
Article:
Bahonar, Hoda, Mirzaei, Abdolreza and Wilson, Richard Charles orcid.org/0000-0001-7265-3033 (2017) Diffusion Wavelet Embedding: a Multi-resolution Approach for Graph Embedding in Vector Space. Pattern Recognition. pp. 1-37. ISSN 0031-3203

https://doi.org/10.1016/j.patcog.2017.09.030

Diffusion Wavelet Embedding: a Multi-resolution Approach for Graph Embedding in Vector Space

Hoda Bahonar[a,*], Abdolreza Mirzaei[a], Richard C. Wilson[b]

[a] *Dept. of Electrical and Computer Engineering, Isfahan University of Technology, Iran*
[b] *Dept. of Computer Science, University of York, UK*

Abstract

In this article, we propose a multiscale method of embedding a graph into a vector space using diffusion wavelets. At each scale, we extract a detail subspace and a corresponding lower-scale approximation subspace to represent the graph. Representative features are then extracted at each scale to provide a scale-space description of the graph. The lower-scale is constructed using a super-node merging strategy based on nearest neighbor or maximum participation and the new adjacency matrix is generated using vertex identification. This approach allows the comparison of graphs where the important structural differences may be present at varying scales. Additionally, this method can improve the differentiating power of the embedded vectors and this property reduces the possibility of cospectrality typical in spectral methods, substantially. The experimental results show that augmenting the features of abstract levels to the graph features increases the graph classification accuracies in different datasets.

Keywords: Spectral graph embedding, diffusion wavelet, multi-resolution analysis, graph summarization, scale space

1. Introduction

The graph data structure improves the expressiveness of vectors by describing objects in terms of the relationships between parts. This expressiveness in

*Corresponding author
Email addresses: H.Bahonar@ec.iut.ac.ir (Hoda Bahonar), Mirzaei@cc.iut.ac.ir (Abdolreza Mirzaei), Richard.Wilson@york.ac.uk (Richard C. Wilson)

addition to the simplicity of presentation makes them so popular and there is a rapidly growing interest in representation of different objects by graphs in different fields of science. The graph of covalence relation between chemical molecules [1], the network of tertiary structure of proteins [2], and the skeleton of objects in images and videos [3] are some examples of objects represented by graphs.

The graph structure is determined by an unordered set of edges between an unordered set of nodes. This flexibility is a double-edged sword. On the one hand, it causes the simplicity of object representation and this simplicity results in better understanding of the internal relations of the objects. On the other hand, it makes the procedure of graph handling so time consuming. Different node permutations result in identical (or isomorphic) graphs, but the checking of this relationship between two graphs is an NP-complete problem [4]. There are some other problems in handling graphs, mentioned in [5, 6, 7, 8], which motivate researchers to embed graphs in vector space during the last two decades. Graph embedding in vector space tries to extract the differentiating graph features and insert them into vectors. These vectors can be processed subsequently through the numerous statistical pattern analysis methods in order to recognize patterns in corresponding graphs.

Of course some information is lost due to embed the flexible graph structure into a relatively limited vector structure, but the advantage of the embedded vectors is that we can utilize the full power of numerous statistical pattern recognition and machine learning techniques. It is crucial therefore that the embedding method captures as much of the rich graph structure as possible. Multi-resolution approaches have proved very effective in many areas of pattern recognition, particularly in image analysis and computer vision, where the multi-scale representation can pull out important features even when they appear at varying scales. This is the motivation for this paper, where we generate a multi-resolution embedding of a graph based on the graph diffusion wavelet concept [9]. We anticipate that such a representation will be richer and improve comparison and classification by revealing important structure at the different scales.

A scale-based representation is already implicit in a number of graph embedding methods, particularly those based on the graph eigendecomposition, where the eigenvalue, in some sense, reflects the scale of the feature. Several works in the literature establishes connection between the structural properties of graphs and the spectral features of their representation matrices [10]. These attempts are grouped in the field of spectral graph theory and they are used as the feature extraction approach in spectral graph embedding in vector space [5, 11, 12]. The mentioned spectral features are obtained in polynomial time [13] from the eigendecomposition of graph, which is stated as follows: $A = \Phi\Lambda\Phi^*$, where A, Φ and Λ are the representation matrix, the matrix with eigenvectors on its columns and the diagonal matrix of the eigenvalues, respectively. None of these generate an explicit scale-space representation or can control the scale of the representation.

The graph spectrum (i.e. the vector of graph eigenvalues) is invariant to different node permutations and has proved to be a useful graph embedding method [14]. Unfortunately the problem of cospectrality, i.e. different graphs having the same spectrum, and lack of distinctiveness limits this approach. Almost all trees have cospectral mate [15] and there are works to produce cospectral non-isomorphic graphs [16, 17]. This problem can be alleviated by using eigenvectors [18] but in this approach there is no direct connection to the structural aspects of the graph.

In this article, we try to create a rich graph embedding through a multi-resolution approach. Structural properties of graphs exist at different scales within the graph. Two graphs may be similar at one scale while they are different at another scale. For example, consider the comparison of two social networks. The networks may be different or similar in the communication between individuals (small scale), the structure of the constitutive communities (intermediate scale), and the overall structure of the network and connections between communities (large scale). The point at which scales convey the useful information about the differences/similarities depends on the application. So, mere extracting features from the initial graph, as it was done in previous works,

does not seem reasonable enough for all the applications.

For multi-resolution embedding proposed in this article, the spectrum of different levels of approximation and detail are encapsulated into a feature vector. The mapping of graph into the approximation and detail subspaces is done through the diffusion wavelet [9]. The summary graph of each level is extracted using this wavelet and some heuristics. This abstract graph[1] is utilized as the input of the diffusion wavelet for the next level. To the best of our knowledge, there is not any application of multi-resolution signal processing methods into graph embedding. The experimental results show that this approach can improve the classification accuracy in different applications.

In Section 2, some related works are presented. After declaring some basic concepts in Section 3, the proposed method is described in Section 4. The experimental results are displayed in Section 5 and finally Section 6 presents some conclusion remarks and future works.

2. Related work

The graph embedding methods can be divided into three groups: probing-based, prototype-based and spectral methods. In probing-based methods [6, 7], the frequent features are extracted from the graphs and their frequencies are embedded into the fixed sized vectors. In prototype-based methods [8, 21], some special prototypes are selected or made from training graphs and the differences of other graphs to these graphs are encoded into their feature vectors.

The spectral graph embedding is a prominent group of graph embedding methods, whose process can be divided into two steps. In the first step, the graph is coded into a matrix representing the binary relations between its vertices. In the second step, the invariants are extracted from the representation matrix and inserted into feature vector such that they can differentiate different graphs and assign similar vectors to similar graphs [18].

[1] The graph abstraction is used as a synonym for graph summarization in the literature [19, 20]. The output of graph summarization is an abstract graph.

Diverse concepts are used for defining the representation matrix. Adjacency matrix [5, 22], adjacency of oriented line graph [11], Laplacian matrix [23, 24], heat kernel [12, 25], and transition matrix of quantum random walk [26] are some instances of the introduced representation matrices. There are three approaches in the invariant extraction step. The first approach is to use the elements of representation matrix directly, such as β-complexity which uses the coefficients of decomposition of representation matrix into other matrices [27]. The second approach is to apply functions on the eigenvalues, e.g. min and max [5, 23, 26], sum [22], product [11, 12], and product of inverse [12] of the eigenvalues. The last approach is to augment feature vectors by the functions on the eigenvectors. The eigenvector related to the more important eigenvalue [28], the power series coefficients of heat kernel content [12], and the symmetric polynomials [18] are some instances of this approach. As it is noted in the introduction, here we use a different approach which utilizes the information contained both in the eigenvalues and the eigenvectors.

Motivated from the extensive and successful application of wavelet transform in the signal processing domain [29, 30, 31], during the last decade, the researchers tried to transfer this concept to the graph domain. A graph signal is a signal with the number of samples equated to the graph order (i.e. the number of graph vertices), each sample is assigned to a single vertex [32]. The time and frequency domains in classical signals correspond to the vertex and spectral domains in graph signals, respectively. Accordingly, the graph wavelets are divided into two categories: the vertex domain wavelets [33, 34, 35] and the spectral domain wavelets [9, 36].

Random transforms [35], shortest path wavelet [33], and lifting-based wavelet [34] are some examples of vertex domain wavelet designs. In random transforms two groups of bases are defined for each graph, the first group is defined based on the weighted average of graph signal samples in the neighborhood and the second group is defined based on the weighted difference of them. In shortest path wavelet, for each vertex, the weighted averages of graph signal samples on its neighborhood with different radii are computed as its large scale descriptions.

Lifting-based wavelet design divides the vertices into two sets and defines the wavelet coefficients of each set based on their neighbors in the other set.

Spectral graph wavelet [36] and diffusion wavelet [9] are some examples of spectral domain wavelets. The spectral graph wavelet is introduced according to the eigendecomposition of the Laplacian matrix. The mother wavelet is defined by utilizing the eigenvectors and applying a specially designed kernel function on the eigenvalues. Scaling of the mother wavelet is defined by multiplying in the spectral domain and its translation to a special vertex is done by applying the spectral graph wavelet on a pulse located on that vertex. The diffusion wavelet is a successful wavelet design in the spectral domain in which the approximation and detail subspaces are computed based on the orthonormal bases of these subspaces. These subspaces offer a multi-resolution analysis for the graph domain.

Diffusion wavelet is applied to some applications successfully. Gudivada [37] constructed a fully connected similarity graph of data points using a Gaussian kernel function. Afterwards, he calculated the data points coordinates in the reduced dimension space of a special scale using the extended scaling functions of the diffusion wavelet. He used the extended scaling functions of the covariance matrix as the eigenfaces in face recognition application. He also used this idea in optical flow estimation by comparing the scaling functions of each block of a frame against the scaling functions of the blocks of the search window in the previous frame. Wang and Mahadevan [35] utilized the extended scaling functions of the diffusion wavelet to identify the topic hierarchy contained in a document corpora and inferred the correlation of the documents to the topic hierarchy.

All of the works related to the graph signal processing is applied on a single graph or multiple graphs with identical node sets, because the graph signal processing is conceptually node order variant. Unlike the other works, in this article, the information gathered from diffusion wavelet is used for comparing different graphs with different vertex and edge sets. The extended scaling and wavelet functions are used for extracting the summary and detail sub-graphs

in every processed graph. A base embedding method is then employed on top of this information and makes the graph feature vector more informative. This claim is assessed using different experiments.

3. Basic concepts

Let $G = (V, E) \in \mathcal{G}$ be an undirected and unlabeled graph with V as its vertex set and $E \subseteq V \times V$ as its edge set.

The adjacency matrix A of the undirected and unlabeled graph G is a symmetric $|V| \times |V|$ square matrix, which is defined as:

$$A = \left[a_{ij} = \begin{cases} 1 & \text{if } \{i, j\} \in E \\ 0 & \text{otherwise} \end{cases} \right]. \tag{1}$$

The Laplacian matrix L for graph G is derived through equation $L = D - A$, where D is the diagonal matrix of the vertex degrees. The transition matrix T represents the possibility of transition between the pair of vertices in a single time-step of a random walk and here it is derived from the adjacency matrix through the equation

$$T = D^{-1/2} A D^{-1/2}. \tag{2}$$

This can be considered as a diffusion operator on the graph [9]. The t^{th} power of this matrix, i.e. T^t, represents transitions between pairs of vertices after t time-steps. This property is a motivation for defining a multi-resolution analysis for graphs in [9], where the dyadic powers of the transition matrix T are used. The l^{th} resolution level is found after $t_l = 2^l$ time-steps and the transition matrix at this resolution level is $T_l = T^{2^l}$.

The graph $G \setminus \{v_i, v_j\}$ is the vertex identification on graph G for the vertices v_i, v_j, such that in the new graph, two vertices v_i and v_j are gathered into one single vertex $v_{\{i,j\}}$ incident to all the edges which have either of two vertices v_i and v_j as one of their ends. The edge $\{v_i, v_j\}$ (if exists) is converted to a loop inserted on $v_{\{i,j\}}$.

The key utility of diffusion wavelets is that, as T is a diffusion operator, the detail present in any function on the graph reduces at each time-step, and so can be represented more compactly. The graph can therefore be reduced to an approximation subspace, representing a lower resolution graph, and a detail subspace, representing the detail removed at the current level. If S_l is the approximation subspace at level l, then the detail subspace W_l is the orthogonal complement:

$$S_l = S_{l+1} \oplus W_l. \tag{3}$$

As the decomposition progresses, we expect the approximation subspace S_l to get smaller, although not necessarily at every step as it may remain the same size.

We now introduce some key terms relating to the wavelet decomposition and graph representation.

- *Approximation subspace*: The subspace which contains the approximate wavelet representation at the current resolution level. This reduces in size as the resolution decreases.
- *Detail subspace*: The subspace which contains the detailed wavelet representation at the current resolution level, representing the information discarded at this level.
- *Summary graph, summary features*: A graph representing the reduced-resolution graph at the current level, the graph embedding features of this graph[2].
- *Detail graph, detail features*: Similarly, a graph representing the detail lost from the original graph at the current level and the graph embedding features of this graph.

[2]For consistency with wavelet and graph summarization literature, in this article, the word "approximation" is used for subspaces and the word "summary" is used for graphs, adjacency matrices and features.

The key to the graph wavelet decomposition is to define a basis for the approximation subspace. This basis is used for representing the approximation subspace at the next level, reducing the size of the subspace. Here we follow [9] and seek a basis set which is localized and can ϵ-approximate the diffusion operator at the next level $l+1$.

At the resolution level l, the approximation subspace S_l is represented in a level specific basis set Φ_l (scaling functions), denoted by $[S_l]_{\Phi_l}$. This basis set is represented in the basis from the previous level, Φ_{l-1}. So there is a transformation from the previous basis to the current one, expressed as $[\Phi_l]_{\Phi_{l-1}}$. The detail subspace $[W_l]_{\Psi_l}$ is represented in the basis of Ψ_l (the wavelet functions). This basis is represented in the current basis as $[\Psi_l]_{\Phi_l}$ and represents the details removed in the current resolution level.

In almost all the applications, there is need to express these functions in the initial space V_0 on some initial basis Φ_0 (usually the standard basis). These extended bases are

$$[\Phi_l]_{\Phi_0} = [\Phi_l]_{\Phi_{l-1}}[\Phi_{l-1}]_{\Phi_{l-2}} \dots [\Phi_2]_{\Phi_1}[\Phi_1]_{\Phi_0}. \tag{4}$$

$$[\Psi_l]_{\Phi_0} = [\Psi_l]_{\Phi_l}[\Phi_l]_{\Phi_{l-1}}[\Phi_{l-1}]_{\Phi_{l-2}} \dots [\Phi_2]_{\Phi_1}[\Phi_1]_{\Phi_0}. \tag{5}$$

Following the methods of Coifman and Maggioni [9], the scaling functions are obtained by a QR-decomposition of the level l diffusion operator, represented in Φ_l on the domain and Φ_{l+1} on the range, i.e. $[T^{2^l}]_{\Phi_l}^{\Phi_{l+1}}$. Similarly, the wavelet function is obtained through decomposition of $I - [\Phi_{l+1}]_{\Phi_l}[\Phi_{l+1}]_{\Phi_l}^T$. In this process, the columns of the input matrix are considered as the vectors of the underlying space and the basis is derived through orthogonalization of these vectors. $\{\phi_{l,k}\}_{k\in J_l}$ and $\{\psi_{l,k}\}_{k\in K_l}$ are the bases of extended scaling functions and extended wavelet functions, respectively, where J_l and K_l are the index sets of the selected columns for these bases by orthogonalization process. This process is explained in detail in Section 4.3.

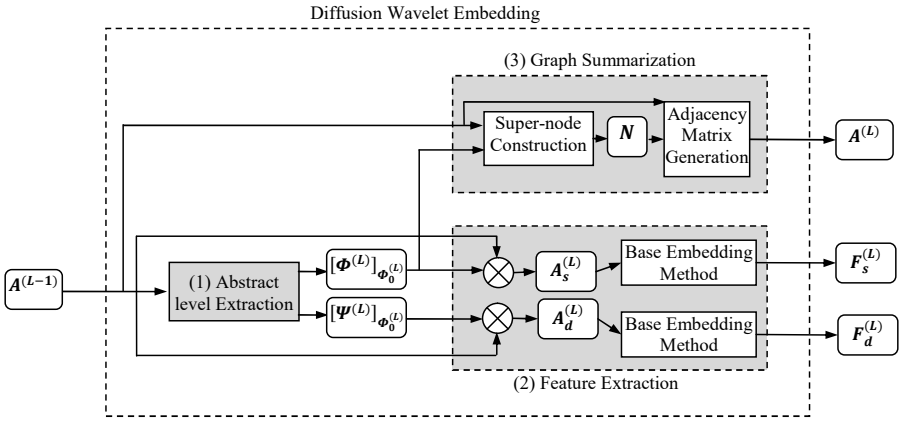


Figure 1: The flowchart of one level of DWE. Three basic sub-routines are determined in bold face, each of which is described in its corresponding paragraph.

4. Proposed method

A multi-resolution approach for graph comparison needs to extract different abstract levels[3] of graphs and be able to compare the corresponding abstract levels with each other. The proposed multi-resolution method called diffusion wavelet embedding (DWE) method, relies on diffusion wavelet decomposition for extracting these abstract levels. Subsequently, a graph feature extraction method is used to enable efficient comparison between the graphs extracted at different levels. The flowchart of one level of DWE is drawn in Fig. 1.

As it can be seen, DWE consists of three sub-routines: 1) Abstract level extraction, 2) Feature extraction, and 3) Graph summarization. In the first sub-routine $A^{(L-1)}$, the adjacency matrix of the previous abstract level, is processed through the diffusion wavelet and $[\Phi^{(L)}]_{\Phi_0^{(L)}}$ and $[\Psi^{(L)}]_{\Phi_0^{(L)}}$, the bases of approximation and detail subspaces of L^{th} abstract level, are returned. Note that L, the abstract level, is not the same as the resolution level l. The summary and detail features of this level, $F_s^{(L)}$ and $F_d^{(L)}$, are extracted in the second sub-

[3]One abstract level is identified with multiple resolution levels in the diffusion wavelet decomposition. As it will be described, each abstract level encapsulates the diffusion wavelet computations for one or more resolution levels. In this article, the superscript $^{(L)}$ indicates the L^{th} abstract level, while the subscript $_l$ represents the l^{th} resolution level.

routine through applying a base feature extraction method on $A_s^{(L)}$ and $A_d^{(L)}$, the symmetric matrices obtained from the mappings of the input graph to the approximation and detail subspaces. Finally, the next abstract graph, $A^{(L)}$, is built through graph summarization sub-routine and transmitted to the next level for further processing. This sub-routine is performed in two steps. In the first step, the set of super-nodes, N, are constructed using $A^{(L-1)}$ and $[\Phi^{(L)}]_{\Phi_0^{(L)}}$ and in the second step, the adjacency matrix of the abstract graph is generated from $A^{(L-1)}$ and N.

4.1. Abstract level extraction

The objective here is to extract the next abstract level from the input graph through application of the diffusion wavelet. We begin with the input graph G with adjacency A^0. This matrix can be written as

$$A = [u_1, u_2, \ldots, u_{|V|}], \tag{6}$$

where u_i is the i^{th} column of the adjacency matrix. The j^{th} element of u_i, $A_{i,j}$, represents the power of connection between the vertices v_i and v_j. The strategy is to apply the diffusion wavelet decomposition on diffusion operator T to reduce the dimension of the approximation subspace. At each application of the wavelet decomposition the level of detail is reduced, but not necessarily enough to allow us to represent the approximation space in a lower dimension. As a result, we may need to apply the decomposition multiple times before we can reduce the dimensionality of the subspace. Each dimensionality reduction is an *abstract level* and contains one or more *resolution levels*.

Let S_0 be the approximation subspace in which the abstract graph of level $L-1$, $G^{(L-1)}$, and its adjacency matrix, $A^{(L-1)}$, are defined. Assume that at abstract level L, a dimension reduction occurs after $m^{(L)}$ diffusion wavelet decompositions. This means that we found a sequence of wavelet decompositions

such that

$$S_0 = S_1 = \cdots = S_{m^{(L)}-1} \supset S_{m^{(L)}}, \tag{7}$$

where S_l is the approximation subspace obtained from applying diffusion wavelet decomposition l times. In the intermediate subspaces the number of dimensions is equal to the number of dimensions of S_0, so each vertex is described with the same number of coordinates as the vertices in $G^{(L-1)}$. On the other hand, the dimension reduction in $S_{m^{(L)}}$ means that every vertex can be represented by fewer coordinates and the number of vertices describing the graph can be reduced.

4.2. Feature extraction

At the current level, $[\Phi^{(L)}_{m^{(L)}}]_{\Phi_0^{(L)}}$ and $[\Psi^{(L)}_{m^{(L)}}]_{\Phi_0^{(L)}}$ are the bases of the approximation and the detail subspaces, which are derived through eq. 4 and eq. 5, respectively. We need to map $A^{(L-1)}$ into these subspaces in order to find the graph representation at level L. The vertex coordinates are mapped into these subspaces by projection onto the bases:

$$X^{(L)} = A^{(L-1)} \times [\Phi^{(L)}_{m^{(L)}}]_{\Phi_0^{(L)}}, \tag{8}$$

$$Y^{(L)} = A^{(L-1)} \times [\Psi^{(L)}_{m^{(L)}}]_{\Phi_0^{(L)}}, \tag{9}$$

where $X^{(L)}$ represents the summary embedding and $Y^{(L)}$ represents the detail embedding. The resulting $X^{(L)}$ and $Y^{(L)}$ matrices are not square because of the reduction in dimensionality. For example, $X^{(L)}$ is a $|S_0| \times |S_{m^{(L)}}|$ matrix. For yielding real-valued eigenvalues and eigenvectors (which are the raw materials of almost all the spectral embedding methods), the processed matrix needs to be square and symmetric. This task is done through the following equation:

$$A_s^{(L)} = X^{(L)} {X^{(L)}}^T. \tag{10}$$

There is a similar equation for the detail space. The resulting $A_s^{(L)}$ and $A_d^{(L)}$ have the same dimensions as $A^{(L-1)}$, i.e. $|S_0| \times |S_0|$, but they possess less information. This kind of representation allows us to identify the original graph vertices with the reduced resolution version. In fact the mappings of the input graph to the approximation and detail subspaces should have the dimensions of $|S_{m^{(L)}}| \times |S_{m^{(L)}}|$ and $(|S_0| - |S_{m^{(L)}}|) \times (|S_0| - |S_{m^{(L)}}|)$, respectively. But it should be noted that the resulting matrices $A_s^{(L)}$ and $A_d^{(L)}$, are not used as a lower resolution representations. These matrices are used as the inputs of some base graph feature method $f(G)$ to extract the graph features in the next step. The proposed diffusion wavelet embedding method for this purpose is defined as follows:

Definition 1. *Diffusion Wavelet Embedding (DWE):* Given the set of abstract subgraphs $\mathcal{A} = \{A\} \cup \{A_s^{(L)} | L = 1 \dots \rho\} \cup \{A_d^{(L)} | L = 1 \dots \rho\}$ for reference graph $G \in \mathcal{G}$ and the base embedding method $f : \mathcal{G} \to \mathbb{R}^m$, diffusion wavelet embedding F is:

$$\begin{aligned} &F_{f,\mathcal{A}} : \mathcal{G} \to \mathbb{R}^{m \times (2\rho+1)} \\ &F_{f,\mathcal{A}}(G) = [f(A), f(A_s^{(1)}), f(A_s^{(2)}), \dots, f(A_s^{(\rho)}), \\ &\qquad\qquad f(A_d^{(1)}), f(A_d^{(2)}), \dots, f(A_d^{(\rho)})]. \end{aligned} \tag{11}$$

4.3. Graph Summarization

To proceed with the DWE at the next abstract level, a graph summarization method is needed. Here a diffusion wavelet-based method is proposed which derives the abstract graph, $A^{(L)}$, from the input graph, $A^{(L-1)}$, through applying the basis of the summary space $[\Phi_{m^{(L)}}]_{\Phi_0}$ to it. This procedure consists of two steps: 1) super-node construction and 2) adjacency matrix generation. In the first step, the vertices of the input graph should be partitioned into super-nodes which play the role of the vertices of the graph at the next level. For this purpose, $V(G^{(L-1)})$ is divided into two groups: the participant and the deleted vertices. These two classes of vertex are defined during the wavelet decomposition process. There, the basis space is constructed by taking the first column of

A as the first basis. The next basis is formed through orthogonalization of the vector with greater difference to its image on the basis constructed so far. This process is repeated until the difference is lower than a pre-specified threshold. The columns which participate in basis construction through orthogonalization indicate participant vertices (with the indices included in $J_{m^{(L)}}$) and the vertices which are discarded up to desired threshold are called deleted vertices.

Each participant vertex is considered as the representative of a super-node. So there exist $|J_{m^{(L)}}|$ super-nodes, $N = \{n_1, n_2, \ldots, n_{|J_{m^{(L)}}|}\}$. Now it is time to assign the deleted vertices to the super-nodes. Two approaches are proposed here for this purpose: Nearest Neighbor (NN) and Maximum Participation (MP).

In NN approach, the deleted vertices are assigned to their nearest super-node in the target space. For this purpose, a 1NN classifier is used while the columns of $X^{(L)}$ corresponding to the participant vertices are the training points and the columns of $X^{(L)}$ corresponding to the deleted vertices are the test points. $X^{(L)}$ is computed through eq. 8.

Motivated from [38], in the MP approach, the j^{th} entry of the extended scaling function corresponding to the super-node n_k, is considered to be the amount of participation of v_j in constructing this super-node. Thus, each deleted vertex v_j is assigned to the super-node n_k which has more participation in making this super-node relative to the others. In other words, v_j shows more tendency to make n_k rather than other super-nodes. This heuristic is expressed in the following equation:

$$\mathcal{S}(v_j) = \underset{k, k \in J_{m^{(L)}}}{\operatorname{argmax}} \phi^{(L)}_{m^{(L)},k}(j), \tag{12}$$

where $\mathcal{S}(v_j)$ is the super-node which the deleted vertex v_j is assigned to.

After assigning all the deleted vertices to the super-nodes, partitioning the members of $V(G^{(L-1)})$ into super-nodes, $N = V(G^{(L)})$, is completed. The next step is to insert edges between these vertices and form $E(G^{(L)})$ through adjacency matrix generation. For this purpose the vertex identification operator

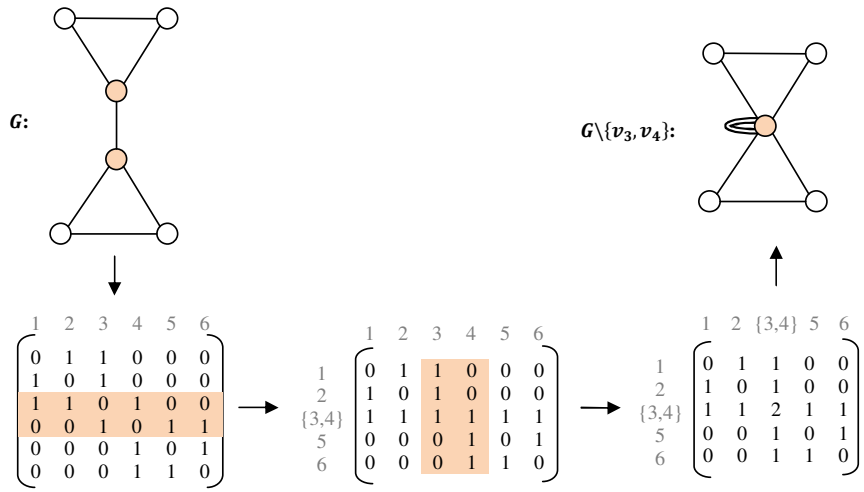


Figure 2: An example for vertex identification on the vertices of the super-nodes.

is used, and an example of this operation is shown in Fig. 2. Two vertices $\{v_3, v_4\}$ are merged in the natural way by including any edges external to the pair. Internal edges become self-loops. The resulting graph $G \setminus \{v_3, v_4\}$ is shown at the top-right of Fig. 2.

4.4. The Algorithm

The algorithm of the proposed method is shown in Algorithm 1. The inputs of this algorithm are the graph adjacency matrix A, the number of abstract levels ρ, the diffusion wavelet threshold ε, the super-node construction method θ, and the base embedding method $f : \mathcal{G} \to \mathbb{R}^m$. The output is the $m \times (2\rho+1)$ dimensional feature matrix F. At first, the embedded vector of the reference graph, $f(A)$, is considered as the first column of the feature matrix F. The main body of the algorithm is repeated ρ times. In each iteration, the operation corresponding to a single abstract level is performed. In step (i), each time, one level of diffusion wavelet is applied on the transition matrix T through the function `Diffusion_Wavelet`. If the dimension of the approximation subspace is reduced, this subspace is considered as the approximation subspace of the abstract level L, otherwise another level of the diffusion wavelet is applied

on the projection of T on the scaling functions. The inputs of the function `Diffusion_Wavelet` are the transition matrix T, the extended scaling functions of the previous resolution level $[\Phi_{l-1}]_{\Phi_0}$, the basis of the initial space Φ_0, and the threshold ε. Its outputs are the mapping of transition matrix T on the approximation subspace of the current resolution level, the extended scaling and wavelet functions of this level $([\Phi_l]_{\Phi_0}, [\Psi_l]_{\Phi_0})$, and the indices J_l, K_l of the dimensions involved in the summary and detail spaces of T, respectively. In step (ii), The vertex coordinates in the previous abstract level is mapped to the approximation and wavelet subspaces, the resulting matrices are converted to the square symmetric ones, and their embedded vectors are inserted into feature matrix F. In step (iii), every participant node initiates a super-node as its representative. The super node of every deleted node is either set to be its nearest representative in the mapped coordinates or selected according to eq. 12, depending on the choice of method θ being NN or MP. For the first case, determining the 1-nearest neighbor is equivalent to finding the nearest point in terms of Euclidean distance (or any other distance metric) and the mapped coordinates is fetched from matrix X. In step (iv), the adjacency matrix of the new abstract graph is computed through applying function `vertex_identify` each time on the members of one super-node. This function takes the adjacency matrix A and the super-nodes $N(i)$ as input and identifies the members of $N(i)$ in the output adjacency matrix. Finally, the transition matrix and its space are updated for the next iteration.

The time complexity of the algorithm is principally dependent on either the base embedding method or the diffusion wavelet decomposition process. The diffusion wavelet decomposition relies on QR decomposition of a matrix which is typically $O(|V|^3)$, with the usual speed-up for sparse graphs. This process must be repeated for each wavelet level, but in the experiments here, the number of levels required is small (1 in half of the cases and less than 7 in 93 percent of the cases). We use the Laplacian spectrum embedding method, which is again $O(|V|^3)$ and the overall complexity is $O(l|V|^3)$ where l is the total number of wavelet levels required.

Algorithm 1 The algorithm of diffusion wavelet embedding method.

Inputs: A: adjacency Matrix
ρ: Number of abstract levels
ε: Diffusion wavelet precision
θ: the super-node construction method (NN or MP)
f: the base embedding method
Output: F: the feature matrix

Add $f(A)$ as a column of matrix F
$T \leftarrow$ `Transition_Matrix`(A)
$S \leftarrow S_0$
for $L = 1, \ldots, \rho$ **do**
▷ (i)**. Abstract level extraction**
  $\Phi_0 \leftarrow$an orthonormal basis which ε-spans S
  $J_0 \leftarrow \{1 \ldots \mathrm{col}(T)\}$
  $l \leftarrow 1$
  while $J_{l-1} = J_l$ **do**
    $[[\Phi_l]_{\Phi_0}, [\Psi_l]_{\Phi_0}, (J_l, K_l), T] \leftarrow$ `Diffusion_Wavelet`$(T, [\Phi_{l-1}]_{\Phi_0}, \Phi_0, \varepsilon)$
    $l \leftarrow l + 1$
  end while
▷ (ii)**. Feature extraction**
  $X \leftarrow A \times [\Phi_l]_{\Phi_0}$
  $A^s \leftarrow XX^T$
  $A^d \leftarrow (A \times [\Psi_l]_{\Phi_0})(A \times [\Psi_l]_{\Phi_0})^T$
  Add $f(A^s)$ and $f(A^d)$ as two columns of matrix F
▷ (iii)**. Graph Summarization: Super-node construction**
  for every participant node i, add i to $\mathcal{S}(i)$
  for every deleted node i, add i to $\mathcal{S}\left(\begin{cases} \underset{k,k\in J_l}{\mathrm{argmin}} \|X(J_l(k)) - X(i)\|_2 & \text{if } \theta \text{ is NN} \\ \underset{k,k\in J_l}{\mathrm{argmax}}\, \Phi_l(i,k) & \text{if } \theta \text{ is MP} \end{cases}\right)$
▷ (iv)**. Graph Summarization: Adjacency matrix generation**
  for every $\mathcal{S}(i)$, $A \leftarrow$ `vertex_identify`$(A,$ members of $\mathcal{S}(i))$
  $T \leftarrow$ `Transition_Matrix`(A)
  $S \leftarrow \langle \{\phi_{l,k}\}_{k\in J_l} \rangle$
end for

As it can be seen, unlike the previous embedding methods which return a feature vector for each graph, the output of DWE is a feature matrix.

5. Experiments

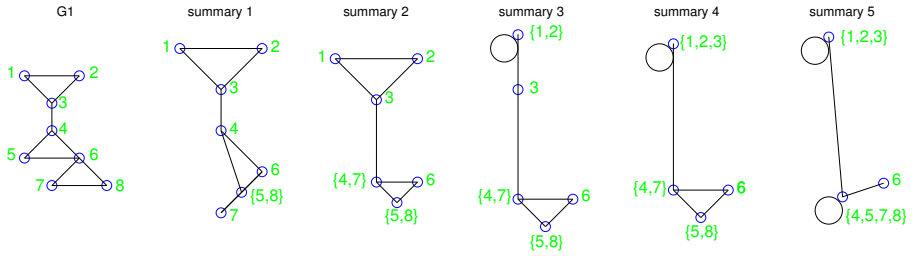
In this section, we report the results of experiments to show the effectiveness of the summarization and the embedding method. To begin, we examine the properties of the graph summarization; in section 5.1 we use a toy example to illustrate how the graph reduces in resolution whilst maintaining the key structures. In section 5.2 we evaluate the accuracy of the summarization using synthetic data with varying structure.

In section 5.3 we evaluate the performance of multi-scale representation with respect to edit distance, which is considered the gold-standard measure of graph dissimilarity. Finally, in section 5.4 we explore whether real data has the multi-scale properties which make our approach useful and compare the performance of the method with other state-of-the-art algorithms on real datasets.

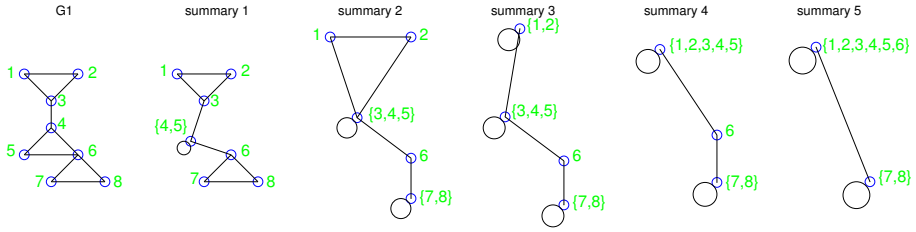
In these experiments, three abstract levels are used in addition to the base level. So the feature vector set is $\{f(A), f(A_s^{(1)}), f(A_s^{(2)}), f(A_s^{(3)}), f(A_d^{(1)}), f(A_d^{(2)}), f(A_d^{(3)})\}$. The base embedding method is Laplacian spectrum, which is simple and powerful. The classification accuracies are estimated using 5NN classifier. The reported values are the averages of 10 separate runs of 5-fold cross validation.

5.1. Toy example for summarization

In this section, the summaries of a sample graph are represented for comparison between NN and MP approaches in super-nodes construction. This graph, G, illustrated in Fig. 3a has three triangle structures composed of three node sets $\{1, 2, 3\}$, $\{4, 5, 6\}$, and $\{6, 7, 8\}$. The second and the third triangles are closer to each other and they are separated from the first triangle by means of the edge $\{3, 4\}$. It is expected for an appropriate summarization method to encapsulate the vertices of each triangle into a separate super-node in initial abstract levels and cluster the second and the third triangles into one super-node



(a) DWE-NN method



(b) DWE-MP method

Figure 3: The example for different super-node construction approaches in proposed graph summarization.

in further iterations. As shown in Fig. 3a, NN approach which clusters nodes with similar relations in every step, could not differentiate between the second and the third triangles which are located close to each other, but it could specify the coarser node clusters in further steps. As Fig. 3b exhibits, MP approach met the first expectation by constructing a node cluster for each triangle up to the third summary level (however the super-node members are not as we expect), but the coarser node clusters could not be established in further steps using this approach, as the separating structure of edge $\{3, 4\}$ disappeared in initial summarization steps. It can be concluded that using NN approach offers more hope for constructing suitable medium and large size super-nodes, but the impact of imprecise small super-nodes on overall summarization accuracy should be more investigated.

5.2. Accuracy of summarization on random graphs

The previous section exhibited some behaviors of NN and MP approaches in super-node construction. In order to judge between these two approaches, they should be applied on a set of graphs with known multi-scale structures. We prefer to use a synthetic graph set rather than a real one. The reason is twofold. First, there is not a straightforward method to realize the inherent node clusters in a real dataset, while in a synthetic dataset, the composition of the inherent node clusters is under control. Second, although the overall accuracy of DWE method using either of these two super-node construction approaches can be estimated on real datasets (as in section 5.4), we cannot conclude from the overall accuracy that the differences are merely due to the difference between the selected super-node construction methods, because the final results affect by the subsequent steps as well. The behaviors of these steps may vary in applying on different structured graphs resulted from NN and MP approaches.

In the synthetic graph set, we composed some inherent node clusters called communities such that they have many within-cluster relations but a few between-cluster relations. The members of these communities are recorded for estimating the accuracies of NN and MP approaches, in the following. We use a synthetic graph set composed of 100 random graphs in this experiment. The number of communities and the number of vertices included in each community are picked randomly from the sets $\{2, 3, 4\}$ and $\{6, 7, 8\}$, respectively. These numbers are chosen as a compromise between providing a test for the graph summarization algorithm, and the computational complexity of testing on large sets of large graphs. Every vertex has 4 to 8 adjacent vertices within the community and every community has 1 to 4 adjacent communities. The adjacent vertices which connect the communities are picked randomly and every two graph communities are connected through at most 1 edge. The graphs are connected. Some of the synthesized graphs are drawn in Fig. 4. These graphs have a community structure which varies in size and number.

To estimate the node clustering accuracy, the last level super-nodes with the size greater than one are considered for every graph and all possible two

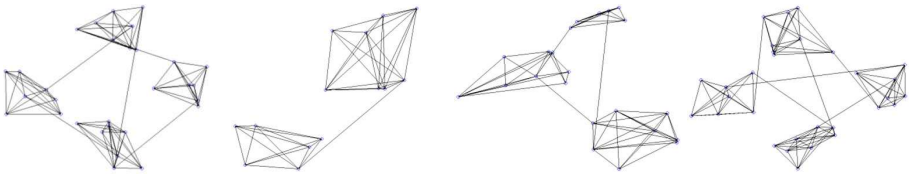


Figure 4: Some of the random graphs synthesized for comparing NN and MP approaches against each other.

Table 1: The estimation of node clustering accuracy for NN and MP approach.

Method	$\#homos$	$\#heteros$	$\#all$	$\%SNCprecision$
NN	3266	655	3921	83.2951
MP	2318	917	3235	71.6538

subsets of their vertices are checked for being in the same community. We call the relation of being in the same community as homogeneity and that of being in different communities as heterogeneity. The number of homogeneities and heterogeneities in every graph is computed and the results are summed over all graphs to obtain the values $\#homos$ and $\#heteros$, respectively. The precision of every super-node construction method is estimated as follows:

$$SNCprecision = \frac{\#homos}{\#all}, \tag{13}$$

where $\#all = \#homos + \#heteros$ is the total number of two subsets checked for all graphs. The results are tabulated in Table 1. As it can be seen from the $\#all$ column, NN approach can cluster more nodes and obtain more abstract graphs, additionally its $\#homos$ and $\#heteros$ are greater and less than their counterparts in MP approach, respectively. Thus we use NN approach in super-node construction from now, as its estimation of node clustering accuracy is better than MP approach.

5.3. Following the edit distance

Graph edit distance between two graphs is defined as the minimum cost edit operations (e.g. edge deletion) needed for transforming first graph into second one [39], and is considered the gold standard to assessing the similarity

of two graphs. The feature distance of an appropriate embedding method is expected to follow the graph edit distance (which is accurate but expensive to compute). In our case, if two graphs have a small edit distance from each other, the distance of their DWE feature vector should be small as well and vice versa. Our experiment for studying this requirement is as follows.

A seed Delaunay triangulation graph, shown in Fig. 5, is generated with 100 vertices, the (x, y) coordinates of which are the real numbers picked randomly from the range $[1-100]$. We then delete successive random edges from the graph to yield a sequence of graphs with increasing edit distance from the original. In this way, we produce a set of graphs with known edit distance from 1 to 30. The long vector of DWE feature matrix of each graph in the sequence is extracted and its Euclidean distance from the feature vector of the seed graph is considered. This process is repeated 1000 times and the average feature distance of graphs with each value of edit distance is computed. Fig. 6 shows this average values in contrast to their corresponding edit distances. NN and MP approaches are used as the super-node construction approach in DWE-NN and DWE-MP, respectively. It can be seen that the feature distances follow the trend of the edit distances, especially in DWE-NN method. However the deviation from the mean value is considerable.

In these experiments, we use the Laplacian spectrum as the base embedding method. It is well known that this embedding suffers from the problem of cospectrality, which means that two graphs have the same spectrum and hence the same embedding [14, 26]. As a result, the embedded distance is zero, but the edit distance is non-zero. The diffusion wavelet embedding can alleviate this problem because it explores multiple scales where the cospectrality problem may not exist.

We examined a number of cospectral graph sets from [26]. Three sets of Strongly Regular Graphs (SRGs) and two sets of Balanced Incomplete Block Designs (BIBDs) are used. The DWE successfully distinguishes the cospectral sets SRG(25,12,5,6), SRG(26,10,3,4) and BIBD(15,3,1) but not SRG(36,15,6,6) and BIBD(23,11,5). DWE does not offer a complete solution to the problem of

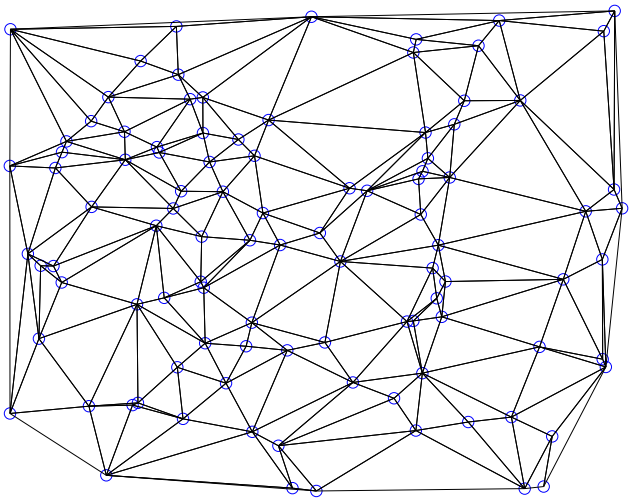
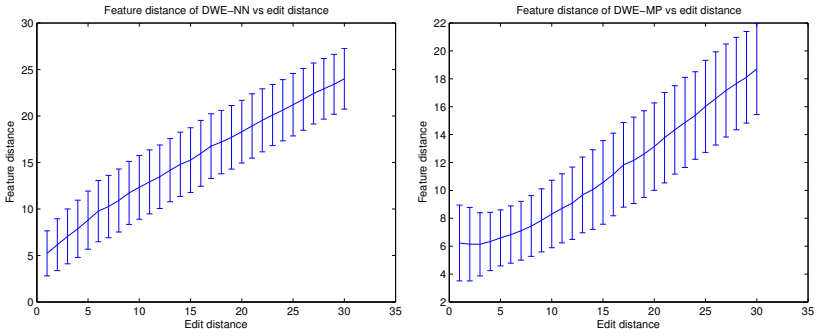


Figure 5: The random Delauney triangulation seed graph.



(a) DWE-NN method (b) DWE-MP method

Figure 6: The feature distances of different versions of DWE vs edit distances.

cospectrality but improves the performance of the base embedding method.

5.4. Classification accuracy of DWE on real datasets

In our final experiment, we examine the performance of DWE of real-world data in classification problems. We use eight different graph datasets which are common in the graph classification literature, the properties of which are tabulated in Table 2. The first five datasets consist of object detection graphs which are extracted from images; however the method of the graph extraction, the graph sizes and their other properties are different. Among them, The COIL15 dataset is of 15 classes of COIL-DEL dataset of [40] with the relatively large graphs. The final three datasets are bio- and chemo-informatics datasets. Enzymes contains graphs describing the teriary structures of protein molecules. PTC is a chemical structure dataset with graphs representing atoms and bonds. PPI is a protein-protein interaction dataset.

The key motivation for our method, diffusion wavelet embedding, is that the important structures in graph datasets can occur at different scales, and we need a scale-space representation. To explore whether this is true with real data, we begin by looking at the discriminating power of the different wavelet levels. We use three wavelet levels, and each level has an approximation and detail feature sets. There is also a feature set for the original reference graph, giving 7 feature sets in total. We denote these by $\{R, A1, A2, A3, D1, D2, D3\}$ where the letter refers to the type of representation, and the number to the level. A and D refer to all the approximation features and all the detail features respectively. In the first set of experiments, we combine the chosen features naïvely, by concatenating them into a single long-vector representation. Fig. 7 shows the accuracy of some level combinations for different datasets.

It can be seen that the accuracies of different level combinations differ from one dataset to another. It can be concluded that, for different datasets, the important structural information is laid in different abstract levels. For example, PPI consists of the relatively large graphs and the results of its two last wavelet decomposition combinations suggest that its large scale interactions are more

Table 2: The description of the tested real datasets.

Dataset	Description	#Vertices (min, max, ave)	#Edges (min, max, ave)	#Graphs	#Classes
Llow [40]	The graphs of handwriting letters with low, medium and high distortions, respectively. The objective is the letter class detection.	(1, 8, 4.68)	(0, 6, 3.13)	2250	15
Lmed [40]		(1, 9,4.67)	(0, 7, 3.21)	2250	15
Lhigh [40]		(1,9,4.67)	(0, 9, 4.5)	2250	15
Shock [3]	The graphs of skeleton of object images. The objective is object detection.	(4, 33, 13.17)	(6, 64, 24.33)	150	10
COIL15 [40]	The triangulation object graphs with corners as vertices. The objective is object detection.	(18, 77, 42.73)	(45, 222, 116.49)	585	15
Enzymes [2]	The graphs of protein tertiary structures. The objective is to detect the type of enzyme.	(2, 126, 32.63)	(1, 149, 62.14)	600	2
PTC [41]	The graphs of chemical compounds. The objective is to detect the carcinogenicity.	(2, 109, 25.56)	(2, 216, 51.92)	344	2
PPI [42]	The protein-protein interaction networks. The objective is to detect the type of bacteria.	(3, 232, 109.60)	(4, 3006, 864.37)	86	2

informative than its small scale interactions. This property is not observed in the COIL15 dataset, even though its graphs are relatively large. The COIL15 graphs are the object structure graphs and the reference graph consists of the subtle structures is so important, accordingly removing this level is a big mistake.

It is clear from Fig. 7 that the chosen features have a big impact and this varies between datasets. This supports our contention that different scales are important in different datasets. Generating the long vector of the level combinations makes the implicit assumption of the same importance for all levels of detail and the results of Fig. 7 show that it is not an effective strategy. Instead we need to learn which levels are important for the data. To this end, we now use ensemble learning.

In ensemble learning methods, a single classifier is learned based on each feature set and the final decision is made by merging the decisions of these

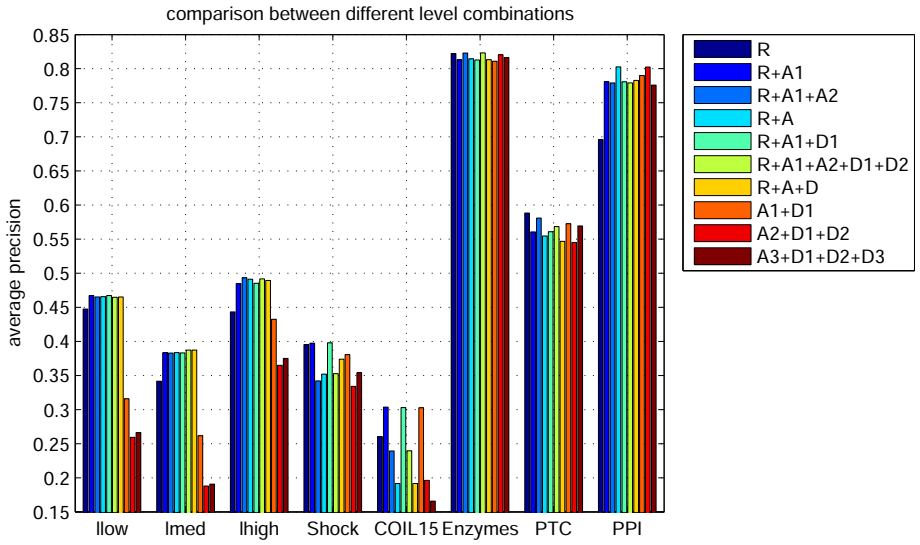


Figure 7: Accuracies of applying 5NN classifier on some different level combinations' long vectors in the tested datasets.

single classifiers [43]. Here, seven 5NN classifiers $D_i, i = 1, \ldots, 7$ are learned for the feature vectors of the base level, each of the approximation levels and each of the detail levels, separately. $DP_x(i,j) = \hat{P}_i(C_j|x)$, is the probability of being sample x in class j estimated by classifier i. In our case, this value is defined as the fraction of the five nearest neighbors of sample x which are in class j. For example, in a two class problem, if classifier i confronted with three neighbors of the first class and two neighbors of the second class for sample x, we have $DP_x(i,.) = [0.6, 0.4]$. Different ensemble learning methods apply different combination strategies to combine these values estimated by different classifiers to conclude about the probability of being x in each class j, $\hat{P}(C_j|x)$. Of course the class of the given sample x, $\hat{F}(x)$, is the class with the bigger probability value.

We explore a number of different ensemble combination methods. Max, min, average, and product methods simply combine the values $DP_x(i,j), i = 1 \ldots 7$ by max, min, average, and product operators, respectively. In the majority vote method, $\hat{F}(x)$ is the class which the majority of the classifiers vote for

it. Weighted vote is similar to majority vote except that the vote of classifier i for class j is equal to $DP_x(i,j)$. The Bayes ensemble method uses values of confusion matrices of different classifiers to estimate $\hat{P}(C_j|x)$. In the decision template method, the responses of different classifiers to the training samples of each class are structured in the decision template of that class. $\hat{F}(x)$ is the class with maximum similarity of its decision template to DP_x. In Dempster-Shafer method, the belief degree of each classifier about sample x being in each class enter to the computations. For detailed information, please refer to [43].

Fig. 8 plots the average classification accuracies of different ensemble methods for different datasets, compared with the *all* strategy which is the classification of the long vector of all features ($R + A + D$ in the previous section). The results show that for different datasets, the best performing ensemble method is different, but it can be selected through a validation step. For this purpose, in each run, the accuracies of all ensemble learning methods are estimated using 4-fold cross validation and the most accurate one is selected for applying on the test data. Fig. 9 compares the best-validated ensemble method for each dataset against its best-performing level combination. In six out of eight cases, the best-validated ensemble learning method outperforms the best-performing level combination. Furthermore, using ensemble methods is more statistically reasonable. Finally, the classification accuracies of DWE-NN are compared with other embedding methods in Table 3. We chose the best-validated ensemble learning method (best-ens) as the combination strategy for DWE-NN. The results using the *all* strategy are also inserted in order to have a fair comparison. The other embedding methods include Laplacian spectrum (Lspec), BackTrackless Walk [44] (BTW), Ihara Zeta Function [45] (IZF), sorted HKSs [46] (HKSsort), histogram of HKSs [46] (HKShist), heat kernel Trace [12] (HIT), and the coefficients of heat content power series [12] (HIP). It can be observed that in half of the datasets, DWE-NN using the *all* strategy, which simply uses the long vector of features of all abstract levels, demonstrated better classification accuracy in comparison with the other tested methods. However DWE-NN using the best validated ensemble method is recommended, as this approach obtained the best

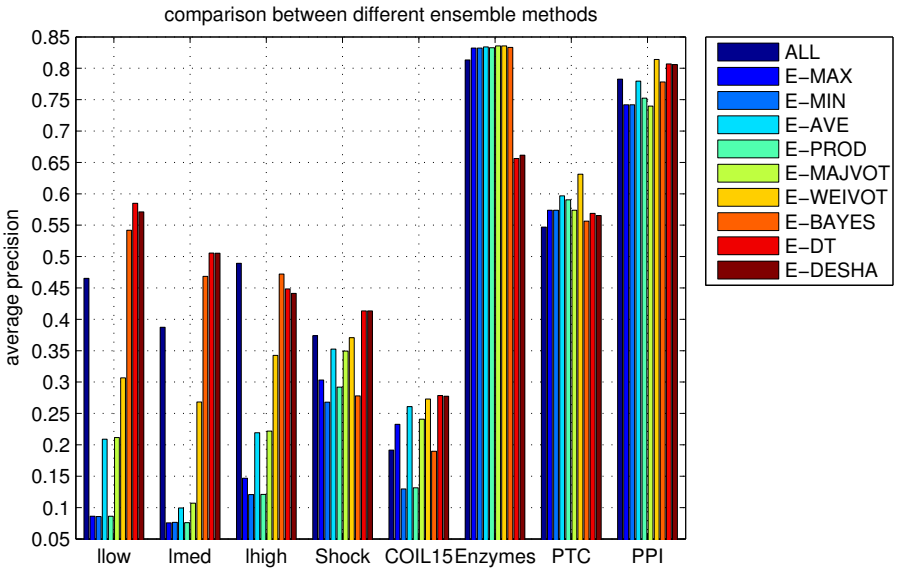


Figure 8: The classification accuracies of applying different ensemble learning methods on different datasets. The base classifier is 5NN.

accuracies in comparison with the other tested embedding methods, for classification of the graphs in all tested datasets. The average classification accuracy improvement for DWE in comparison with the best of the other accuracies in the tested datasets is 6.25%.

Table 3: The classification accuracies of DWE-NN against other embedding methods for all datasets. The best classification accuracy for every dataset is indicated by * symbol. The accuracies of DWE-NN variants which are better than other tested embedding methods are demonstrated in bold face.

	Llow	lmed	lhigh	Shock	coil15	Enzymes	PTC	PPI
Lspec	44.764	34.413	44.364	40.733	26.346	82.283	59.955	70.569
BTW	32.12	24.813	27.022	35.333	27.358	81.05	53.669	66.373
IZF	6.667	6.733	17.871	10	6.788	81.717	44.16	46.536
HKSsort	43.867	35.711	44.48	38.467	22.983	79.517	55.087	71.608
HKShist	6.782	6.667	6.72	8.8	5.798	82.167	45.375	57.105
HIT	12.382	11.742	10.68	7.4	7.605	82.583	49.818	43.673
HIP	13.076	13.356	10.769	16.267	7.604	81.083	51.324	48.333
DWE-NN (best-ens)	**57.271***	**50.547***	**46.276***	**41.667***	**37.103***	**83.05***	**63.629***	**77.634***
DWE-NN (all)	**46.52**	**38.74**	**48.94**	37.40	19.17	81.33	54.69	**78.28**

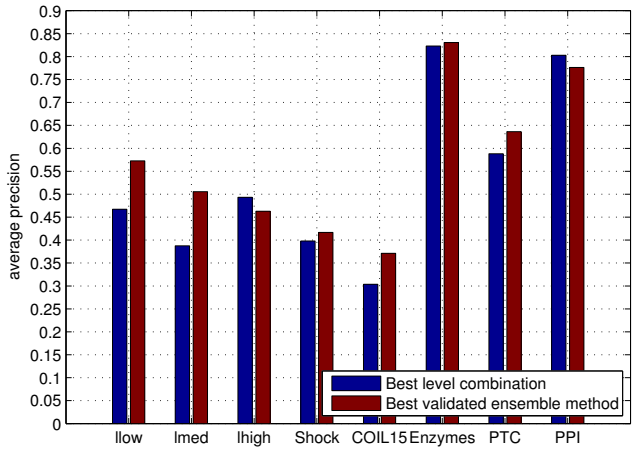


Figure 9: The average classification accuracies of the best-performing level combination and the selected ensemble learning for all datasets.

6. Conclusion

In this article, Diffusion Wavelet Embedding (DWE) is proposed which is a multi-resolution embedding method using diffusion wavelet. This method maps the reference graph to the approximation and detail subspaces of different abstract levels and embeds each of these mappings into vector space using a base embedding method. The abstract graphs of different levels are constructed by applying a diffusion wavelet based summarization method on the abstract graph of the previous level.

The graph summarization using diffusion wavelet is a good option for discovering the inherent node clusters within the graphs. The nearest neighbor approach for super-node construction and the vertex identification for adjacency matrix generation are appropriate operators for graph summarization.

DWE can decrease the cospectrality effects by adding the features of different levels to the base feature set. This method removed the cospectrality effect in three out of five tested sets. DWE uses the information of eigenvectors for mapping the graph into the approximation and detail subspaces at one hand and the eigenvalues to extract the information from these mappings at the other hand. As the eigenvalues and the eigenvectors are two components which are

sufficient for reconstructing the graph uniquely, this method has a good quality for cospectrality reduction.

DWE can improve the classification accuracy in the wide range of applications, using the advantages of diffusion wavelet as a powerful tool for discovering the small and large scale structures within the graph. Five image object detection datasets with different graph extraction methods and three chemical datasets with different graph structures are tested and DWE enhanced their classification accuracies.

The scale of the most informative structures of the graphs differs from one dataset to another. So, different level combinations have different performances on different datasets. Utilizing the ensemble learning methods for combining the information gathered from different levels is a suitable strategy, provided that the more appropriate ensemble method for the special application is applied.

The experimental results suggest that the multi-resolution graph embedding is a promising approach provided that the large scale descriptions of the graphs are extracted precisely. For future work a validation phase for exploring the most informative combination of the abstraction levels can be applied. A validation method should be adopted to set the number of the abstraction levels and the threshold of the diffusion wavelet. The roles of the class number, the graphs size and the noise degree in the ensemble method selection should be verified.

[1] A. K. Debnath, R. L. Lopez de Compadre, G. Debnath, A. J. Shusterman, C. Hansch, Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity, Journal of medicinal chemistry 34 (2) (1991) 786–797.

[2] L. Bai, E. R. Hancock, Depth-based complexity traces of graphs, Pattern Recognition 47 (3) (2014) 1172–1186.

[3] L. Bai, L. Rossi, A. Torsello, E. R. Hancock, A quantum Jensen–Shannon graph kernel for unattributed graphs, Pattern Recognition 48 (2) (2015) 344–355.

[4] A. Lubiw, Some NP-complete problems similar to graph isomorphism, SIAM Journal on Computing 10 (1) (1981) 11–21.

[5] B. Luo, R. C. Wilson, E. R. Hancock, Spectral embedding of graphs, Pattern recognition 36 (10) (2003) 2213–2230.

[6] M. M. Luqman, J.-Y. Ramel, J. Lladós, T. Brouard, Fuzzy multilevel graph embedding, Pattern Recognition 46 (2) (2013) 551–565.

[7] J. Gibert, E. Valveny, H. Bunke, Graph embedding in vector spaces by node attribute statistics, Pattern Recognition 45 (9) (2012) 3072–3083.

[8] K. Riesen, H. Bunke, Classifier ensembles for vector space embedding of graphs, in: International Workshop on Multiple Classifier Systems, Springer, 220–230, 2007.

[9] R. R. Coifman, M. Maggioni, Diffusion wavelets, Applied and Computational Harmonic Analysis 21 (1) (2006) 53–94.

[10] D. M. Cvetković, P. Rowlinson, S. Simic, Eigenspaces of graphs, 66, Cambridge University Press, 1997.

[11] P. Ren, R. C. Wilson, E. R. Hancock, Graph characterization via Ihara coefficients, IEEE Transactions on Neural Networks 22 (2) (2011) 233–245.

[12] B. Xiao, E. R. Hancock, R. C. Wilson, Graph characteristics from the heat kernel trace, Pattern Recognition 42 (11) (2009) 2589–2606.

[13] B. N. Parlett, D. S. Scott, The Lanczos algorithm with selective orthogonalization, Mathematics of computation 33 (145) (1979) 217–238.

[14] R. C. Wilson, P. Zhu, A study of graph spectra for comparing graphs and trees, Pattern Recognition 41 (9) (2008) 2833–2841.

[15] A. J. Schwenk, Almost all trees are cospectral, New directions in the theory of graphs (1973) 275–307.

[16] W. H. Haemers, E. Spence, Enumeration of cospectral graphs, European Journal of Combinatorics 25 (2) (2004) 199–211.

[17] C. D. Godsil, B. McKay, Constructing cospectral graphs, Aequationes Mathematicae 25 (1) (1982) 257–268.

[18] R. C. Wilson, E. R. Hancock, B. Luo, Pattern vectors from algebraic graph theory, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (7) (2005) 1112–1124.

[19] V. Bulitko, N. R. Sturtevant, J. Lu, T. Yau, Graph Abstraction in Real-time Heuristic Search., J. Artif. Intell. Res.(JAIR) 30 (2007) 51–100.

[20] A. Rensink, E. Zambon, Pattern-based graph abstraction, in: International Conference on Graph Transformation, Springer, 66–80, 2012.

[21] A. Torsello, E. R. Hancock, Graph embedding using tree edit-union, Pattern recognition 40 (5) (2007) 1393–1405.

[22] A. Shokoufandeh, D. Macrini, S. Dickinson, K. Siddiqi, S. W. Zucker, Indexing hierarchical structures using graph spectra, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (7) (2005) 1125–1140.

[23] P. Ren, R. C. Wilson, E. R. Hancock, Spectral embedding of feature hypergraphs, in: Joint IAPR International Workshops on Statistical Techniques

in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), Springer, 308–317, 2008.

[24] B. Xiao, A. Torsello, E. R. Hancock, Isotree: Tree clustering via metric embedding, Neurocomputing 71 (10) (2008) 2029–2036.

[25] B. Xiao, E. R. Hancock, R. C. Wilson, Geometric characterization and clustering of graphs using heat kernel embeddings, Image and Vision Computing 28 (6) (2010) 1003–1021.

[26] D. Emms, S. Severini, R. C. Wilson, E. R. Hancock, Coined quantum walks lift the cospectrality of graphs and trees, Pattern Recognition 42 (9) (2009) 1988–2002.

[27] F. Escolano, E. R. Hancock, M. A. Lozano, Birkhoff polytopes, heat kernels and graph complexity, in: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, IEEE, 1–5, 2008.

[28] B. Bonev, F. Escolano, D. Giorgi, S. Biasotti, Information-theoretic selection of high-dimensional spectral features for structural recognition, Computer Vision and Image Understanding 117 (3) (2013) 214–228.

[29] M. Farge, Wavelet transforms and their applications to turbulence, Annual review of fluid mechanics 24 (1) (1992) 395–458.

[30] P. Kumar, E. Foufoula-Georgiou, Wavelet analysis for geophysical applications, Reviews of geophysics 35 (4) (1997) 385–412.

[31] T. Li, Q. Li, S. Zhu, M. Ogihara, A survey on wavelet applications in data mining, ACM SIGKDD Explorations Newsletter 4 (2) (2002) 49–68.

[32] D. Shuman, S. K. Narang, P. Frossard, A. Ortega, P. Vandergheynst, Signal Processing on Graphs, Signal Processing (2/35).

[33] M. Crovella, E. Kolaczyk, Graph wavelets for spatial traffic analysis, in: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE

Computer and Communications. IEEE Societies, vol. 3, IEEE, 1848–1857, 2003.

[34] R. Wagner, V. Delouille, R. Baraniuk, Distributed wavelet de-noising for sensor networks, in: Proceedings of the 45th IEEE Conference on Decision and Control, IEEE, 373–379, 2006.

[35] W. Wang, K. Ramchandran, Random multiresolution representations for arbitrary sensor network graphs, in: 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, vol. 4, IEEE, IV–IV, 2006.

[36] D. K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory, Applied and Computational Harmonic Analysis 30 (2) (2011) 129–150.

[37] S. K. N. Gudivada, Applications of Diffusion Wavelets .

[38] C. Wang, S. Mahadevan, Multiscale Analysis of Document Corpora Based on Diffusion Models., in: IJCAI, 1592–1597, 2009.

[39] H. Bunke, On a relation between graph edit distance and maximum common subgraph, Pattern Recognition Letters 18 (8) (1997) 689–694.

[40] K. Riesen, H. Bunke, IAM graph database repository for graph based pattern recognition and machine learning, in: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), Springer, 287–297, 2008.

[41] G. Li, M. Semerci, B. Yener, M. J. Zaki, Effective graph classification based on topological and label attributes, Statistical Analysis and Data Mining 5 (4) (2012) 265–283.

[42] F. Escolano, E. R. Hancock, M. A. Lozano, Heat diffusion: Thermodynamic depth complexity of networks, Physical Review E 85 (3) (2012) 036206.

[43] L. I. Kuncheva, Combining pattern classifiers: methods and algorithms, John Wiley & Sons, 2004.

[44] F. Aziz, R. C. Wilson, E. R. Hancock, Backtrackless walks on a graph, IEEE transactions on neural networks and learning systems 24 (6) (2013) 977–989.

[45] P. Ren, R. C. Wilson, E. R. Hancock, Pattern vectors from the Ihara zeta function, in: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, IEEE, 1–4, 2008.

[46] R. C. Wilson, Graph Signatures for Evaluating Network Models, in: Pattern Recognition (ICPR), 2014 22nd International Conference on, IEEE, 100–105, 2014.