



Hierarchical graph embedding in vector space by graph pyramid



Seyedeh Fatemeh Mousavi, Mehran Safayani*, Abdolreza Mirzaei, Hoda Bahonar

Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran

ARTICLE INFO

Article history:

Received 14 October 2015

Received in revised form

13 May 2016

Accepted 30 July 2016

Available online 1 August 2016

Keywords:

Graph embedding
Multi-resolution theory
Graph pyramid
Graph summarization
Graph classification

ABSTRACT

Loss of information is the major challenge in graph embedding in vector space which reduces the impact of representational power of graphs in pattern recognition tasks. The objective of this article is to present a hierarchical framework which can decrease this loss in a reasonable computational time. Inspired by multi-resolution ideas in image processing, a graph pyramid is formed based on a selected graph summarization algorithm which can provide the required information for classification. All the pyramid levels or some of them are embedded into a vector through an available embedding method which constructs an informative description containing both local and global features. The experiments are conducted on graphs with numerical and categorical attributes. In the numerical case, a proposed summarization algorithm is applied while in the categorical case, *k*-SNAP graph summarization is applied. The results indicate that this new framework is efficient in terms of accuracy and time consumption in the context of classification problems. It is observed that this improvement is achieved regardless of selected embedding techniques.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

The superior representational power of graphs introduces these data structures as popular replacements for vectors in many pattern recognition applications. Lack of algorithmic tools is one of the main drawbacks of the graph structure. To overcome this drawback, graph embedding in vector space [1–7] provides the opportunity to benefit from the rich repository of vector based methods which are the results of many years of research and experience. However, the problem of finding appropriate vector representations for graphs is not an easy task. Two issues concern embedding methods: first, embedding procedures should not include costly operations and second, extracted features should preserve graph information, the structural details in particular as much as possible. These opposite requirements become more remarkable in larger and more complex graphs. In fact, with growing size of a graph, the cost of extracting useful information from all its components increases. Moreover, the augmented structural complexity raises the possibility of noise and distortion in structure and content of the original graph and this phenomenon makes the feature extraction more critical. As a result, in order to obtain a vectorial representation of a graph, a compromise between the

embedding time and the ability of preserving information must be established, with respect to application. To accomplish this objective, a hierarchical framework is introduced in this article.

Multi-resolution processing is employed successfully in computer vision and image processing algorithms. This method tries to simulate the main characteristics of human visual perception [8]. For this purpose, the applied data structure should be able to represent both the distributed local information of the raw data and the hidden global information within. This structure is named the hierarchical architecture [9] the general scheme of which is shown in Fig. 1. It is observed that this hierarchical structures can be applied to efficiently extract the global information (image interpretations) from the local ones (array of pixel intensities) in a recursive manner. This structure provides an appropriate manner in order to cope with the limitations of memory and time and is considered as a bridge between local information and global interpretation.

In the notion of the hierarchical structure, the graphs are very similar to the images, that is, although the local processing through checking labels and adjacency relations can provide useful information about the content of the graph, accurate understanding of the global graph information requires the processing of substructures in different scales. This processing may be more important in matching and classifying large graphs, since the effective features cannot be detected merely with visual inspection and they can occur in every scale. Depending on the application, the similarities on one or all the scales may be of interest.

* Corresponding author.

E-mail addresses: sf.mousavi@ec.iut.ac.ir (S.F. Mousavi), safayani@cc.iut.ac.ir (M. Safayani), mirzaei@cc.iut.ac.ir (A. Mirzaei), h.bahonar@ec.iut.ac.ir (H. Bahonar).

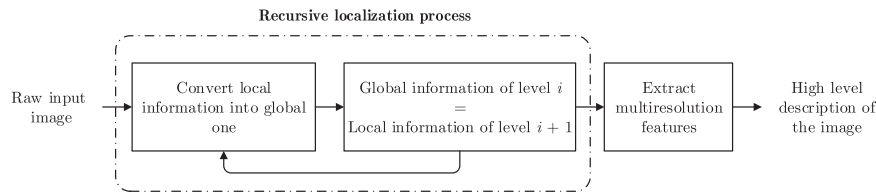


Fig. 1. A general scheme of hierarchical architecture in image processing. A recursive process is run to localize information in lower levels and extract global and informative features from higher levels. The selected features from all or some levels provide a useful description about the input image.

Therefore, the processing at different scales is necessary in the graph domain. Accordingly, transferring the concept of image pyramid to the graph domain can be beneficial.

Based on this idea, this article introduces a framework named *pyramidal graph embedding* (PyrGE) the objective of which is to provide easy access to a variety of information of the graph hierarchy and to apply them in reducing the missing information during embedding. It is expected that this framework increases the classification accuracy by reducing the complexity of managing large graph data. In this approach, a graph summarization algorithm [10,11] is employed as an operator for graph data localization in different scales. The graphs of different levels are embedded into the vectors using the single scale embedding methods and the obtained vectors are combined into a single vector.

In brief, the main contribution of this article is to apply an extracted hierarchical graph set rather than just applying the original graph for embedding in order to extend the representational power of the obtained embedded graph. The proposed framework is generic enough to handle different types of graphs provided that proper graph summarization and embedding methods are adapted to the type of graphs. In order to show this adaption, the experiments are run on three datasets, the first two datasets contain numerical node labels and last one contains categorical node label. For the numerical datasets, a new summarization procedure appropriate for this kind of graphs is proposed while for the categorical dataset, an existing summarization method, *k*-SNAP [11] is applied. The main achievement of this assessment is that regardless of the selected embedding method for each level, embedding procedure through this proposed hierarchical architecture improves the classification accuracy in all experiments in a significant manner.

The outline of this paper is organized as follows. The literature review is presented in Section 2; the primary concepts are discussed in Section 3; the details on how graph pyramid can be applied to enhance the existing embedding methods is expressed in Section 4; the design of several scenarios with respect to this framework is presented in Section 5; the concluding remarks and future works complete this article in Section 6.

2. Literature review

2.1. Graph embedding

The embedding methods can be divided into three groups based on their feature extraction approach. The first group is inspired by dissimilarity representations proposed in [12]. Riesen and Bunke [1] present the vectorial description of a graph by its distances to a number of prototype graphs. The second group is based on the frequencies of appearance of some specific knowledge-dependent substructures, capturing topology information and content of graph through its labels. This group is known as graph probing based methods [13]. For example, in [14] the vector is built by counting the occurrences of non-isomorphic model graphs included in the target graph. The recent approach of Gibert et al. [2] is based on different statistics on the node labels and the

edge relations among them. Luqman et al. [3] consider the graph information in several levels of topology, structure and attributes. The third group, the spectral graph embedding, is a prominent group based on feature extraction from eigen-decomposition of adjacency or Laplacian matrices [15,4]. For example, Ren et al. [5] extracted cycle frequencies as the graph features through the Ihara Zeta function. Aziz et al. [6] applied backtrackless walks to capture the path and cycle information by avoiding tottering.

Each one of these groups has some drawbacks, which become more apparent in the case of large graphs. Dissimilarity-based methods (group 1) are able to handle arbitrary graphs and cope with structural distortions due to the use of graph edit distance as the basic dissimilarity measure. However, its computational cost is a challenge in dealing with large graphs. Probing based methods (group 2) are capable of using specific domain knowledge by finding substructures, but this procedure has high computational cost of subgraph isomorphism. Newer probing-based methods rely more on labels and binary relations and somewhat ignore the complex substructures of relatively large graphs. The spectral methods (group 3) provide intriguing and meaningful properties of a graph with polynomial time complexity, while they have the built-in problem of sensitivity to noise and are restricted to graphs with strongly limited label alphabets.

2.2. Pyramidal models

A wide range of applications such as image segmentation [16], region description and detection [17] make use of pyramid as a useful structure. The image pyramid is a collection of reduced resolution images, which are computed by an iterative procedure [18]. The pyramids can be divided into two categories: the regular pyramids [19] which have some problems [20] due to their fixed decimation process and the irregular pyramids [20] which solve these problems through dynamic graph structures, but their handling is more time consuming due to not specified height of pyramid and size of each level.

Pyramidal models can provide an approach to simulate the mental mechanisms as well as the transformation process from perception to reasoning. This approach can offer sub-optimal algorithms which have the linear computational complexities similar to the mental processes' [21]. Pizlo et al. [22] adopt a multi-level pyramidal model for modeling the size transformation and size perception of human. Haxhimusa et al. [23] applied this approach in order to find near-optimal tour in the traveling salesman problem. They formed a graph where cities were its nodes and the neighboring cities' correlations were its edges. The graph pyramid was built based on the minimum spanning tree principle. The nodes of every level were grouped into trees with a predefined height and applied as the nodes of the next level.

2.3. Graph summarization

Graph summarization methods have a major contribution in resolution reduction of the graphs. The researchers in different disciplines apply different approaches to extract graph summaries. In physics and biology, statistical measures [24] such as degree

distribution are considered as the summaries of the graph. In pattern extraction, the summaries are a set of frequent subgraphs [25]. In community detection, resulting partitions from a node partitioning algorithm are applied as the summaries of the graph [26]. In this study, the focus is on the methods the output of which is the graph.

These graph summarization algorithms can be classified into the structural and the semantical approaches. The structural methods summarize graphs according to the edges' structure and common neighborhoods, for example, the authors of [10] proposed the S-node representation the super-nodes and super-edges of which are the pointers to the partitions of nodes and edges of the given unlabeled web graph, respectively. Navlakha et al. [27] enhanced this summary representation with a set of edge-corrections necessary to regenerate the original graph. In the semantical summarization, attributes and statistics of the nodes and edges are applied, for example, Tian et al. [11] introduced the k -SNAP operation which produces a summary graph of size k by grouping nodes based on preselected node attributes and relationships.

A node clustering based algorithm can be of assistance for summarizing the numerical attributed graphs. Almost all node clustering algorithms apply the affinity matrix in modeling the similarities between nodes. With respect to the processing method of this matrix, the node clustering approaches can be divided into: optimization of graph clustering objectives [28], spectral methods [29], kernel-based methods [30] and multi-level methods [28,31].

3. Concepts and notations

3.1. Graph

Definition 1 (Graph). Let L_V and L_E be a finite or infinite label sets for nodes and edges, respectively. A graph g is a four-tuple $g = (V, E, \mu, \gamma)$, where V is the finite set of nodes, $E \subseteq V \times V$ is a set of edges, $\mu: V \rightarrow L_V$ is the node labeling function and $\gamma: E \rightarrow L_E$ is the edge labeling function.

The graph domain \mathcal{G} is the set of all graphs over the label alphabets L_V and L_E . The values $|V|$ and $|E|$ are the graph order and size, respectively.

3.2. Graph edit distance

Graph edit distance is the most flexible error tolerant graph matching method able to deal with arbitrary graphs and has

obtained robust results against noise.

Definition 2 (Graph Edit Distance). If $g_1 = (V_1, E_1, \mu_1, \gamma_1)$ is the source graph and $g_2 = (V_2, E_2, \mu_2, \gamma_2)$ is the target graph, then the graph edit distance between g_1 and g_2 is defined as

$$d(g_1, g_2) = \min_{(\zeta_1, \dots, \zeta_k) \in \mathcal{I}(g_1, g_2)} \sum_{i=1}^k c(\zeta_i) \quad (1)$$

where $c(\zeta_i)$ is the cost of edit operation ζ_i and $\mathcal{I}(g_1, g_2)$ is the set of edit paths from g_1 to g_2 .

An edit path from g_1 to g_2 is a sequence of edit operations that convert g_1 into g_2 [32]. A standard set of edit operations is defined as follows:

$$Z = \{ins(u), del(v), sbt(u, v), ins(e), del(f), sbt(e, f) \mid u \in V_1, v \in V_2, e \in E_1 \text{ and } f \in E_2\} \quad (2)$$

where, $ins(a)$ and $del(a)$ are the insertion and deletion of node (edge) a , respectively, and $sbt(a, b)$ is the substitution of the node (edge) a with the node (edge) b . Here, edit distance computation is run by the suboptimal method proposed in [33] which has the polynomial complexity $O(n^3)$ in the number of nodes in two graphs.

4. Pyramidal graph embedding

Regardless of applied method, graph embedding in vector space leads to the loss of some graph information due to the lower representational power of the vectors. Moreover, any embedding procedure applied only on a single scale may miss some information at the other scales. Inspired by multi-resolution theory, pyramidal graph embedding seeks to retrieve some of the lost information. In order to accomplish this task, the global information in small scales is localized into the large scales through a summarization algorithm. In this manner, effective global and local features of the given graphs are extracted from different scales simultaneously.

The overall view of pyramidal graph embedding framework is shown in Fig. 2. As observed, this framework consists of two independent subroutines: (1) the graph pyramid extraction subroutine that takes the base level graphs as the input and returns the graph pyramids as the output by applying an adequate localization operator repeatedly and (2) multi-scale embedding subroutine that uses the graph pyramids as the input and embeds the graphs of different levels into the vector space, following by

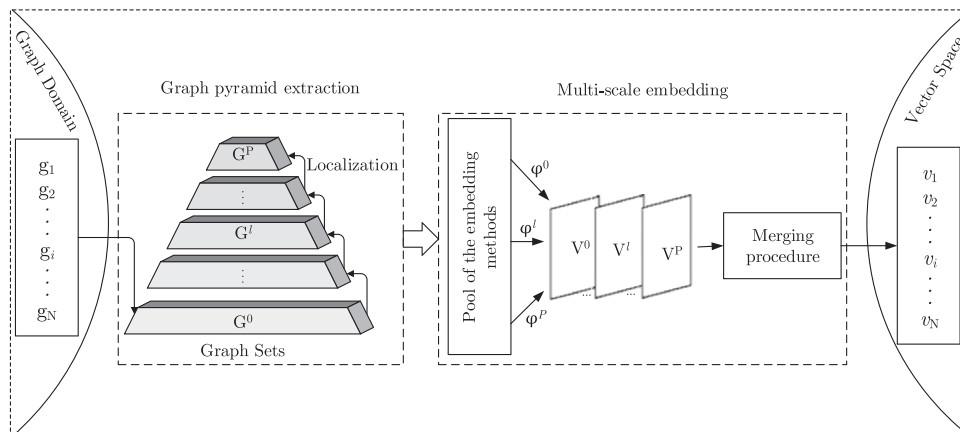


Fig. 2. The outline of pyramidal graph embedding framework. A set of graphs are delivered as the input. For each graph, the framework constructs its graph pyramid, embeds different levels of the pyramid into vector space and returns the merged vector of the graph as the output.

combining the resulting vectors from each graph through the merging procedure and produces the final vectors.

4.1. Graph pyramid

Assume the graph $g = (V, E, \mu, \gamma)$ and its corresponding pyramid as $\Pi_{\lambda, P}(g)$, shown in Fig. 3. There exists a reduced graph of g at every level of $\Pi_{\lambda, P}(g)$. The graph order (here considered as the graph resolution) is reduced with a reduction factor of $\lambda > 1$, by moving towards the next upper layer. Therefore, the number of levels is equal to $L = \lceil \log_{\lambda} |V| \rceil$. The base level (zero level) is the original graph with graph order $|V|$ and the top level (L th level) is the trivial graph with graph order 1. The intermediate level l is $g^l = (V^l, E^l, \mu^l, \gamma^l)$, where $0 \leq l \leq L$ has $|V^l| = \lfloor \frac{|V|}{\lambda^l} \rfloor$ nodes. The graph pyramid limits itself to P ($1 \leq P \leq L$) reduced resolution approximations from the original graph, because usually the high abstract levels like the two top levels displayed in Fig. 3, are not informative enough; Hence, the graph pyramid is truncated to $P+1$ levels such that the highest abstract level is the P th level and the most detailed level is the base level.

As mentioned earlier, the nodes of the graph of level $l+1$ are obtained by applying a reduction procedure on the nodes of level l , that is, a set of nodes in the lower level are grouped into a node in the upper level through a contraction scheme. These vertical

this process. Although finding some specific substructures may be considered as another option to solve this problem, it has high computational complexity of subgraph isomorphism. Likewise, the spectral embedding methods require fewer computations, while they have difficulties in meeting structural distortions. Consequently, graph summarization is an adequate option which can offer a proper view of the given graph with relatively low computational complexity. In the framework, depending upon the kind of graphs, the appropriate summarization procedure should be adopted. The following subsections describe two different graph summarization alternatives which are employed in the experiments.

4.1.1. Numerical attributed graph summarization

An appropriate graph summarization algorithm for numerical node attributed graphs is proposed here (Algorithm 1). This algorithm is a node clustering based approach that has ability to deal with numerical attributes and to control the resolution of summary graphs. In step I, the super-nodes are generated by applying a clustering technique (as the contraction scheme) on the nodes of g^l and considered as the nodes of g^{l+1} . In step II, the super-edges are inserted according to the number of edges between each pair of super-nodes; In step III, the labeling functions of g^{l+1} are defined with respect to the labeling functions of g^l .

Algorithm 1. The suggested graph summarization algorithm.

Input: graph

Input: graph $g^l = (V^l, E^l, \mu^l, \gamma^l)$ and parameters λ , δ_{loop} and δ_{edge} .

Output: graph $g^{l+1} = (V^{l+1}, E^{l+1}, \mu^{l+1}, \gamma^{l+1})$

I. Generate super-nodes

$$k = \left\lfloor \frac{|V^l|}{\lambda} \right\rfloor$$

Define $\mathbf{A} = \text{AFFINITYMATRIX}(g^l)$

Compute $\{C_1, \dots, C_k\} = \text{CLUSTERINGALGORITHM}(g^l, \mathbf{A}, k)$

$V^{l+1} = \{C_1, \dots, C_k\}$

II. Insert super-edges

$E^{l+1} = \{\}$

repeat

Select a non-visited pair $(C_i, C_j) \in V^{l+1} \times V^{l+1}$

if $i = j$ **then** $\delta = \delta_{loop}$ **else** $\delta = \delta_{edge}$.

if $\frac{\sigma_{C_i, C_j}}{\Sigma_{C_i, C_j}} \geq \delta$ **then** $E^{l+1} = E^{l+1} \cup \{(C_i, C_j)\}$.

until there is not any non-visited pair in $V^{l+1} \times V^{l+1}$

III. Define node labeling μ^{l+1} and edge labeling γ^{l+1}

relations are indicated with dashed lines in Fig. 3. The horizontal relations are the edges of the reduced graphs which are formed based on the adjacency relations and inter-node similarities. These relations are shown with straight lines in Fig. 3.

The graph summarizations which allow users to control the resolutions of summaries are suggested as the localization operator for graphs. The summarization algorithm is run on g^l in the l th level and produces the output graph g^{l+1} . The graphs of different levels (g^0 to g^P) are obtained through recursive running of

I. Generate super-nodes: Although any node clustering algorithms can be used to generate super-nodes, spectral methods due to its efficiency and production of high-quality partitions is of interest when eigenvector computation is feasible. Hence, a simple algorithm based on spectral factorization [34] is used, here. The input of this method is an affinity matrix $\mathbf{A} \in \mathbb{R}^{|V^l| \times |V^l|}$ whose entry $a_{uv} = a_{vu}$ encodes the cost associated with cutting the edge between nodes u and v , depending on knowledge and assumptions of the underlying problem. After making \mathbf{A} doubly stochastic, the

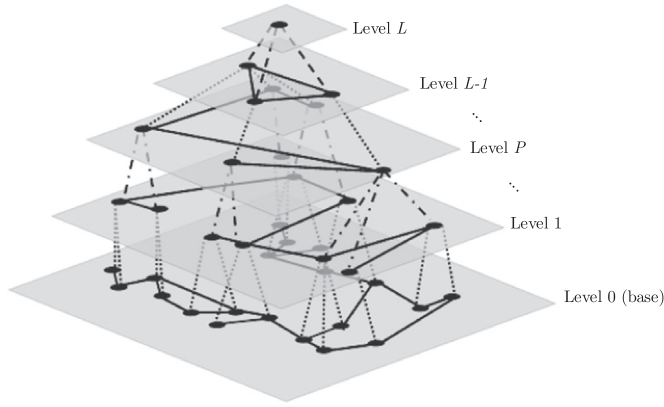


Fig. 3. Graph pyramid.

matrix $\mathbf{U} \in \mathbb{R}^{V \times k}$ is formed by inserting the eigenvectors corresponding to the k largest eigenvalues of \mathbf{A} in its columns. The k -means algorithm is performed on the rows of \mathbf{U} as the representation of the graph nodes to get k partitions as the super-nodes.

II. Insert super-edges: In order to make decision about the structure of the summary graph, a thresholding method is adopted. Assume $\Sigma_{C_i C_j}$ is equal to the number of possible edges between two super-nodes C_i and C_j , and $\sigma_{C_i C_j}$ is the existing edges between them. If the ratio of $\sigma_{C_i C_j}$ to $\Sigma_{C_i C_j}$ is greater than the specific threshold δ , it means that those super-nodes have sufficient connection and the edge insertion between them is reasonable. If $C_i \neq C_j$, $\Sigma_{C_i C_j}$ is the number of edges of a complete bipartite graph between C_i and C_j , and the threshold is called δ_{edge} . If $C_i = C_j$ is the number of edges of a complete graph containing the members of C_i , and the threshold is called δ_{loop} . The δ_{loop} should be greater than δ_{edge} , provided that there is a tendency to keep the simplicity of the graph.

III. Labeling nodes and edges: Various labels can be assigned to the nodes and edges depending on the application and the importance of graph reconstruction. The label of super-node C_i can represent some statistical measures like the number of the internal nodes of C_i and the average and the standard deviation of their labels. The label of the super-edge (C_i, C_j) can either be $\sigma_{C_i C_j}$ or $\sigma_{C_i C_j} / \Sigma_{C_i C_j}$. For the purpose of this article, the average of the node coordinates as the labels of super-nodes would suffice, because the objective here is to obtain a more abstract view of graphs in order to embed them.

4.1.2. Categorical attributed graph summarization

We suggest an existing algorithm called k -SNAP [11] to summarize categorical attributed graphs. Given a graph g^l , the goal of k -SNAP is to produce a summary graph where a node grouping of size k , $\Phi = \{C_1, \dots, C_k\}$, is considered as set of super-nodes. A super-edge is inserted between two super-nodes if there exists at least one edge connecting two nodes of them. k -SNAP tries to find a summary that has best quality based on a proposed Δ -measure which assesses number of differences between the resulted summary with the ideal summary. If $P_{C_j}(C_i) = \{u | u \in C_i \text{ and } \exists v \in C_j \text{ s.t. } (u, v) \in E\}$ and the participation ratio of (C_i, C_j) is defined as $p_{i,j} = \left(\frac{|P_{C_j}(C_i)| + |P_{C_i}(C_j)|}{|C_i| + |C_j|} \right)$, Δ -measure of a grouping Φ is

$$\Delta(\Phi) = \sum_{C_i, C_j \in \Phi} (\delta_{C_j}(C_i) + \delta_{C_i}(C_j)) \quad (3)$$

where

$$\delta_{C_j}(C_i) = \begin{cases} |P_{C_j}(C_i)|, & \text{if } p_{i,j} \geq 0.5 \\ |C_i| - |P_{C_i}(C_j)|, & \text{otherwise.} \end{cases} \quad (4)$$

Finding the optimal summary with the minimum Δ -value is NP-Complete problem, so a heuristic based approach called top-down is used. It begins with a compatible grouping in which all nodes in each group have the same value on preselected attributes. Until the size of grouping is less than k , it iteratively chooses a group C_i which has a maximum value δ with one of its neighbor groups, and then splits it based on its connections with $C_t = \text{argmax}_{C_j} \delta_{C_j}(C_i)$.

4.2. Multi-scale embedding

The last step in pyramidal framework is multi-scale embedding. To obtain the feature vector, a pool of the embedding methods and the subsequent selection strategy could be applied. This selection strategy can be set up in a manner that it picks a single method for all levels, a separate method for each level or a bunch of methods for every level. For example, assume that in an application the graphs are large and noisy; here, high computational methods (e.g. dissimilarity-based) or noise sensitive methods (e.g. spectral) face challenges, and when these graphs move towards the higher levels, the high computational and noise sensitive methods can be adopted due to a reduction in size and error of the graphs. The vector merging procedure combines the resulting vectors of the pyramid levels into a single vector.

In this article, in order to study the effect of the hierarchical framework in improving the existing embedding methods, a single method is adopted for embedding all levels of the graph pyramid and they are combined simply by concatenating them in a long vector where all level combinations are compared in terms of accuracy and time. Meta-parameters of each level are calculated by a validation procedure with respect to other levels of graph pyramid. According to the above descriptions, the formal definition of this proposed framework can be stated as follows:

Definition 3 (Pyramidal graph embedding). Consider that the graph set $G^0 = \{g_1^0, g_2^0, \dots, g_N^0\}$ and the corresponding graph pyramid set $\Pi = \{\Pi_{h,p}(g_1^0), \Pi_{h,p}(g_2^0), \dots, \Pi_{h,p}(g_N^0)\}$. If the base single-scale embedding method applied on graph set $G^l = \{g_1^l, g_2^l, \dots, g_N^l\}$ of Π is

$$\begin{aligned} \varphi^l: G &\rightarrow \mathbb{R}^{k^l} \\ \varphi^l(g_i^l) &= (v_i^l(1), v_i^l(2), \dots, v_i^l(k^l)) \end{aligned} \quad (5)$$

where, $0 \leq l \leq P$, then the pyramidal graph embedding φ_Π is defined as follows:

Table 1

The characteristics of each dataset. Size of the training (tr), validation (va) and test (te) sets, the number of classes (#Cls) and the minimum (min), maximum (max) and average (avg) number of nodes (|V|) and number of edges (|E|).

Dataset	Size (tr, va, te)	#Cls	V (min, max, avg)	E (min, max, avg)
COIL	(360, 75, 150)	15	(18, 79, 42.6)	(45, 228, 116.1)
ODBK	(300, 150, 150)	50	(41, 200, 123.3)	(106, 589, 352.8)
ENZYME	(240, 120, 120)	6	(14, 96, 35.56)	(29, 133, 68.97)

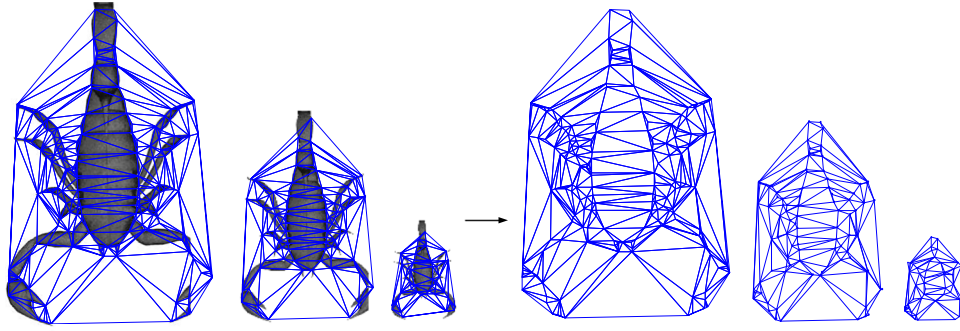


Fig. 4. The global structure extraction of the proposed summarization algorithm. The left side of arrow contains the extracted Delaunay graphs of three levels of Gaussian image pyramid and the right side contains the graphs of summarization algorithm.

$$\begin{aligned} \mathcal{G} &\rightarrow \mathbb{R}^{k^0} \times \mathbb{R}^{k^1} \times \dots \times \mathbb{R}^{k^P} \\ \varphi_{\mathcal{H}}(\mathbf{g}_i^l) &= \left(v_i^0(1), v_i^0(2), \dots, v_i^0(k^0), \dots, v_i^l(1), v_i^l(2), \dots, v_i^l(k^l) \right. \\ &\quad \left. , \dots, v_i^P(1), v_i^P(2), \dots, v_i^P(k^P) \right). \end{aligned} \quad (6)$$

In the following experiments, the strength of this pyramidal framework is verified empirically in order to shed more light on its ability in dealing with graphs of some different domains.

5. Experiments

5.1. Graph datasets

To demonstrate the generality of the framework, experiments are performed on three datasets including Columbia Object Image Library (COIL) [35] and the Object DataBank (ODBK) [36] which are two object detection graph datasets containing numerical labels, and Enzymes dataset [37] with categorical labels consists of the graphs of tertiary structures of proteins. Since the power of this hierarchical framework is more evident in large graphs with a meaningful hierarchy, graphs with appropriate size from these datasets are selected.

The COIL image dataset consists of 128 images of 100 different objects acquired from 72 equally spaced poses. The ODBK image dataset consists of 450 images of 14 views of 209 3D object models, while only 12 views of them are applied here. In order to extract the underlying graphs of the given images, the corners are detected through Harris corner detector [38] and considered as the graph nodes. The edges are established through Delaunay triangulation between these nodes. The obtained graphs are unweighted and the coordinates of the corner points represent the node labels. For these experiments, 15 and 50 classes, with maximum average number of nodes are selected from COIL and ODBK graphs, respectively.

The graphs of Enzymes dataset have one categorical type label on their nodes and their edges are free of labels. By selecting the connected graphs with greater graph orders, a graph dataset of 480 Enzymes, 80 graphs for each class is obtained. The characteristics of all datasets are tabulated in Table 1.

5.2. Numerical attributed graphs

The objective here is twofold: (I) the proposed summarization method is examined to show how this method can favorably preserve the necessary information to classify the graphs, (II) exhibit the power of this framework in enriching the three embedding methods, each selected from either of groups. The objective II is empirically confirmed through k -Nearest Neighbor (k -NN) and Support Vector

Table 2

The classification accuracies (%) of k -NN in graph domain on three abstract levels of COIL and ODBK graphs.

k -NN	Accuracy on COIL			Accuracy on ODBK		
	Original	1st abstract	2nd abstract	Original	1st abstract	2nd abstract
$k=1$	100.00	98.17	87.00	66.67	66.67	60.00
$k=3$	97.00	94.83	81.67	65.33	62.67	55.33
$k=5$	90.00	88.83	78.17	63.33	62.00	53.33

Machine (SVM) classifiers. The classifier k -NN, as a very simple classifier can reveal the improvement in representational power of the resulting vectors in a proper manner. The SVM classifier is applied as one of the most flexible and powerful known classifiers in improving the classification accuracy.

In all experiments, three levels including: original graphs, the first and the second abstract level graphs are embedded ($P=2$). With respect to the applications in this study, the higher level summaries are similar and it is observed that the first three levels convey appropriate information for classification. The reduction factor of the summarization algorithm is $\lambda=2$, inspired from the regular image pyramid. For the topological graphs of the underlying application, the proposed summarization algorithm is used. The affinity matrix is calculated by Gaussian kernel function, where $a_{uv} = \exp(-\|\mu(u) - \mu(v)\|_2^2 / 2\theta^2)$ if $(u, v) \in E^l$ and 0 otherwise. The scaling parameter θ is considered as the standard deviation of distances among all pairs of node labels. The parameter δ depends on the structure of the underlying graphs. Because in these applications the graphs are sparse, i.e., they contain relatively few edges, a small value (≈ 0) of the δ_{edge} is selected here, allowing for insertion of one super-edge between any two super-nodes of a summary graph even if there is an edge between any two nodes. Here, δ_{loop} is set to 0.5, indicating that if the number of edges in a super-node is at least equal to half the number of edges of a complete graph, a loop is inserted on the super-node.

5.2.1. Performance evaluation of summary graphs

The objective of graph summarization algorithm is to preserve the main structures of original graphs as much as possible. To have a better perspective of the representational power, the extracted graphs from the images of Gaussian pyramid are compared with summary graphs obtained from this summarization algorithm in Fig. 4. It is observed that there exists a similarity between the summary graphs discovery of global structures and graphs of the reduced resolution images. The witness of this claim is the similar structures of the corresponding graphs.

To study this effect more precisely, a k -NN classifier is assessed here based on edit distances in graph domain. Regarding the numerical nature of node labels, the Euclidean cost function [1] is

Table 3

The classification accuracies (%) for all states of PyrDGE on COIL and ODBK datasets.

State	COIL					ODBK				
	1-NN	3-NN	5-NN	SVM	GD	1-NN	3-NN	5-NN	SVM	GD
100	95.33	92.67	80.67	98.00	100	65.33	61.33	62.00	62.67	66.67
010	92.00	88.00	84.67	95.33	98.17	62.67	67.33 ⁺	63.33	69.33 ⁺	66.67
001	81.33	81.33	78.00	88.67	87.00	54.00	58.00	53.33	63.33	60.00
011	91.33	85.33	86.00	96.67	–	64.00	67.33 ⁺	63.33	69.33 ⁺	–
101	98.67 *	95.33	88.00	98.00	–	65.33	62.27	64.00	68.00 ⁺	–
110	98.67 *	96.67 *	90.00 *	99.33 **	–	68.00 ⁺	66.67	66.00 *	67.33 ⁺	–
111	98.00	94.00	89.33	98.67	–	68.00 ⁺	64.67	66.00 *	70.67 ** ⁺	–

The statistically significant improvement over DGE method (first row) using a Z-test with significance level $\alpha=0.05$ is displayed in bold face. GD is abbreviated for "Graph Domain" and indicates the best graph domain k -NN classifier among different k 's.

* Specifies the best accuracies of the column.

** Specify the whole experiments related to the specific dataset.

⁺ Indicates the cases that PyrDGE accuracy is superior to the best graph domain classification accuracy.

applied in order to compute the edit distance. The cost for edit operations is set through a validation procedure similar to [1]. However, some heuristics for estimating the initial intervals of costs are applied. As is observed in Table 2, the summarization algorithm has the considerable ability of preserving the required information for classification. It is confirmed by the close and sometimes equal classification accuracies of the summary and original graphs. These results open up the possibility of using the summaries instead of their larger counterparts to avoid time-consuming graph edit distance computations.

5.2.2. Pyramidal graph embedding on single-scale methods

Here, the results of the pyramidal graph embedding application on three selected methods are presented. In all experiments here, the $S = \{001, 010, 100, 011, 101, 110, 111\}$ indicates the set of all possible combinations of the three levels applied in the graph pyramid. At each $s = s_0s_1s_2$ combination, $s_i=1$, if the i th abstract level is available in the final feature vector, and $s_i=0$ otherwise. Obviously, $s=100$ is related to the selected single-scale embedding method.

5.2.3. PyrDGE method

The dissimilarity-based Graph Embedding (DGE) method introduced by Riesen and Bunke [1] is usually adopted as a reference method in comparing with other embedding methods. It is of high accuracy and of course high time complexity as well and can be applied on all kinds of graphs. Here, the Spanning Prototype Selector (SPS) is applied as the prototype selection strategy for this method [39].

In PyrDGE experiments, DGE is applied on all combinations of the pyramidal framework and the classification accuracies of k -NN ($k \in \{1, 3, 5\}$) and SVM are reported in Table 3. In k -NN test, the number of prototypes of each available level should be tuned for every combination. This tuning is made by varying these parameters in the certain intervals and searching for the best configuration regarding the classification accuracy on the validation set simultaneously. In SVM test, for each level of combination, fixing the best performing number of prototypes obtained in k -NN test, the SVM meta-parameters are validated in a separate step. Although this operation yields the sub-optimal results, it is accepted in this study, since the joint task is computationally infeasible. The best validated linear kernel SVM and the best validated RBF kernel SVM are applied on the test set separately and the maximum of the accuracies is reported. In most cases, the maximum accuracy is obtained through RBF kernel, therefore this kernel is suggested for SVM classification.

It is observed that the accuracy improves when the base level is augmented with even a single summary level. This fact implies that the global information extracted from the summary levels can compensate the information loss occurred during the embedding

Table 4

The classification accuracies (%) for all states of PyrNRS on COIL and ODBK datasets.

State	COIL				ODBK			
	1-NN	3-NN	5-NN	SVM	1-NN	3-NN	5-NN	SVM
100	90.67	86.00	82.00	89.33	50.67	49.33	49.33	54.67
010	74.67	67.33	70.67	76.67	50.67	51.33	48.00	58.00
001	52.00	47.33	49.33	72.67	44.00	39.33	40.67	48.67
011	75.33	83.33	67.33	82.67	56.00 *	54.00	52.00 *	62.00
101	94.67	86.67	90.00 *	92.67	52.00	49.33	49.33	55.33
110	93.33	92.00	88.00	92.67	51.33	51.33	48.67	59.33
111	96.00 *	92.67 *	90.00 *	96.67 **	56.00 *	57.33 *	51.33	64.00 **

The statistically significant improvement over NRS method (first row) using a Z-test with significance level $\alpha=0.05$ is displayed in bold face.

* Specifies the best accuracies of the column.

** Specify the whole experiments related to the specific dataset.

of the original graphs to some extent.

The resulting accuracies of $s=011$ combination are superior in comparison with the base level embedding in almost half of the cases. This property is observed in assessing other test embedding methods more or less. Here it is deduced that in these cases, the global information is of substantial importance in classification process. In some methods like DGE where the computation time of edit distance between graphs increases dramatically with an increase in graph order, applying two summary levels instead of the base level can make a tradeoff between accuracy and speed.

To have a better comparison between the accuracies of k -NN applied on graph domain and the obtained vectors of DGE and PyrDGE, the best graph domain accuracies from Table 2 are replicated into Table 3. A surprising point worth noting here is that the best PyrDGE accuracy is close to the graph domain classification accuracy, even higher in some cases. This phenomenon indicates that different aspects of the graph structure extracted by the pyramidal embedding contribute to the preservation of the superior representational power of graphs of the application. Although the loss of some graph information is a built-in characteristic of the graph embedding approach, putting the vectors of different levels together can better reconstruct the graph features than to merely taking advantage from the original graphs.

5.2.4. PyrNRS method

The selected method from the probing based group is the hard version of Node Representative Statistics (NRS) method [2,40]. This method can process attributed-node graphs in the linear time complexity due to just local information usage. In the experiments of NRS and this newly developed PyrNRS, a validation phase for the meta-parameters of k -NN and SVM classifiers is run in a

Table 5

The statistical characteristics of the selected classes of ODBK for evaluation of PyrIZF.









Statistical measures	Classes							
								
AVG. of graph orders	200.00	68.83	137.92	94.00	119.50	75.00	115.17	102.75
STD. of graph orders	0.00	10.70	21.55	19.47	13.16	10.51	6.09	14.63

Table 6

The classification accuracies for all states of PyrIZF and different number of coefficients (NC) on selected classes of ODBK dataset.

State	NC=10				NC=15			
	1-NN	3-NN	5-NN	SVM	1-NN	3-NN	5-NN	SVM
100	45.83	46.67	44.27	48.13	45.73	46.98	43.13	49.27
010	43.65	40.63	42.60	48.13	45.00	40.42	41.56	48.02
001	47.19	48.85*	50.00*	43.54	48.12	48.96*	49.48*	44.17
011	43.44	46.67	44.48	43.54	43.65	47.19	46.25	44.17
101	49.06*	45.83	42.60	54.27**	49.69*	45.94	44.06	54.48**
110	45.94	43.85	41.67	49.69	45.31	43.13	42.60	49.79
111	42.50	41.77	38.54	53.13	41.88	43.44	38.75	53.54

The improvement over IZF method (first row) is displayed in bold face.

* Specifies the best accuracies of the column.

** Specifies the whole experiments related to the specific number of coefficients (NC).

manner similar to that of the previous experiment, except that the set of representatives should be selected instead of the prototypes. The representatives are collected through a simple spanning method on the set of all node labels in the training graphs. The results obtained from all combinations of PyrNRS are tabulated in Table 4.

NRS method takes the graph content and the local information into account by counting the node representatives and the binary relations among them, respectively. Hence, there is a high possibility that two graphs with completely different global structures but similar label content are mapped into the close vectors. The node label relations in different abstract levels of two graphs are compared through PyrNRS and the possibility of this effect is reduced. This claim is justified by the improved precision reported in Table 4.

The authors of [2] suggested the use of DGE in spite of its high computational complexity in cases that the NRS method is of the low classification rates. The results obtained here indicate that PyrNRS can be considered as an appropriate alternative for DGE in such cases. The PyrNRS accuracy is noticeable against DGE (e.g. 64% of PyrNRS in comparison with 62.67% of DGE on SVM classification of ODBK graphs), while the computation time is lower than that of the DGE.

5.2.5. PyrIZF method

The spectral methods are appropriate options in processing the unlabeled graphs which afford an opportunity for assessing the power of pyramidal framework in enhancing the capability of feature vectors in graph structure representation. Ihara Zeta Function (IZF) method [5] applies the reciprocal of the Zeta function, $Z_g^{-1}(u) = \det(I - uT)$, where T is the adjacency matrix of the oriented line graph and I is the identity matrix. The $Z_g^{-1}(u)$ can be represented by a linear combination of the powers of u as $Z_g^{-1}(u) = c_0 + c_1u + \dots + c_{2|E|-1}u^{2|E|-1} + c_{2|E|}u^{2|E|}$. The embedded vector

Table 7

The classification accuracies (%) for all states of PyrDGE on Enzymes dataset.

State	Enzymes			
	1-NN	3-NN	5-NN	SVM
100	52.5	53.33	53.30	63.30
010	46.67	43.31	44.17	50.00
001	37.50	36.67	37.50	40.83
011	46.67	49.17	43.33	50.80
101	58.33	58.33	55.82	63.32
110	57.50	55.83	55.00	66.66
111	60.00*	59.17*	55.84*	67.50**

The statistically significant improvement over DGE method (first row) using a Z-test with significance level $\alpha=0.05$ is displayed in bold face.

* Specifies the best accuracies of the column.

** Specifies the whole experiments.

in this method is computed through the limited number of coefficients c_s because the higher order coefficients contain redundant information. The IZF method collects the global information by considering the number of prime cycles with different lengths, and can be regarded as an embedding method with the global nature.

Our initial experiments indicate that decreasing the classification accuracy by an increase in the number of classes in IZF is more than DGE and NRS. Moreover, the similarity between graph structures of different classes reduces the accuracy significantly. Hence, eight different classes from ODBK dataset with different node distributions exhibited in Table 5 are selected to study the behavior of the pyramidal framework. The graphs are assumed free of labels, in other words the node labels can be considered to have equal value. With this consideration, the calculated affinity matrix by Gaussian kernel function is equated to the adjacency matrix.

To assess PyrIZF, two distinct experiments with 10 and 15 coefficients are run where each level of graph pyramid is embedded into the same length vector space. The classification accuracies are computed similar to that of [6], where, 10-fold cross validation is applied and the averages of 10 runs are reported in Table 6 of this study. The results of k -NN applied on PyrIZF vectors indicate that the structural similarities and dissimilarities are more vivid in the second level of abstraction; however, this phenomenon cannot be predicted in advance. For example, while the accuracy of IZF for eight classes is 48.13%, the same for PyrIZF is 50.00% where only second layer of abstraction is applied. Additionally, findings of some other experiments imply that PyrIZF can decrease the sensitiveness of IZF method to the increasing in the number of classes and between class structural similarities.

5.3. Categorical attributed graphs

The objective here is to show how to adopt embedding and summarization methods for the categorical dataset. In this regard, similar setting of the pyramidal framework to Section 5.2.2 is used,

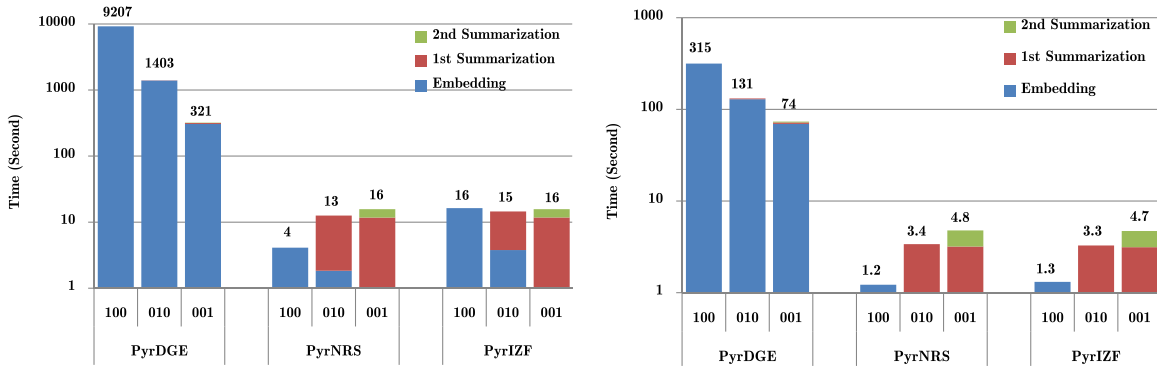


Fig. 5. Run times of the pyramidal framework applying on three single-scale embedding methods. The parameter values are on a logarithmic scale to the basis 10. Left: ODBK dataset; Right: COIL dataset.

but suggested summarization algorithm of Section 4.1.2 (k -SNAP) is considered to deal with this kind of graphs.

The Enzymes dataset has only three distinct labels for a single node attribute. The NRS method, by considering these labels as representatives set, embeds each graph into a short vector based on the frequency of appearance of each representative and the six relations between them [40]. However, this simple representation is not informative enough for differentiating the complex graphs of this dataset. Therefore, NRS is not fit for this application. The IZF method considers graphs without their labels and is not fit for assessing graphs with categorical labels. Hence, the selected embedding method for this experiment is DGE due to its ability in dealing with graphs with different characteristics. The dirac function [1] is used for node substitution cost in computing graph edit distances. The precisions for k -NN classifier with $k=1, 3, 5$ and SVM classifier are tabulated in Table 7. The results indicate that PyrDGE outperforms DGE in the cases that one or more abstract levels are added to the base level.

5.4. Memory requirement and runtime

It is obvious that the required amount of memory to store pyramid nodes is of $O(|V|)$; however, the memory cost is mainly associated with the neighborhood relationships between the nodes; therefore, the maximum required memory in this case is $\left\lfloor \frac{\lambda}{\lambda-1} |V|^2 \right\rfloor$ which can be decreased through the sparse representation of the graphs.

In our implementation, the time complexities of both DGE and IZF methods are of $O(|V|^3)$ and the pyramidal framework does not increase them provided that the time complexity of the employed summarization technique is at most $O(|V|^3)$, which holds true for the proposed summarization algorithm. This statement is not true for NRS method which is of linear time order in the number of nodes and edges, however the extra time is bearable regarding to the remarkable increased accuracy.

The running times for summarization and embedding of different levels of test graphs of ODBK and COIL datasets are presented in Fig. 5. These times are obtained on Intel Core i5 2.5 GHz machine running Windows 8 operating system. In order to validate comparison, a unique configuration of meta-parameters is applied for embedding of all levels. For DGE method, 50 graphs are selected randomly from the training set and the different levels of these graphs are considered as the fixed sets of prototypes. It is observed that there is a notable difference between the embedding time of the base level and two other levels, while the summarization time in all levels is negligible. In ODBK dataset, the embedding time in the third level of pyramid is almost 3.5% of that of the base level. For NRS method, a representative set of size 30 is selected randomly from the range of node labels for embedding each level. It is observed that the summarization procedure increases the overall embedding time of the second and third level in compare to the time of the base level, but these runtimes is still much

less than that of DGE method. For IZF method, 15 Ihara coefficients are utilized for embedding of each level. The runtime is similar to that of PyrNRS and the same discussion can be presented.

6. Conclusions

One of the drawbacks of graph embedding paradigm is the reduction in the informative graph representation strength. To ease this drawback, a hierarchical framework through graph pyramid is proposed which extracts local and global features from different scales of given graph at the same time. In order to confirm the ability of this framework, it is applied on three different embedding methods which have different perspectives on embedding. This framework is applied on the topological graphs of objects as well as the graphs of proteins and the results indicate that it can be beneficial in different graph domains, provided that the appropriate graph embedding and graph summarization methods are selected. Some of the advantages of pyramidal framework in particular are: (I) finding models and sub-structures of lower levels through the low cost analysis of upper levels, (II) reducing the computational complexity through processing at higher levels of abstraction, (III) processing of different resolutions of graph based on application on an independent basis, (IV) validation of the importance of different abstract levels when the structure is not known in advance, and (V) bridging the gap between elementary and more descriptive features.

The objective here is to exhibit the advantages of this proposed framework in terms of accuracy and time in a package. So, experiments are free of any pre-processing and post-processing operations. Feature selection methods, dimension reduction of merged vectors, weighting and rescaling vector components can be beneficial in obtaining more discriminative and shorter vectors. Among all, an approach to learn the number of levels and the importance of each level followed by a weighting procedure will be advantageous. Additionally, different embedding methods can be applied in different graph levels according to the characteristics of the underlying graphs (i.e. the graph order distribution and the amount of noise).

This pyramidal graph embedding can assist the analysis of large graphs like social networks. Of course a tradeoff between the accuracy and time should be established through a subtle assessment.

Acknowledgments

The authors of this paper would like to thank Prof. Wilson for the code of IZF method. The experiments are performed partly in national high performance computing center of Sheikh Bahayi¹ located in

¹ <http://nhpcc.iut.ac.ir>

Isfahan university of technology (IUT) and partly in computing center of department of electrical and computer engineering of IUT.

References

- [1] K. Riesen, H. Bunke, *Graph Classification and Clustering Based on Vector Space Embedding*, World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2010.
- [2] J. Gibert, E. Valveny, H. Bunke, Graph embedding in vector spaces by node attribute statistics, *Pattern Recognit.* 45 (9) (2012) 3072–3083.
- [3] M.M. Luqman, J.-Y. Ramel, J. Lladós, T. Brouard, Fuzzy multilevel graph embedding, *Pattern Recognit.* 46 (2) (2013) 551–565.
- [4] R. Wilson, E. Hancock, B. Luo, Pattern vectors from algebraic graph theory, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (7) (2005) 1112–1124.
- [5] P. Ren, R. Wilson, E. Hancock, Graph characterization via ihara coefficients, *IEEE Trans. Neural Netw.* 22 (2) (2011) 233–245.
- [6] F. Aziz, R. Wilson, E. Hancock, Backtrackless walks on a graph, *IEEE Trans. Neural Netw. Learn. Syst.* 24 (6) (2013) 977–989.
- [7] B. Xiao, E.R. Hancock, R.C. Wilson, Graph characteristics from the heat kernel trace, *Pattern Recognit.* 42 (11) (2009) 2589–2606.
- [8] E.H. Adelson, C.H. Anderson, J.R. Bergen, P.J. Burt, J.M. Ogden, Pyramid methods in image processing, *RCA Eng.* 29 (6) (1984) 33–41.
- [9] J.M. Jolion, A. Rosenfeld, *A Pyramid Framework for Early Vision: Multi-resolutional Computer Vision*, Kluwer Academic Publishers, Norwell, MA, USA, 1994.
- [10] S. Raghavan, H. Garcia-Molina, Representing web graphs, in: 19th International Conference on Data Engineering, 2003, pp. 405–416.
- [11] Y. Tian, R.A. Hankins, J.M. Patel, Efficient aggregation for graph summarization, in: SIGMOD, ACM, New York, NY, USA, 2008, pp. 567–580.
- [12] E. Pekalska, R.P.W. Duin, *The Dissimilarity Representation for Pattern Recognition: Foundations And Applications*, World Scientific, USA, 2005.
- [13] M.M. Luqman, Fuzzy multilevel graph embedding for recognition, indexing and retrieval of graphic document images, Ph.D. thesis, University of Tours, France, 2012.
- [14] N. Sidere, P. Heroux, J.-Y. Ramel, A vectorial representation for the indexation of structural informations, in: *Structural, Syntactic, and Statistical Pattern Recognition*, vol. 5342, Springer, Berlin Heidelberg, 2008, pp. 45–54.
- [15] B. Luo, R.C. Wilson, E.R. Hancock, Spectral embedding of graphs, *Pattern Recognit.* 36 (10) (2003) 2213–2230.
- [16] R. Marfil, L. Molina-Tanco, A. Bandera, F. Sandoval, The construction of bounded irregular pyramids with a union-find decimation process, in: *Graph-Based Representations in Pattern Recognition*, vol. 4538, Springer Berlin Heidelberg, 2007, pp. 307–318.
- [17] R. Vázquez-Martín, R. Marfil, P. Núñez, A. Bandera, F. Sandoval, A novel approach for salient image regions detection and description, *Pattern Recognit. Lett.* 30 (16) (2009) 1464–1476.
- [18] R. Gonzalez, R. Woods, *Digital Image Processing* (3rd Edition), Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.
- [19] P. Burt, T.-H. Hong, A. Rosenfeld, Segmentation and estimation of image region properties through cooperative hierarchical computation, *IEEE Trans. Syst. Man Cybern.* 11 (12) (1981) 802–809.
- [20] R. Marfil, L. Molina-Tanco, A. Bandera, J. Rodríguez, F. Sandoval, Pyramid segmentation algorithms revisited, *Pattern Recognit.* 39 (8) (2006) 1430–1451.
- [21] W.G. Kropatsch, Y. Haxhimusa, Z. Pizlo, G. Lings, Vision pyramids that do not grow too high, *Pattern Recognit. Lett.* 26 (3) (2005) 319–337.
- [22] Z. Pizlo, A. Rosenfeld, J. Epelboim, An exponential pyramid model of the time-course of size processing, *Vis. Res.* 35 (8) (1995) 1089–1107.
- [23] Y. Haxhimusa, W. Kropatsch, Z. Pizlo, A. Ion, A. Lehrbaum, Approximating tsp solution by mst based graph pyramid, in: *Graph-Based Representations in Pattern Recognition*, vol. 4538, Springer Berlin Heidelberg, 2007, pp. 295–306.
- [24] D. Chakrabarti, C. Faloutsos, Graph mining: laws, generators, and algorithms, *ACM Comput. Surv.* 38 (1) (2006), <http://dx.doi.org/10.1145/1132952.1132954>.
- [25] J. Han, J. Pei, Y. Yin, R. Mao, Mining frequent patterns without candidate generation: A frequent-pattern tree approach, *Data Mining Knowl. Discov.* 8 (1) (2004) 53–87.
- [26] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, *Phys. Rev. E* 69 (2004) 026113.
- [27] S. Navlakha, R. Rastogi, N. Shrivastava, Graph summarization with bounded error, in: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD '08*, ACM, 2008, pp. 419–432.
- [28] I.S. Dhillon, Y. Guan, B. Kulis, A unified view of kernel k-means, spectral clustering and graph cuts, University of Texas at Austin, Technical Report UTCS TR-04-25.
- [29] J. Shi, J. Malik, Normalized cuts and image segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (8) (2000) 888–905.
- [30] B. Kulis, S. Basu, I. Dhillon, R. Mooney, Semi-supervised graph clustering: a kernel approach, *Mach. Learn.* 74 (1) (2009) 1–22.
- [31] G. Karypis, V. Kumar, Multilevel k-way partitioning scheme for irregular graphs, *J. Parallel Distrib. Comput.* 48 (1) (1998) 96–129.
- [32] X. Jiang, A. Munger, H. Bunke, A median graphs: properties, algorithms, and applications, *IEEE Trans. Pattern Anal. Mach. Intell.* 23 (10) (2001) 1144–1151.
- [33] K. Riesen, H. Bunke, Approximate graph edit distance computation by means of bipartite graph matching, *Image Vis. Comput.* 27 (7) (2009) 950–959.
- [34] J.P. Hespanha, *grPartition* a MATLAB function for graph partitioning, Available at (<http://www.ece.ucsb.edu/~hespanha>) (october 2004).
- [35] S. Nayar, S. A. Nene, H. Murase, Columbia object image library (coil 100), Department of Computer Science, Columbia University, Technical Report CUCS-006-96.
- [36] M. Tarr, The object databank, (<http://www.cnb.cmu.edu/tarlab/stimuli/objects/index.html>), 2011.
- [37] K.M. Borgwardt, C.S. Ong, S. Schönauer, S.V.N. Vishwanathan, A.J. Smola, H.-P. Kriegel, Protein function prediction via graph kernels, *Bioinformatics* 21 (1) (2005) 47–56.
- [38] C. Harris, M. Stephens, A combined corner and edge detector., in: 4th Alvey vision conference, vol. 15, 1988, pp. 147–151.
- [39] E.Z. Borzeshi, M. Piccardi, K. Riesen, H. Bunke, Discriminative prototype selection methods for graph embedding, *Pattern Recognit.* 46 (6) (2013) 1648–1657.
- [40] J. Gibert, Vector space embedding of graphs via statistics of labeling information, Ph.D. dissertation, Department of Computer Science, University of Alabama at Birmingham (UAB), 2012.

Seyedeh Fatemeh Mousavi was born in Ahvaz. She received her B.Sc. degree with 1st rank in Computer Engineering/Software from Shahid Chamran University of Ahvaz, Iran in 2012. She also received her M.Sc. degree as one of the best students in Computer Engineering/Artificial Intelligence from Isfahan University of Technology, Isfahan, Iran in 2015. Her main area of research interests are pattern recognition machine learning and graph embedding.

Mehran Safayani received his B.S. degree in computer engineering from Isfahan University in 2002. Then, he received the M.Sc. and Ph.D. degrees from Sharif University of Technology in computer architecture and artificial intelligence in 2006 and 2011 respectively. Since 2012, he is an assistant professor of Electrical and Computer Engineering at Isfahan University of Technology. His research interests include statistical pattern recognition, machine learning, soft computing and probabilistic graphical models.

Abdolreza Mirzaei was born in Isfahan, Iran. He received the B.S. (first-class honors) degree in computer engineering from Isfahan University, in 2001, the M.Sc. degree in artificial intelligence from Iran University of Science and Technology, Tehran, Iran, in 2003, and the Ph.D. degree in artificial intelligence from Amirkabir University of Technology, Tehran, in 2009, respectively. He is currently with the Department of Electrical and Computer Engineering, Isfahan University of Technology. His research interests include statistical and structural classification methods, digital image processing, computer vision, multiple classifier systems, and learning methods.

Hoda Bahonar received her B.Sc and M.Sc. in computer engineering from Payamnoor University and Tarbiat Modares University, in 2005 and 2009 respectively. She is currently pursuing her Ph.D. in Computer Engineering at Isfahan University of Technology, Isfahan, Iran. Her current research interests include Statistical and Structural Pattern Recognition and Graph clustering and classification.