & 0 & 0 & 0.6 & 0.6 & 0 & 0.6 & 0.6 \\ 0 & 0 & 0 & 0 & 0 & 0.6 & 0 & 0.2 \\ 0 & 0 & 0 & 0 & 0.2 & 0.6 & 0.2 & 0 \end{pmatrix}. \tag{3}$$

Notice that if w_1, w_2 an w_3 are all set to 1, the connection matrix A_C equals the adjacency matrix.

2.2 Dissimilarity and Base Classifiers

Given m graphs with n distinctive labels among the graphs, we want to create n base classifiers with respect to the labels. So, for a certain label, we derive an altered full graph and its connection matrix from each graph by assigning different weights as described above. With these m altered full graphs, the dissimilarities are calculated pairwise with the JoEig approach as described in the next Section. As a result, we can obtain an $m \times m$ dissimilarity matrix for each label. With this dissimilarity matrix, we can build a base classifier for this label in the dissimilarity space [10]. In the end, we can construct n label base classifiers by doing the same to each label. Dissimilarity space uses (selected) object dissimilarities as axes and objects as points. That is, axis 1 is the dissimilarity to object 1, axis 2 the dissimilarity to object 2 and so on. Object points are located in this space by their dissimilarities to each (selected) objects. These selected objects are also called the representation set. With this setting, we can project the objects into a vector space and build a classifier in it.

3 JoEig: Graph Comparison in Joint Eigenspace

JoEig [7] projects each pair of two graphs into a joint eigenspace. This joint eigenspace is expanded by both sets of eigenvectors.

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 $L_G = V_G D_G V_G^T$ and $L_H = V_H D_H V_H^T$ 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. With the joint projection vector $V_G V_H^T$, both graphs G and H will be projected to their joint eigenspace as $L_G V_G V_H^T$ and $V_G V_H^T L_H$. The difference between two graphs using JoEig is defined as $\|V_G D_G V_H^T - V_G D_H V_H^T\|^2$. The JoEig approach approximates a graph by relocating its eigenvalues in the joint eigenspace constructed by the eigenvectors of both graphs.

There are also three possibilities for setting the number of eigenvectors to compare graphs with different sizes in JoEig. In this work, we choose 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.

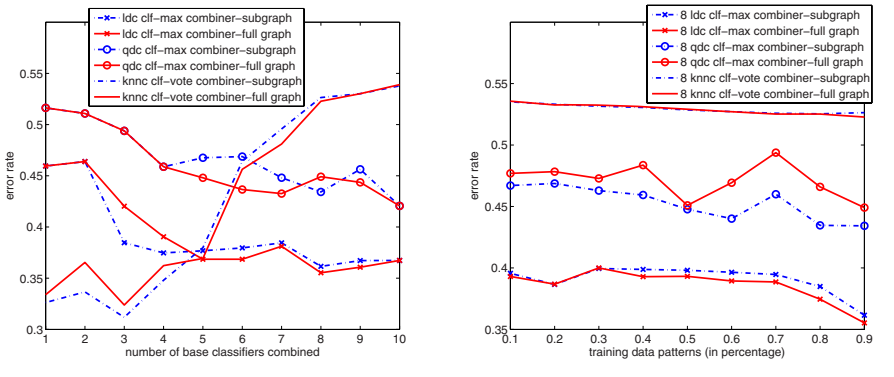
4 Experiments

In this section, we compare the performance of the multiple classifier systems built on the subgraphs and the altered full graphs, respectively. Linear discriminant classifier (ldc), quadratic discriminant classifier (qdc) and k-nearest neighbor classifier (knnc) are adopted to build base classifiers in the dissimilarity space [10], respectively. For knnc, the 3 nearest neighbors are considered. All the base

classifiers and the classifier combiner are built with the PRTOOLS [4]. Two real-world datasets, i.e., Mutagenicity and AIDS [13], are used in the experiments where 60% of objects are randomly selected and used as the training and testing datasets, 20% are used as the validation set for indicating the performance of individual base classifiers, and the other 20% are used as the other validation set for searching the best values for weights, i.e., w_1,w_2, and w_3. We randomly select 15% of training objects and use them as the representative objects to construct the dissimilarity space for both datasets. Also, the eigenvalue diagonal and eigenvector matrices are resized to the size of the smaller graph with the JoEig approach. 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: Mutagenicity Dataset

Mutagenicity is one of the numerous adverse properties of a compound that hampers its potential to become a marketable drug. The molecules are converted into graphs in a straightforward manner by representing atoms as nodes and the covalent bonds as edges. Nodes are labeled with the corresponding chemical symbol, and there are 10 different symbols in total. The average number of nodes of a graph is 30.3 ± 20.1, and the average number of edges is 30.7 ± 16.8. The Mutagenicity dataset is divided into two classes, i.e., mutagen and nonmutagen. There are in total 4,337 elements (2,401 mutagen elements and 1,936 nonmutagen elements). In the experiments, 50% of objects are randomly selected and used as the training dataset. In Figure 3 (a), we add the base classifiers (10 base classifiers from subgraphs or altered full graphs) one by one. At each step, the base classifier performing best against the validation set will be selected as the next base classifier to be added. The max combination rule is used for ldc



(a) (b)

Fig. 3. Compare subgraphs and full graphs for building base classifiers with (a) combination results of different number of base classifiers and (b) learning curves of combining best 8 base classifiers for Mutagenicity dataset

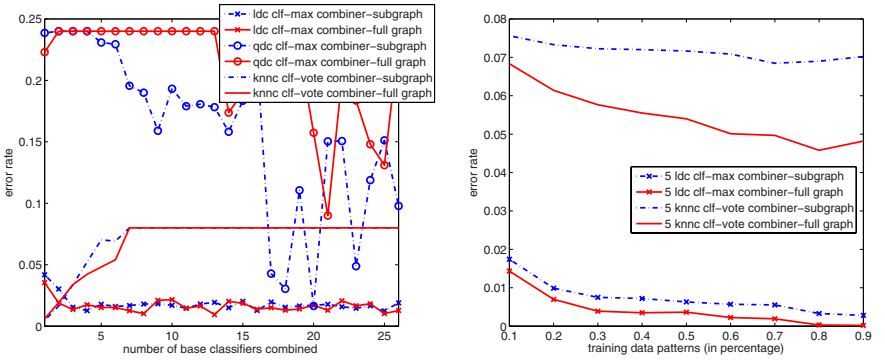
and qdc while the voting combination rule is used for knnc. For chemical compounds, atoms 'C' and 'H' are very common elements among and within objects. Especially for atom 'C', the full graphs or subgraphs constructed based on label 'C' can preserve most structures of the original graphs and therefore the base classifier constructed on label 'C' usually has the best individual performance. At first, knnc has base classifiers that are significantly better than ldc and qdc, which means the data distribution is rather nonlinear and knnc is more suitable for such a problem. But knnc is easily over-trained and also has unreliable confidence and therefore its performance decreases dramatically when more and more base classifiers are combined. If the max combination rule is used for knnc instead of voting, the performance of knnc will decrease even faster. Nevertheless, combining the best 3 individual knnc base classifiers can reach the optimal performance which is much better than all the combination results of ldc and qdc. Therefore, it might be beneficial to use knnc as base classifiers but selecting the best set of base classifiers is a crucial problem.

From Figure 3(a), we can see that base classifiers built on full graphs give better combination results than base classifiers built on subgraphs when more base classifiers are combined. However, with a few base classifiers, subgraphs perform better than full graphs with ldc and knnc. This means full graphs give more information about the structure of the original graph when more different labels are considered. On the other hand, the base classifiers built on subgraphs are more diverse in the beginning. We can also observe from Figure 3(a) that for ldc and qdc, the performance increases when there are more and more base classifiers combined. Because adding base classifiers is like adding features, when there is a sufficient number of objects, combining different base classifiers yields higher possibilities of having better performance than individual classifiers.

To fairly investigate the limitations and capabilities of subgraphs and full graphs, the learning curves of combining the best 8 base classifiers for both methods are drawn in Figure 3(b). Clearly, we can see that the subgraphs work better with small sample sizes and the full graphs on the other hand are better with medium and large sample sizes for ldc. Since graphs are usually with complex structures, it is possible to overfit when the number of objects is not sufficiently large. With subgraph selection, the structures are decomposed into simpler format and this problem might be avoided. In Figure 3(a), we can see that when the number of base classifiers is 8, using full graphs for qdc is worse than using subgraphs and therefore, we can also expect the same from Figure 3(b).

4.2 Experiment 2: AIDS Dataset

The AIDS dataset consists of graphs representing molecular compounds. The graphs are constructed from the AIDS Antiviral Screen Database of Active Compounds (molecules). This dataset consists of two classes, active and inactive, to indicate molecules with activity against HIV or not. The molecules are converted into graphs in a straightforward manner by representing atoms as nodes and the covalent bonds as edges. Nodes are labeled with the corresponding chemical symbol, and there are 26 labels in total. The average number of nodes of



(a) (b)

Fig. 4. Compare subgraphs and full graphs for building base classifiers with (a) combination results of different number of base classifiers and (b) learning curves of combining best 5 base classifiers for AIDS dataset

a graph is 15.6 ± 13.1, and the average number of edges is 16.1 ± 15.0. There are 2,000 elements in total (1,600 inactive elements and 400 active elements). In the experiments, 50% of objects are randomly selected and used as the training dataset.

In Figure 4(a), the base classifiers (26 base classifiers from subgraphs or altered full graphs) are added one by one using the same technique as described above. The AIDS dataset is a much easier dataset to classify compared to the Mutagenicity dataset, and the best individual ldc classifiers built on subgraphs and full graphs, respectively, already reach very small error rates which makes it difficult for the combiner to improve the individual performance. There is not much difference for both methods in combining different numbers of ldc classifiers. The qdc and knnc classifiers preform significantly much worse than the ldc classifier with this dataset. The learning curves of combining the best 5 base classifiers for both methods are drawn in Figure 4(b). We can see that the full graphs perform better than the subgraphs with a larger number of objects for knnc but the difference is rather small for ldc.

5 Discussions and Conclusions

We solve the labelled graph classification problem with the multiple classifier system by decomposing labelled graphs into unlabelled full graphs based on their labels and building base classifiers from the full graphs. The full graphs preserve the topology from the original graph and therefore carry more information than subgraphs. Therefore using full graphs is beneficial when there is a sufficient number of objects. On the other hand, because of the complex structure of graphs, it is possible to encounter the problem of high dimensionality. Adopting subgraphs is a better solution in this case.

For highly nonlinear problems, knnc is probably a good solution and it is actually commonly adopted in graph classification problems. Therefore, how to select a proper set of knnc classifiers to combine for graph classification problems could be a direction for future study.

References

1. Bunke, H., Irniger, C., Neuhaus, M.: Graph Matching - Challenges and Potential Solutions. In: Roli, F., Vitulano, S. (eds.) ICIAP 2005. LNCS, vol. 3617, pp. 1–10. Springer, Heidelberg (2005)
2. Bunke, H., Riesen, K.: Graph Classification Based on Dissimilarity Space Embedding. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) SSSPR 2008. LNCS, vol. 5342, pp. 996–1007. Springer, Heidelberg (2008)
3. Dijkstra, E.W.: A Note on Two Problems in Connexion with Graphs. Numerische Mathematik 1, 269–271 (1959)
4. Duin, R.P.W., Juszczak, P., Paclik, P., Pękalska, E., de Ridder, D., Tax, D.M.J.: PRTOOLS 2004, A Matlab Toolbox for Pattern Recognition. Delft University of Technology, ICT Group, The Netherlands (2004), `http://www.prtools.org`
5. Ho, T.K.: The Random Subspace Method for Constructing Decision Forests. IEEE Trans. Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
6. Kuncheva, L.I.: Combining Pattern Classifiers. In: Methods and Algorithms. Wiley, Chichester (2004)
7. Lee, W.J., Duin, R.P.W.: An Inexact Graph Comparison Approach in Joint Eigenspace. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) SSSPR 2008. LNCS, vol. 5342, pp. 35–44. Springer, Heidelberg (2008)
8. Lee, W.J., Duin, R.P.W.: A Labelled Graph Based Multiple Classifier System. In: Benediktsson, J.A., Kittler, J., Roli, F. (eds.) MCS 2009. LNCS, vol. 5519, pp. 201–210. Springer, Heidelberg (2009)
9. Neuhaus, M., Bunke, H.: Edit Distance-Based Kernel Functions for Structural Pattern Classification. Pattern Recognition 39, 1852–1863 (2006)
10. Pękalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition. In: Fundations and Applications. World Scientific, Singapore (2005)
11. Qiu, H.J., Hancock, E.R.: Spectral Simplication of Graphs. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 114–126. Springer, Heidelberg (2004)
12. Riesen, K., Bunke, H.: Classifier Ensembles for Vector Space Embedding of Graphs. In: Haindl, M., Kittler, J., Roli, F. (eds.) MCS 2007. LNCS, vol. 4472, pp. 220–230. Springer, Heidelberg (2007)
13. Riesen, K., Bunke, H.: IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) SSSPR 2008. LNCS, vol. 5342, pp. 287–297. Springer, Heidelberg (2008)
14. Schenker, A., Bunke, H., Last, M., Kandel, A.: Building Graph-Based Classifier Ensembles by Random Node Selection. In: Roli, F., Kittler, J., Windeatt, T. (eds.) MCS 2004. LNCS, vol. 3077, pp. 214–222. Springer, Heidelberg (2004)
15. Skurichina, M., Kuncheva, L.I., Duin, R.P.W.: Bagging and Boosting for the Nearest Mean Classifier: Effects of Sample Sizes on Diversity an Accuracy. In: Roli, F., Kittler, J. (eds.) MCS 2002. LNCS, vol. 2364, pp. 62–71. Springer, Heidelberg (2002)