Neurocomputing 000 (2017) 1–12

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Semi-supervised multi-graph classification using optimal feature selection and extreme learning machine

Jun Pang[a,b,c,*], Yu Gu[c], Jia Xu[d], Ge Yu[c]

[a] College of Computer Science and Technology, Wuhan University of Science and Technology, Hubei 430065, China
[b] Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Hubei, 430065, China
[c] College of Computer Science and Engineering, Northeastern University, Liaoning 110169, China
[d] School of Computer, Electronics and Information, Guangxi University, Guangxi 530004, China

ARTICLE INFO

Article history:
Received 30 September 2016
Revised 27 December 2016
Accepted 4 January 2017
Available online xxx

Keywords:
Multi-graph
Semi-supervised
Feature selection
Extreme learning machine

ABSTRACT

A multi-graph is represented by a bag of graphs. Semi-supervised multi-graph classification is a partly supervised learning problem, which has a wide range of applications, such as bio-pharmaceutical activity tests, scientific publication categorization and online product recommendation. However, to the best of our knowledge, few research works have be reported. In this paper, we propose a semi-supervised multi-graph classification algorithm to handle the semi-supervised multi-graph classification problem. Our algorithm consists of three main steps, including the optimal subgraph feature selection, the subgraph feature representation of multi-graph and the semi-supervised classifier building. We first propose an evaluation criterion of the optimal subgraph features, which not only considers unlabeled multi-graphs but also considers the constraints between the multi-graph level and the graph level. Then, the optimal subgraph feature selection problem is equivalently converted into the problem of mining m most informative subgraph features. Based on those derived m subgraph features, every multi-graph is represented by an m-dimensional vector, where the ith dimension equals to 1 if at least one graph involved in the multi-graph contains the ith subgraph feature. At last, based on these vectors, semi-supervised extreme learning machine(semi-supervised ELM) is adopted to build the prediction model for predicting the labels of unseen multi-graphs. Extensive experiments on real-world and synthetic graph datasets show that the proposed algorithm is effective and efficient.

1. Introduction

A multi-graph is modeled as a bag of graphs. It is a powerful model to represent complicated structures of objects in physical world. For example, a paper can be represented as a multi-graph, shown as in Fig. 1. A labeled multi-graph is a multi-graph with a class label. If once one of graphs in a multi-graph is labeled as positive, the multi-graph is labeled as positive. Otherwise, the multi-graph is labeled as negative. Multi-graph classification problem aims to learn a prediction model with the aid of labeled multi-graphs and to predict the class labels of those unlabeled multi-graphs, having many practical application, including drug activity detection [1,2], science publication classification [1,2] and product recommendation [1,2]. Two application examples of the multi-graph classification are shown in Example 1 and Example 2, respectively.

Example 1. Given a collection of papers, we use the domain fields of papers, such as Artificial Intelligence, Computer Vision and so on, as the class labels of these papers. The multi-graph classification methods can be used to predict the domain fields of unlabeled papers with these labeled papers [1,2].

Example 2. A molecule has many forms. If one of its forms resists a certain disease, the molecule in this form can be used to manufacture drugs to cure such disease. The specific form of a molecule can be described as a graph. Under such circumstances, a multi-graph represents different forms of the molecule. The multi-graph classification algorithms can be utilized to predict the molecules activities [1,2].

However, because data often are labeled through manual works which are time-consuming with high-cost, it is usually unrealistic to get many labeled multi-graphs in practice, which leads to low classification accuracy. Although it is difficult to get labeled multi-

[*] Corresponding author at: College of Computer Science and Technology, Wuhan University of Science and Technology, Hubei 430065, China.

E-mail addresses: pangjun@wust.edu.cn (J. Pang), guyu@mail.neu.edu.cn (Y. Gu), xujia@gxu.edu.cn (J. Xu), yuge@mail.neu.edu.cn (G. Yu).

http://dx.doi.org/10.1016/j.neucom.2017.01.114
0925-2312/© 2017 Elsevier B.V. All rights reserved.

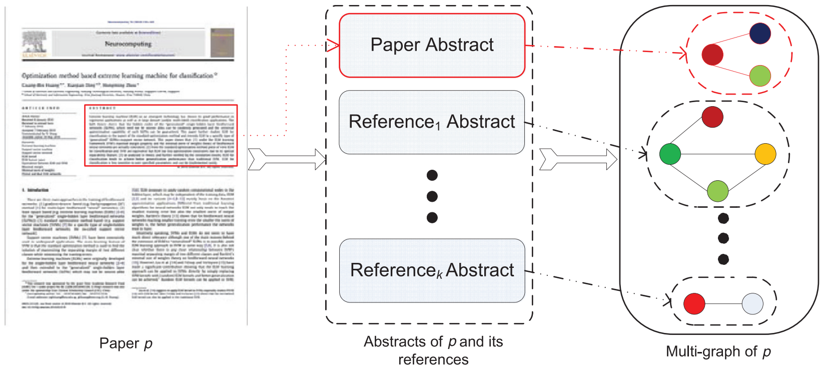


Fig. 1. An example of multi-graph representation model. (Paper p is modeled as a multi-graph, where each graph inside the multi-graph corresponds to the abstract of p or the abstract of the reference cited by p. A graph is constructed by using keywords of the abstract as nodes and their correlations as edges.)

graphs, it is much easier to collect unlabeled multi-graphs. It may be an alternative solution to train the classifier not only by the limited labeled multi-graphs but also by the rich unlabeled multi-graphs. In other words, the semi-supervised learning methods can be used here to improve the classification accuracy of the multi-graph classification problem.

The target of multi-graph semi-supervised learning is to learn a classifier from a small amount of labeled multi-graphs and a large amount of unlabeled multi-graphs to predict the class labels of unlabeled multi-graphs as accurately as possible. Inspired by the methods of semi-supervised subgraph feature selection [3] and multi-graph classification [1,2], we first obtain optimal subgraph features from the multi-graph dataset based on our criterion. Optimal subgraph features are subgraphs, which are most useful to distinguish multi-graphs. Then, every multi-graph is represented by an m-dimensional vector, where the jth dimension equals to 1 if at least one graph involved in the multi-graph contains the jth subgraph feature. Lastly, based on these vectors, an existing semi-supervised algorithm is utilized to build a classifier. After unseen multi-graphs are represented by vectors based on these selected optimal subgraph features, their labels are predicted using the built classifier. It is a non-trivial task to carry out semi-supervised multi-graph classification, mainly facing the following two challenges.

1. Multi-graph partially supervised learning can be divided into two categories [4] i.e., positive and unlabeled learning, and labeled and unlabeled learning (semi-supervised learning). In positive and unlabeled learning, the class labels of all labeled multi-graphs are positive. While, in the second category, the class labels of labeled multi-graphs are not only involve positive labels but also involve other labels. To the best of our knowledge, hardly any works about semi-supervised multi-graph classification have been reported. Wu et al. [5] propose a new learning framework puMGL to solve positive and unlabeled multi-graph classification problem, which obtains higher classification accuracy than the baseline methods. However, puMGL is specially designed to process the datasets which only contain a small number of positive multi-graphs.
2. In order to gain more profits, many practical applications, such as marketing decision and product recommendation, require that the prediction algorithms can ensure a high classification accuracy. However, using existing subgraph feature evaluation criterions [1–3] and the traditional semi-supervised classification models [4] can only achieve a low classification accuracy. Because existing solutions need a large amount of labeled multi-graphs. However, in our problem setting, there are only a small amount of labeled multi-graphs. In addition, existing subgraph feature evaluation criterions for semi-supervised graph classification problem do not consider the constrains of the multi-graph level.

Facing these challenges, we propose an algorithm to solve the semi-supervised multi-graph classification problem. In order to improve the classification accuracy, we propose an novel evaluation criterion to select optimal subgraph features, and adopt semi-supervised ELM to build the prediction model. Compared to the subgraph feature evaluation criterions of supervised multi-graph classification [1,2], our evaluation criterion not only considers labeled multi-graphs but also considers unlabeled multi-graphs. Compared to the subgraph feature evaluation criterion of semi-supervised graph classification [3], our evaluation criterion considers both of the constrains of the graph level and the constrains of the multi-graph level. Moreover, since semi-supervised ELM has a good performance [6]. It is adopted to further improve the classification accuracy in this paper. In addition, a upper bound strategy is derived to improve the efficiency. The major contributions of this paper are summarized as follows.

1. We propose a evaluation criterion of optimal subgraph feature, based on which we design an algorithm to select the optimal subgraph features. In addition, an upper bound pruning strategy is proposed to improve the efficiency of feature selection.
2. We adopt semi-supervised ELM to improve the classification accuracy. Moreover, we propose an algorithm based on our subgraph feature selection algorithm and semi-supervised ELM to solve the semi-supervised multi-graph classification problem.
3. We have conducted extensive experiments on both real and simulated data sets to verify the effectiveness and efficiency of our proposals.

The remainder of this paper is organized as follows. Related works are introduced in Section 2. Problem definitions are discussed in Section 3. The proposed algorithms are provided in Section 4. Experimental results and discussions are presented in Section 5. We conclude the paper in Section 6.

2. Related works

Related works of our study include multi-graph partially supervised learning, subgraph feature selection and semi-supervised extreme learning machine.

2.1. Multi-graph partially supervised learning

Some works about positive and unlabeled multi-graph classification have been reported. Wu et al. [5] propose a learning framework, called puMGL, to solve positive and unlabeled multi-graph classification problem. A subgraph feature selection metric is designed to gain higher accuracy. Inspired by traditional positive and

Please cite this article as: J. Pang et al., Semi-supervised multi-graph classification using optimal feature selection and extreme learning machine, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.01.114

unlabeled classification algorithms, puMGL contains two iterative processes: (1) selecting subgraph features and (2) deriving distinguish model. The problem setting of puMGL is different from our problem, since it supposes that training datasets contain a small quantity of positive and a large number of unlabeled multi-graphs. While, in our works, we suppose that training datasets contain a small quantity of labeled multi-graphs for every category and a large number of unlabeled multi-graphs. Thus, puMGL framework can not be directly applied to solve our problem.

2.2. Subgraph feature selection

Existing subgraph feature selection algorithms can be divided into three types: unsupervised algorithms, semi-supervised algorithms and supervised algorithms.

The first type of algorithms treats frequent subgraphs as subgraph features. For example, the top-*m* frequent subgraphs are used as subgraph features. The second type of algorithms first mines frequent subgraphs, then mines subgraph features from those frequent subgraphs using labeled and unlabeled graph information. For example, Kong and Yu [3] propose the semi-supervised subgraph feature selection algorithm gSSC. Because gSSC not only considers labeled graphs, but also considers unlabeled graphs when selecting subgraph feature, it can derive more valuable subgraph features than other algorithms. Meanwhile, an upper bound pruning strategy is proposed to improve the efficiency of gSSC. The last type of algorithms indirectly or directly mines subgraph features based on many labeled graphs. For example, Yan et al. [7] propose an novel mining framework LEAP(Descending leap mine) to identify the feature subgraphs. Moreover, two new techniques, structural proximity pruning and frequency-descending mining, are proposed to support leap search in graph pattern space. In addition, in order to speed up the process of mining subgraph features and improve the quality of subgraph features, many other techniques are adopted, such as partitioning technique based on similarity [8], fast probing technique based on search history [9] and diversity technique [10].

Although the first type of algorithms do not require the labels of graphs in the graph datasets, experimental results show that its accuracy is far from being satisfied. Because informative subgraph features may not only be frequent subgraphs but also be infrequent ones, it is not appropriate to use only the frequency to select the subgraph features. Although the second type of algorithms only require the graph datasets containing a small number of labeled graphs, they cannot be directly used to solve the semi-supervised multi-graph classification problem because of the differences between the multi-graph and the graph. Compared to the graph, the multi-graph not only maintains the containment relationship between the multi-graph and the graphs, but also has the mutual constraints between the graph label and the multi-graph label. Also, the third type of algorithms are unsuitable for directly solving the semi-supervised multi-graph classification problem because they demand that the training datasets contain a large number of labeled multi-graphs. In this paper, we propose an novel subgraph feature selection algorithm, which is specially designed for the multi-graph setting. Meanwhile, our algorithm considers both of the constrains of the graph level and the constrains of the multi-graph level.

2.3. Semi-supervised extreme learning machine

Huang et al. propose ELM for single hidden-layer feedforward neural networks(SLF-Ns) and then extend it to the “generalized″ SLFNs [11–25]. ELM has better generalization performance, faster learning speed and higher training precision than traditional feedforward neural networks. ELM algorithm has a wide range of applications, such as protein secondary structure prediction [26], classification in P2P networks [27], XML document classification [28], graph classification [29], and multi-graph classification [30].

As a promising algorithm, semi-supervised ELM has begun to attract more researchers’ attentions. Liu et al. [31] introduce the manifold regularization framework into the ELMs model to solve semi-supervised binary classification problem. However, the algorithm is inefficient when the number of hidden neurons is larger than the number of training patterns. Li et al. [32] design a co-training method to repeatedly train ELMs in a semi-supervised setting. The algorithm is inefficient because of its strategy of repeatedly training. Huang et al. [33] extend ELMs for semi-supervised task based on the manifold regularization. The proposed algorithm i.e., SS-ELM (semi-supervised ELM), can handle multi-class classification. Liu et al. [6] propose ESELM(extended semi-supervised ELM) considering the empirical risk and structural risk at the same time. ESELM fits for both low dimension dataset and high dimension dataset. Extensive experimental results show the effectiveness of ESELM. ESELM is inclined to the graph structure while SS-ELM describes the form of semi-supervised method [6]. To our best knowledge, ESELM is the latest semi-supervised ELM algorithm. In this paper, we choose ESELM to build our prediction model.

3. Problem definition

In this section, we introduce related concepts and formulate our problem.

Definition 1. (Connected graph)

A graph G is described by a quaternion $<V, E, \Sigma, f>$, where V and E denote the set of vertices and edges, respectively. Σ indicates the label range of vertices and edges. f is a label function which allocates a label for every vertex and edge. A connected graph is a graph in which there exists at least a path between any two vertices. Under the condition without ambiguity, a connected graph are called a graph for short in this paper. v_i denotes the ith vertex of G and $e(v_i, v_j)$ indicates the edge between v_i and v_j. The label of vertex v_i is represented as $f(v_i)$, the label of edge $e(v_i, v_j)$ is represented as $f(e(v_i, v_j))$, and $l(G)$ denotes the class label of graph G.

Definition 2. (SubGraph)

Graph G' is a subgraph of graph G if G and G' satisfy two conditions: (1) for any vertex $v_i \in V$ of G', $f(v_i) = f(v_j)$ where $v_j \in V$ of G. (2) $f(e(v_i, v_m)) = f(e(v_j, v_n))$, where $e(v_i, v_m) \in E$ of G', $e(v_j, v_n) \in E$ of G and $f(v_i) = f(v_j)$, $f(v_m) = f(v_n)$. $G' \subseteq G$ denotes that G' is a subgraph of G.

Definition 3. (Labeled multi-graph)

A multi-graph is a bag of graphs $MG = \{G_1, \ldots, G_i, \ldots, G_{|MG|}\}(1 < i < |MG|)$. A labeled multi-graph is a multi-graph with binary class label $l(MG) \in \{positive(+), negative(-)\}$. If the class label for one graph of a multi-graph is tagged as positive, the multi-graph is tagged as positive i.e., $\exists G \in MG \ and \ l(G) = positive \Rightarrow l(MG) = positive$. Otherwise, the multi-graph is tagged as negative, i.e., $\forall G \in MG \ and \ l(G) = negative \Rightarrow l(MG) = negative$.

Definition 4. (Optimal subgraph feature selection)

Given a multi-graph set $MG_S = \{MG_1, \ldots, MG_i, \ldots, MG_{|MG_S|}\}$, MG_S’s graph set $G_S = \{G|G \in MG_i, MG_i \in M_S\}$, G_S’s subgraph set $SG = \{Sg|Sg \subseteq G, G \in G_S\}$. Optimal subgraph feature selection aims to find the most valuable subgraph feature set $F \subseteq SG$, which are most useful to distinguish multi-graphs, shown as follows.

$$FS = argmax S(F), \ s.t. |F| = m. \tag{1}$$

where $S(F)$ denotes an evaluation criterion to estimate the usefulness of F, $|F|$ denotes the cardinality of F and m is the maximum number of selected features.

Definition 5. (Subgraph feature representation of graph)

Given a graph set $G_S = \{G_1, \dots, G_i, \dots, G_{|G_S|}\}(1 < i < |G_S|)$, G_S's subgraph set $SG = \{Sg|Sg \subseteq G, G \in G_S\}$ and G_S's subgraph feature set $F = \{Sg_1, \dots, Sg_k, \dots, Sg_m\}(1 < k < m)$, $G_i \in G_S$ is represented as an m dimensions vector $\boldsymbol{X}_i^{G_S} = [(x_i^{Sg_1})^{G_S}, \dots, (x_i^{Sg_k})^{G_S}, \dots, (x_i^{Sg_m})^{G_S}]$. The weight $(x_i^{Sg_k})^{G_S}$ of the kth dimension is 1 *iff* $Sg_k \subseteq G_i$. Otherwise, $(x_i^{Sg_k})^{G_S}$ is 0.

Definition 6. (Subgraph feature representation of multi-graph)

Given a multi-graph set $MG_S = \{MG_1, \dots, MG_j, \dots, MG_{|MG_S|}\}(1 < j < |MG_S|)$, MG_S's graph set $G_S = \{G|G \in MG_j, MG_j \in MG_S\}$, G_S's subgraph set $SG = \{Sg|Sg \subseteq G, G \in G_S\}$ and a subgraph feature set $F = \{Sg_1, \dots, Sg_i, \dots, Sg_m\}$, where $Sg_i \in F (1 \le i \le m)$ denotes a subgraph feature in the set F. $MG_j \in MG_S$ is represented as a vector $\boldsymbol{X}_j^{MG_S} = \{(x_j^{Sg_1})^{MG_S}, \dots, (x_j^{Sg_i})^{MG_S}, \dots, (x_j^{Sg_m})^{MG_S}\}$, where $(x_j^{Sg_i})^{MG_S}=1$ *iff* $\exists G, Sg_i \subseteq G, G \in MG_j$ and $(x_j^{Sg_i})^{MG_S}=0$ otherwise.

Under the condition of no ambiguity, we replace MG_S by M_S in this paper. M_S^u, M_S^+ and M_S^- denote sets consisting of all multi-graphs, whose class labels are unlabeled, positive and negative, respectively. $G_S^{M_S^+} = \{G|G \in G_S,\ G \in M_S^+\}$. $G_S^{M_S^-} = \{G|G \in G_S,\ G \in M_S^-\}$. $|A|$ is the cardinality of the set $A(A = M_S, M_S^u, M_S^+, M_S^-, G_S^{M_S^+}$ or $G_S^{M_S^-})$.

Definition 7. (Semi-supervised multi-graph classification)

Given a multi-graph dataset MG_S containing a small number of labeled multi-graphs for every category and a large amount of unlabeled multi-graphs, we aim to construct prediction model H from labeled and unlabeled multi-graphs to predict unseen multi-graphs with maximum precision.

Next, we enumerate an example of semi-supervised multi-graph classification.

Example 3. Given a training multi-graph dataset consisting of 9 labeled multi-graphs (including 4 positive multi-graphs and 5 negative multi-graphs) and 100 unlabeled multi-graphs. The target of semi-supervised multi-graph classification is to train a classifier based on this training dataset for predicting the class labels of those unlabeled multi-graphs.

In this paper, we focus on solving the semi-supervised multi-graph binary classification problem. However, our algorithm can also be extended to cope with the semi-supervised multi-graph mutli-class classification problem.

4. MGSSL algorithm

In this section, we propose a multi-graph semi-supervised learning algorithm, i.e., MGSSL. At first, we give an the overview of MGSSL. Then, the detailed implementation steps of MGSSL are discussed. Finally, we post the MGSSL algorithm.

4.1. Overview of MGSSL

Given a multi-graph set $MG_S = \{MG_1, \dots, MG_i, \dots, MG_{|MG_S|}\}$ and MG_S's graph set $G_S = \{G_j|G_j \in MG_i, MG_i \in MG_S\}$, the basic idea of MGSSL is as follows: at first, optimal subgraph features F are mined from G; secondly, every $MG_i \in MG_S$ is represented as a vector based on Definition 6; finally, existing semi-supervised learning algorithm, e.g. semi-supervised Bayes (SS-Bayes) [34], semi-supervised SVM(SS-SVM) [35] or ESELM [6], is utilized to build the prediction model. The overview of MGSSL algorithm is shown as Fig. 2. Next, we discuss the proposed subgraph feature evaluation criterion based on labeled and unlabeled multi-graphs, the subgraph feature mining algorithm based on this criterion, an upper bound pruning strategy and the MGSSL algorithm, respectively.

4.2. Optimal subgraph feature selection

In this section, we propose an optimal subgraph feature evaluation criterion and an algorithm to mine optimal subgraph features.

4.2.1. Optimal subgraph feature evaluation criterion

We first discuss the properties that optimal subgraph features should have. Then, based on these properties, we design an optimal subgraph feature evaluation criterion. Lastly, we equivalently convert the optimal subgraph features selection problem into the top-m subgraph features selection problem in order to mine optimal subgraph features.

We assume that the optimal subgraph features should have the following properties.

1. Cannot-link of the multi-graph level: every labeled multi-graph is provided with a known positive or negative class label. Intuitively, the optimal subgraph features should reflect the inconformity of any two multi-graphs with different class labels.
2. Must-link of the multi-graph level: the optimal subgraph features should make sure that any two multi-graphs with the same class label are similar to each other.
3. Separability of the multi-graph level: the class label of every unlabeled multi-graph is unknown. Thus, based on Principle Component Analysis(PCA) principle [36], we select the most discriminative subgraph features to maximally separate any two multi-graphs without class label.
4. Must-link of the graph level: because every graph of a negative multi-graph is labeled as negative, the optimal subgraph features should make sure that any two graphs included in a negative multi-graph are similar to each other.
5. Separability of the graph level: the class labels of graphs contained in a positive multi-graph is unknown. Therefore, based on PCA principle [36], the optimal subgraph features can maximally separate any two graphs involved in a positive multi-graph.

In the above five properties, we not only consider unlabeled multi-graphs but also consider the constraints between the multi-graph level and the graph level. Based on these properties, we design an evaluation criterion $S(F)$ as follow.

$$\begin{aligned} S(F) = {} & \frac{1}{2C}\sum_{k=1}^{m}\sum_{l(MG_i)\times l(MG_j)=-1}((x_i^{G_k})^{M_S} - (x_j^{G_k})^{M_S})^2 \\ & - \frac{1}{2D}\sum_{k=1}^{m}\sum_{l(MG_i)\times l(MG_j)=1}((x_i^{G_k})^{M_S} - (x_j^{G_k})^{M_S})^2 \\ & + \frac{1}{2|M_S^u|^2}\sum_{k=1}^{m}\sum_{\forall MG_i, MG_j \in M_S^u}((x_i^{G_k})^{M_S} - (x_j^{G_k})^{M_S})^2 \\ & - \frac{1}{2|G_S^{M_S^-}|^2}\sum_{k=1}^{m}\sum_{\forall G_i, G_j \in G_S^{M_S^-}}((x_i^{G_k})^{G_S} - (x_j^{G_k})^{G_S})^2 \\ & + \frac{1}{2|G_S^{M_S^+}|^2}\sum_{k=1}^{m}\sum_{\forall G_i, G_j \in G_S^{M_S^+}}((x_i^{G_k})^{G_S} - (x_j^{G_k})^{G_S})^2 \end{aligned} \tag{2}$$

Please cite this article as: J. Pang et al., Semi-supervised multi-graph classification using optimal feature selection and extreme learning machine, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.01.114

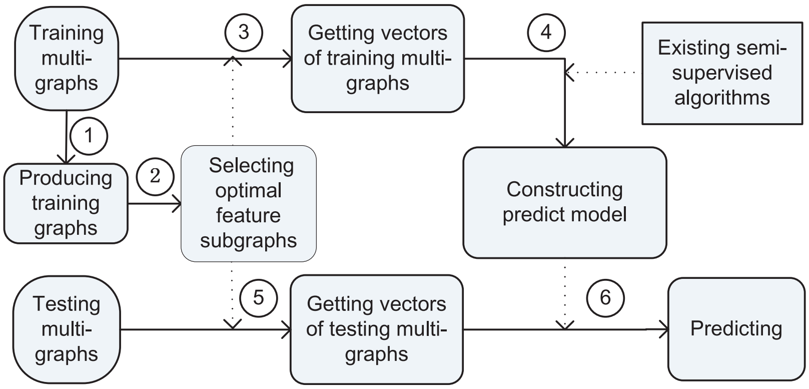

Training multi-graphs
1
Producing training graphs
2
Selecting optimal feature subgraphs
3
Getting vectors of training multi-graphs
4
Existing semi-supervised algorithms
Constructing predict model
Testing multi-graphs
5
Getting vectors of testing multi-graphs
6
Predicting

Fig. 2. An overview of MGSSL.

where $\forall MG_i$, $MG_j \in M_S$, $C = \sum_{l(MG_i)\times l(MG_j)=-1} 1$, $D = \sum_{l(MG_i)\times l(MG_j)=1} 1$. The above five lines of expressions are corresponding to the five properties in turn.

By defining two auxiliary matrixes [3], i.e., $W_{M_S} = [w_{ij}^{M_S}]^{|M_S|\times|M_S|}$, and $W_{G_S} = [w_{ij}^{G_S}]^{|G_S|\times|G_S|}$, as follows.

$$w_{ij}^{M_S} = \begin{cases} \frac{1}{C} & l(MG_i)\times l(MG_j)\text{=-1} \\ -\frac{1}{D} & l(MG_i)\times l(MG_j)\text{=1} \\ \frac{1}{|M_S^u|^2} & MG_i, MG_j \in M_S^u \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$w_{ij}^{G_S} = \begin{cases} -\frac{1}{|G_S^{M_S^-}|^2} & G_i, G_j \in G_S^{M_S^-} \\ \frac{1}{|G_S^{M_S^+}|^2} & G_i, G_j \in G_S^{M_S^+} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Eq. (2) can been rewritten as follows.

$$\begin{aligned} S(F) &= \frac{1}{2}\sum_{k=1}^{m}\sum_{\forall MG_i, MG_j \in M_S}((x_i^{G_k})^{M_S} - (x_j^{G_k})^{M_S})^2 w_{ij}^{M_S} \\ &\quad + \frac{1}{2}\sum_{k=1}^{m}\sum_{\forall G_i, G_j \in G_S}((x_i^{G_k})^{G_S} - (x_j^{G_k})^{G_S})^2 w_{ij}^{G_S} \\ &= S(F)^{M_S} + S(F)^{G_S} \end{aligned}$$

$$where\ S(F)^{M_S} = \frac{1}{2}\sum_{k=1}^{m}\sum_{\forall MG_i, MG_j \in M_S}((x_i^{G_k})^{M_S} - (x_j^{G_k})^{M_S})^2 w_{ij}^{M_S},\ and$$

$$S(F)^{G_S} = \frac{1}{2}\sum_{k=1}^{m}\sum_{\forall G_i, G_j \in G_S}((x_i^{G_k})^{G_S} - (x_j^{G_k})^{G_S})^2 w_{ij}^{G_S} \tag{5}$$

For bag level evaluation $S(F)^{M_S}$, we have

$$\begin{aligned} S(F)^{M_S} &= \frac{1}{2}\sum_{k=1}^{m}\sum_{\forall MG_i, MG_j \in M_S}((x_i^{G_k})^{M_S} - (x_j^{G_k})^{M_S})^2 w_{ij}^{M_S} \\ &= \sum_{k=1}^{m}\sum_{\forall MG_i, MG_j \in M_S}(((x_i^{G_k})^{M_S})^2 w_{ij}^{M_S} - (x_i^{G_k})^{M_S}(x_j^{G_k})^{M_S} w_{ij}^{M_S}) \\ &= \sum_{k=1}^{m}((\boldsymbol{f}_{G_k}^{M_S})^T D_{M_S}\boldsymbol{f}_{G_k}^{M_S} - (\boldsymbol{f}_{G_k}^{M_S})^T W_{M_S}\boldsymbol{f}_{G_k}^{M_S}) \\ &= \sum_{k=1}^{m}(\boldsymbol{f}_{G_k}^{M_S})^T L_{M_S}\boldsymbol{f}_{G_k}^{M_S} \end{aligned} \tag{6}$$

where D_{M_S} is a diagonal matrix, with the diagonal element $d_{ii}^{M_S} = \sum_{j=1}^{|M_S|} w_{ij}^{M_S}$. $L_{M_S} = D_{M_S} - W_{M_S}$ is a Laplacian matrix. Vector $\boldsymbol{f}_{G_k}^{M_S} = [f_{G_k}^{MG_1}, f_{G_k}^{MG_2}, \dots, f_{G_k}^{MG_{|M_S|}}]^T$ denotes whether the subgraph G_k is contained by one graph of every multi-graph MG_i. If one graph of MG_i contains G_k, the value of variable $f_{G_k}^{MG_i}$ is set to 1; otherwise, the value of variable $f_{G_k}^{MG_i}$ is set to 0.

In a similar way, the graph level evaluation $S(F)^{M_S}$ is rewritten as follow:

$$S(F)^{G_S} = \sum_{k=1}^{m}(\boldsymbol{f}_{G_k}^{G_S})^T L_{G_S}\boldsymbol{f}_{G_k}^{G_S} \tag{7}$$

where D_{G_S} is a diagonal matrix, the diagonal element $d_{ii}^{G_S} = \sum_{j=1}^{|G_S|} w_{ij}^{G_S}$. $L_{G_S} = D_{G_S} - W_{G_S}$ is a Laplacian matrix. Vector $\boldsymbol{f}_{G_k}^{G_S} = [f_{G_k}^{G_1}, f_{G_k}^{G_2}, \dots, f_{G_k}^{G_{|G_S|}}]^T$ denotes whether subgraph $G_k \in F$ is contained by every graph $G_i \in G_S$. If G_i contains G_k, the corresponding weight is set to 1; otherwise, the corresponding weight is set to 0.

Thus, based on Eqs. 6 and 7, Eq. 2 is rewritten as

$$\begin{aligned} S(F) &= S(F)^{M_S} + S(F)^{G_S} \\ &= \sum_{k=1}^{m}(\boldsymbol{f}_{G_k}^{M_S})^T L_{M_S}\boldsymbol{f}_{G_k}^{M_S} + \sum_{k=1}^{m}(\boldsymbol{f}_{G_k}^{G_S})^T L_{G_S}\boldsymbol{f}_{G_k}^{G_S} \\ &= \sum_{k=1}^{m}(\boldsymbol{f}_{G_k})^T L\boldsymbol{f}_{G_k}. \end{aligned}$$

$$where,\ f_{G_k} = \begin{bmatrix} \boldsymbol{f}_{G_k}^{M_S} \\ \boldsymbol{f}_{G_k}^{G_S} \end{bmatrix},\ L = \begin{bmatrix} L_{M_S} & 0_{|M_S|\times|G_S|} \\ 0_{|G_S|\times|M_S|} & L_{G_S} \end{bmatrix} \tag{8}$$

By denoting function $H(G_k, L) = (f_{G_k})^T L f_{G_k}$, the optimal subgraph feature selection problem in Eq. (1) is equivalent to finding top m subgraphs, which can maximize the $H(G_k,\ L)$, i.e., $\{G_k | H(G_k, L) \geq H(G_j, L), 1 \leq k \leq m\ and\ m+1 \leq j \leq |SG|\}$. $|SG|$ is the total number of subgraphs contained in the multi-graph dataset M_S. Thus, Eq. (1) can be rewritten as

$$FS = \max_{F}\sum_{G_k \in F} H(G_k, L)\ s.t. |F| \leq m \tag{9}$$

Definition 8 (Value of a single subgraph)**.** Suppose W_{M_S} and W_{G_S} are two auxiliary matrixes defined as Eq. (3) and Eq. (4), respectively. L_{M_S} is a Laplacian matrix defined as $L_{M_S} = D_{M_S} - W_{M_S}$, where D_{M_S} is a diagonal matrix with $d_{ii}^{M_S} = \sum_{j=1}^{|M_S|} w_{ij}^{M_S}$. Similarly, L_{G_S} can be derived. Next, we define L(Eq. (8)), which is a Laplacian matrix consisting of L_{M_S} and L_{G_S}. We define a quality criterion of a subgraph G_k, called $sgScore(G_k)$, in Eq. (10).

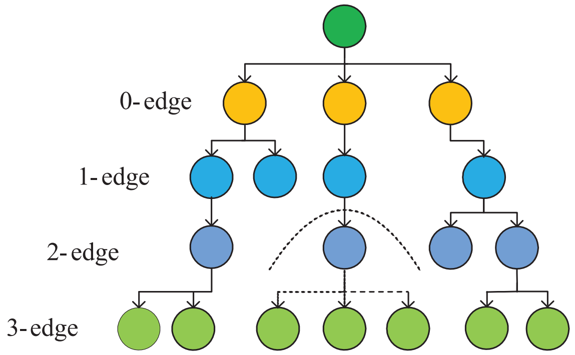


Fig. 3. Illustration diagram of DFS code tree.

$$sgScore(G_k) = H(G_k, L) = (f_{G_k})^T L f_{G_k}$$
$$where\ f_{G_k} = \begin{bmatrix} \boldsymbol{f}_{G_k}^{M_S} \\ \boldsymbol{f}_{G_k}^{G_S} \end{bmatrix}, \boldsymbol{f}_{G_k}^{M_S} (or\ \boldsymbol{f}_{G_k}^{G_S})\ is\ denoted\ as\ Eq.\ 6\ (or\ Eq.\ 7). \tag{10}$$

Obviously, the optimal subgraph feature selection problem denoted in Eq. (1) is equivalent to find m subgraphs, whose value score $sgScore()$ are bigger than ones of other subgraphs in SG(SG is denoted in Definition 4). Suppose the value scores of all subgraphs are indicated as $sgScore(Sg_1) \geq sgScore(Sg_2) \geq \cdots \geq sgScore(Sg_{|SG|})$ in a descending order. The most valuable subgraph feature set F is consisting of Sg_1, ...,Sg_i, ..., and $Sg_m(2 < i < m)$.

4.2.2. Optimal subgraph feature selection algorithm

In this section, we first introduce the basic idea of our optimal subgraph feature selection algorithm. Then, in order to improve the efficiency, we propose a upper bound pruning strategy whose correctness is proved. After that, we propose an algorithm of the optimal subgraph feature selection(OSFS) based on this upper bound pruning strategy. Next, we introduce the basic idea of OSFS which is similar with gSSC [3].

The basic idea of OSFS is shown below. A subgraph is represented as a depth-first search(DFS) code. Therefore, all subgraphs of the multi-graph dataset constitute a DFS code tree shown as Fig. 3. Every node denotes a subgraph with a corresponding DFS code. Different DFS codes may represent the same subgraph. However, a subgraph only has an unique minimum DFS code. If the minimum DFS codes of two subgraphs are the same, the two subgraphs are isomorphic [3]. OSFS starts to traverse the DFS code tree from the root node using the Depth First Search strategy(DFS). Based on Eq. (10), the value of the current node is calculated. If the calculated value is bigger than the minimum value of the current top-m subgraphs, we substitute the subgraph with the minimum value. Otherwise, we move to execute the next step. Then, the proposed upper bound is utilized to prune these non-final-result subgraphs. If the maximum value of all child nodes of the current node is smaller than the minimum value of the current top-m subgraphs, the whole child nodes are pruned safely. Next, we propose the upper bound of the subgraph feature value, then prove its correctness.

Theorem 1. *Given two subgraphs G and G', where $G \subseteq G'$, the upper bound of the value for G' is $\hat{S}(G)$, i.e., $S(G') \leq \hat{S}(G)$, where* $\hat{S}(G) = (f_G)^T \hat{L} f_G,\ \hat{L} = \begin{bmatrix} \hat{L}_{M_S} & 0_{|M_S|\times|G_S|} \\ 0_{G_S\times|M_S|} & \hat{L}_{G_S} \end{bmatrix},\ \hat{L}_{ij}^{M_S} = max(0, L_{ij}^{M_S}),\ and\ \hat{L}_{ij}^{G_S} = max(0, L_{ij}^{G_S})$.

Proof. To prove Theorem 1 is equivalent to prove the inequality, i.e., $S(G') \leq \hat{S}(G)$. According to Eq. (10), the following equation is established:

$$\begin{aligned} S(G') &= (f_{G'})^T L f_{G'} \\ &= \begin{bmatrix} (\boldsymbol{f}_{G'}^{M_S})^T & (\boldsymbol{f}_{G'}^{G_S})^T \end{bmatrix} \times \begin{bmatrix} L_{M_S} & 0_{|M_S|\times|G_S|} \\ 0_{|G_S|\times|M_S|} & L_{G_S} \end{bmatrix} \\ &\quad \times \begin{bmatrix} \boldsymbol{f}_{G'}^{M_S} \\ \boldsymbol{f}_{G'}^{G_S} \end{bmatrix} \\ &= (\boldsymbol{f}_{G'}^{M_S})^T L_{M_S} \boldsymbol{f}_{G'}^{M_S} + (\boldsymbol{f}_{G'}^{G_S})^T L_{G_S} \boldsymbol{f}_{G'}^{G_S} \\ &= \sum_{i,j:MG_i,MG_j \in M_S(G')} L_{ij}^{M_S} + \sum_{i,j:G_i,G_j \in G_S(G')} L_{ij}^{G_S} \end{aligned}$$
$$where\ M_S(G') = \{MG | \forall MG \in M_S\ \exists G \in MG, G' \subseteq G\},\ and$$
$$G_S(G') = \{G | \forall G \in G_S, G' \subseteq G\}. \tag{11}$$

$$\because \hat{L}_{ij}^{M_S} = max(0, L_{ij}^{M_S}),\ \hat{L}_{ij}^{G_S} = max(0, L_{ij}^{G_S})$$

$$\therefore \hat{L}_{ij}^{M_S} \geq 0,\ \hat{L}_{ij}^{G_S} \geq 0$$

$$\begin{aligned} \therefore & \sum_{i,j:MG_i,MG_j \in M_S(G')} L_{ij}^{M_S} + \sum_{i,j:G_i,G_j \in G_S(G')} L_{ij}^{G_S} \\ \leq & \sum_{i,j:MG_i,MG_j \in M_S(G')} \hat{L}_{ij}^{M_S} + \sum_{i,j:G_i,G_j \in G_S(G')} \hat{L}_{ij}^{G_S} \end{aligned} \tag{12}$$

$$\because G \subseteq G',\ \hat{L}_{ij}^{M_S} \geq 0\ and\ \hat{L}_{ij}^{G_S} \geq 0$$

$$\therefore M_S(G') \subseteq M_S(G),\ G_S(G') \subseteq G_S(G)$$

$$\begin{aligned} \therefore & \sum_{i,j:MG_i,MG_j \in M_S(G')} \hat{L}_{ij}^{M_S} + \sum_{i,j:G_i,G_j \in G_S(G')} \hat{L}_{ij}^{G_S} \\ \leq & \sum_{i,j:MG_i,MG_j \in M_S(G)} \hat{L}_{ij}^{M_S} + \sum_{i,j:G_i,G_j \in G_S(G)} \hat{L}_{ij}^{G_S} \\ = & (\boldsymbol{f}_G^{M_S})^T \hat{L}_{M_S} \boldsymbol{f}_G^{M_S} + (\boldsymbol{f}_G^{G_S})^T \hat{L}_{G_S} \boldsymbol{f}_G^{G_S} \\ = & \begin{bmatrix} (\boldsymbol{f}_G^{M_S})^T & (\boldsymbol{f}_G^{G_S})^T \end{bmatrix} \times \begin{bmatrix} \hat{L}_{M_S} & 0_{|M_S|\times|G_S|} \\ 0_{G_S\times|M_S|} & \hat{L}_{G_S} \end{bmatrix} \times \begin{bmatrix} \boldsymbol{f}_G^{M_S} \\ \boldsymbol{f}_G^{G_S} \end{bmatrix} \\ = & (f_G)^T \hat{L} f_G \\ = & \hat{S}(G) \end{aligned} \tag{13}$$

In conclusion, $S(G') \leq S(\hat{G})$. Thus, Theorem 1 holds. □

In the process of the depth-first search, if $\hat{S}(G)$ of the current node G is not more than the minimum value of the current optimal subgraph features, the sub nodes of G can be pruned safely according to Theorem 1. Otherwise, they cannot be removed.

In order to improve the efficiency, the threshold pruning strategy is adopted. If the frequency of the current node, i.e., the upper bound of the frequency of all child nodes for the current node, is less than the frequency threshold, the whole child nodes are pruned. After the whole DFS code tree is traversed, the calculated top-m subgraphs are the expected optimal subgraph features.

Based on the upper bound and the threshold pruning strategies, the pseudo-code of the optimal subgraph feature selection algorithm, i.e., OSFS, is given in Algorithm 1. The details of functions SubgraphMining() and Enumerate() in Algorithm 1 are listed in Algorithm 2 and Algorithm 3, respectively. We first select frequent 0-edge and 1-edge subgraphs, which own 0 edge or 1 edge and are treated as the candidate seeds (lines 1–7). Then, we set every candidate seed as a start node and conduct the deep-first search. At last, the process will not be finished until every candidate seed subgraph is successfully traversed. The subgraphs in set

Please cite this article as: J. Pang et al., Semi-supervised multi-graph classification using optimal feature selection and extreme learning machine, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.01.114

Algorithm 1. OSFS.

```
    Input  : Multi-graph dataset $MG_S$, the support threshold $\beta$, the number of
             subgraph features $m$.
    Output: an optimal feature subgraph set $F = \{F_1, F_2, ..., F_m\}$
 1  Construct graph set $G_S = \{G_i | G_i \in MG_j, MG_j \in MG_S\}$, $G_S^+ = \{G_i | G_i \in MG_S^+\}$ and $G_S^- = \{G_i | G_i \in MG_S^-\}$;
 2  $R = \varnothing$, set the minimum value of the current optimal feature subgraphs $\alpha$ to $-\infty$;
 3  Sort nodes and edges in $G_S$ according the frequency;
 4  Remove infrequent nodes and edges;
 5  Relabel all remaining nodes and edges according to the order;
 6  $R^1 = \{G_{e=1} | support(G_{e=1}) \geq \beta\}$;
 7  Sort $G_{e=1} \in R^1$ according to the DFS dictionary;
 8  $R = R^1$, update $\alpha$;
 9  for $e \in R^1$ do
10      Initialize $s = e$, set $s.GS = \{G | \forall G \in G_S, e \in E(G)\}$($s.GS$ only records the
        ids of the graphs);
11      Calculate $S(s)$ and $\hat{S}(s)$;
12      SubGraphMining($G_S, R, s, S(s), \hat{S}(s), \alpha$);
13      $G_S = G_S - e$;
14      if $|G_S| < \alpha$ then
15          break;
```

Algorithm 2. SubGraphMining().

```
    Input  : A graph set $G_S$, result set $R$, $s$, $S(s)$, $\hat{S}(s)$ and $\alpha$.
    Output: The feature subgraph set $R$ starting at $s$.
 1  if $s \neq$ the minimum DFS code of $s$ then
 2      return;
 3  if $|R| < m$ then
 4      $R = R \cup \{s\}$, update $\alpha$;
 5  else
 6      if $S(s) > \alpha$ then
 7          Remove responding subgraphs of $\alpha$ in R;
 8          $R = R \cup \{s\}$, update $\alpha$;
 9  if $\hat{S}(s) > \alpha$ then
10      $s' = s + e$;
11      Enumerate($s$);
12      for $\forall c \in s'$ do
13          Calculate $S(c)$ and $\hat{S}(c)$;
14          if $Support(c) \geq \beta$ and $\hat{S}(c) \geq \alpha$ then
15              $s$=$c$;
16              SubGraphMining($G_S$, $R$, $s$, $S(s)$, $\hat{S}(s)$, $\alpha$);
```

Algorithm 3. Enumerate().

```
    Input  : A subgraph $s$.
    Output: The supergraphs of $s$ which has one more edge than $s$
 1  for $\forall G \in s.GS$ do
 2      Enumerate all embedding of $s$ in $G$;
 3      for $\forall c \supset s, c \subseteq G$ do
 4          $c.GS = c.GS \cup \{G\}$;
 5      if $G$ overs all subnodes of $s$ then
 6          break;
```

R are the desired optimal subgraph features (lines 8–11). In order to improve the efficiency, we adopt the following strategies: (1) in the process of the deep-first search, the frequency monotonicity [37] and Theorem 1 are utilized to reduce the search space(lines 8–13 in Algorithm 2); (2) the size of G_S is reduced in the condition of not affecting final results(line 12); (3) the cardinality of the surplus G_S is utilized to conduct pruning(lines 13–14); (4) we detect minimum DFS code to avoid the repeated extension of the subgraph [37].

Algorithm 4. MGV.

```
    Input  : The optimal feature subgraph $F = \{Sg_1,\ Sg_2,\ ...,\ Sg_m\}$ and testing
             multi-graphs $MG_T = \{MG_1, ..., MG_i\}$.
    Output: Vectors of testing multi-graphs $X_T = \{\boldsymbol{X}_{t1}^{M_S}, \boldsymbol{X}_{t2}^{M_S}, ..., \boldsymbol{X}_{ti}^{M_S}\}$.
 1  $X_T = \varnothing$;
 2  for $MG_j \in MG_T$ do
 3      Definite the according vector $\boldsymbol{X}_j^{M_S}$ of $MG_j$;
 4      $\boldsymbol{X}_j^{M_S} = GV(F,\ MG)$;
 5      $X_T = X_T \cup \{\boldsymbol{X}_j^{M_S}\}$
```

Algorithm 5. GV().

```
    Input  : The optimal feature subgraph $F = \{Sg_1,\ Sg_2,\ ...,\ Sg_m\}$ and a
             multi-graph $MG = \{G_1, ..., G_{|MG|}\}$.
    Output: The feature vector $\boldsymbol{X}_{MG}^{M_S} = \{(x^{Sg_1})^{M_S}, (x^{Sg_2})^{M_S}, ..., (x^{Sg_m})^{M_S}\}$ of
             $MG$.
 1  Define one zero vector $\boldsymbol{X}_{MG}^{M_S}$ with $m$ dimensions;
 2  for $Sg_j \in F$ do
 3      if $\exists G \in MG, G \supseteq Sg_j$ then
 4          Set the weight $(x^{Sg_j})^{M_S}$ of the $j$th dimension in $\boldsymbol{X}_{MG}^{M_S}$ as 1;
 5  return $\boldsymbol{X}_{MG}^{M_S}$;
```

Table 1
Containment relations between MG_i and $Sg_j(1 \leq i \leq 4, 1 \leq j \leq 5)$.

Multi-graph	Sg_1	Sg_2	Sg_3	Sg_4	Sg_5
MG_1	1	0	1	1	0
MG_2	0	1	1	0	0
MG_3	1	1	0	0	1
MG_4	1	1	1	1	1

4.3. Prediction model construction

In this section, we propose the multi-graph vectorization algorithm(MGV) to represent the multi-graphs into the vectors, based on the mined subgraph features and the subgraph feature representation of multi-graph. After that, the existing semi-supervised learning methods can be used to construct the prediction models, respectively.

4.3.1. Multi-graph vectorization

The basic idea of MGV is as follows. If we mine m subgraph features $F = \{Sg_1, \ldots, Sg_m\}$, every multi-graph $MG_i \in M_S$ is represented as an m dimension vector $\boldsymbol{X}_i^{M_S} \in X_T$ according to the following method. If one of the graphs in MG_i contains the Sg_j, the jth weight of $\boldsymbol{X}_i^{M_S}$ is assigned a value of 1. Otherwise, the jth weight of $\boldsymbol{X}_i^{M_S}$ is assigned a value of 0. The pseudo code of MGV is shown in Algorithm 4, where every multi-graph is transformed into a vector listed in Algorithm 5. The following is an example of a multi-graph vectorization.

Example 4. Given a multi-graph set $M_S = \{MG_1, MG_2, MG_3, MG_4\}$ and a mined subgraph feature set $F = \{Sg_1, Sg_2, Sg_3, Sg_4, Sg_5\}$ from M_S, the containment relations between $MG_i(1 \leq i \leq 4)$ and $Sg_j(1 \leq j \leq 5)$ are shown in Table 1, where “1” means that there is at least one of graphs in $MG_i \in M_S$ containing the subgraph feature $Sg_j \in F$; ”0” means that none of graphs in $MG_i \in M_S$ contains the subgraph feature $Sg_j \in F$. So, the transformational vector set X of M_S is as follow. $X = \{\boldsymbol{X}_{MG_1}^{M_S} = (1, 0, 1, 1, 0),\ \boldsymbol{X}_{MG_2}^{M_S} = (0, 1, 1, 0, 0),\ \boldsymbol{X}_{MG_3}^{M_S} = (1, 1, 0, 0, 1),\ \boldsymbol{X}_{MG_4}^{M_S} = (1, 1, 1, 1, 1)\}$.

4.3.2. Constructing prediction model

Existing semi-supervised Bayes [34], semi-supervised SVM [35] or ESELM [6] can all be utilized to construct a prediction model. However, extensive experimental results show that ESELM has a higher accuracy than semi-supervised Bayes and semi-

Algorithm 6. ESELM.

```
Input  : The training dataset $X_T = \{(\boldsymbol{X}_i, \boldsymbol{t}_i) | \boldsymbol{X}_i \in R_n, \boldsymbol{t}_i \in R_m, i = 1, ..., N\}$
         with a small number of labeled samples, the hidden node number $L$
         and the activation function g($\boldsymbol{X}_i$).
Output: an ELM instance.
produce the input weight $w_i$ and the bias $b_i$ randomly, where $i$=1,..., $L$;
Calculate the permutation matrix $S$ using the small number of labeled samples;
Calculate the hidden layer output matrix $H$;
Calculate the weight matrix $W$ and graph matrix $L$, and construct a
  semi-supervised graph;
Calculate the output weight $\beta$.
```

Algorithm 7. MGSSL.

```
Input  : The training multi-graph dataset $M_S$ with a small number of labeled
         positive and negative mutli-graphs, the testing multi-graph dataset $T$,
         support threshold $\beta$ and the optimal feature subgraph number $m$.
Output: The classification label of a multi-graph in $T$
1 //Tranining phase
2 The optimal feature subgraph set $F$=OSFS($M_S$, $\beta$, $m$);
3 Transform the multi-graphs in $M_S$ into vectors $\boldsymbol{X}^{M_S}$ based on $F$;
4 Construct prediction model $H$ based on the vector set $\{\boldsymbol{X}^{M_S}\}$;
5 //Testing phase
6 Transform the multi-graphs in T into vectors in $X^T$;
7 Predict the classification label of $\forall \boldsymbol{X}_t^{M_S} \in X^T$ using prediction model $H$;
```

supervised SVM. Therefore at next, we give more details of ESELM algorithm in Algorithm 6. The basic idea of ESELM is as follows. At first, we calculate the similarity of labeled samples. And then, we calculate the graph structure of datasets [38,39], where labeled and unlabeled samples are treated as vertices and each sample is connected to its k-nearest neighbors(k is a given parameter). At last, we get the labels of samples from a linear classifier.

4.4. Multi-graph semi-supervised learning

Based on the above proposed algorithms, we design a multi-graph semi-supervised learning algorithm, i.e., MGSSL, whose pseudo-code is as shown in Algorithm 7.

At first, the optimal subgraph feature set F is first mined using our OSFS algorithm(lines 1–2); then, the multi-graphs in the training dataset M_S are transformed into vectors based on Definition 6(line 3); in the next step, the traditional semi-supervised learning algorithm is used to construct the prediction model(line 4); at next, the multi-graphs in the testing dataset T are vectorized(lines 5–6); finally, every testing multi-graph is predicted with the trained prediction model in the testing phase(lines 7).

5. Performance evaluation

In this section, we analyze the performance of the upper bound pruning strategy of MGSSL, compare the performance of MGSSL with baseline methods, and study the different accuracies of our algorithm by varying the parameters over real and synthetic datasets. Datasets and experiment setting are introduced as follows:

DBLP dataset. Every paper P_i is regarded as a multi-graph MG_i. The abstract of P_i is modeled as a graph $G_j \in MG_i$ using E-FCM [40,41]. In addition, the abstract of each reference is represented as a graph. Domain field of a paper is treated as its class label. We select two domain fields to conduct tests in this paper, i.e., artificial intelligence AI(including IJCAI, AAAI, NIPS, UAI, COLT, ACL, KR, ICML, ECML and IJCNN) and computer vision CV(including ICCV, CVPR, ECCV, ICPR, ICIP, ACM multimedia and ICME). We remove those papers which have no references and papers which have references with both the above two domain fields. After preprocessing, there are 7661 AI multi-graphs and 1817 CV multi-graphs. We randomly select 500 AI multi-graphs and 500 CV multi-graphs to test. Here, $|MG^+| = |MG^-| = 500$ and $|G| = 1,289$.

Table 2
Accuracy of MGSSL and MGSSL-U using SS-Bayes, SS-SVM or ESELM on DBLP or SYN dataset with 30 labeled multi-graphs (m=20, β=0.14).

	DBLP			*SYN*		
	SS-Bayes	SS-SVM	ESELM	SS-Bayes	SS-SVM	ESELM
MGSSL	50.21%	57.84%	58.3%	50.09%	60.00%	61.11%
MGSSL-U	50.21%	57.84%	58.3%	50.09%	60.00%	61.11%

The DBLP dataset can be downloaded by accessing the URL of http://arnetminer.org/citation.

Synthetic(SYN) dataset. National Cancer Institute(NCI) cancer screening data sets [42] are widely utilized as a benchmark for graph classification. Each NCI data set belongs to a bio-assay task for anti-cancer activity prediction. Each chemical compound is a graph, where atoms denote nodes and bounds denote edges in the graph. If a chemical compound is active against the corresponding cancer, it is tagged as positive. Otherwise, it is tagged as negative. We generate a synthetic multi-graph dataset with a graph data set(with ID 1) adopting the method used in literatrue [1]. In total, we build 500 positive and 500 negative multi-graphs. The total number of graphs is 10,191.

We randomly select 30, 50 and 70 multi-graphs as labeled samples from DBLP and SYN dataset, respectively, where the remaining multi-graphs are used as unlabeled samples. In DBLP dataset, 30, 50 and 70 labeled multi-graphs averagely contain 35, 69 and 92 graphs, respectively. In SYN dataset, 30,50 and 70 labeled multi-graphs averagely contain 306, 509 and 663 graphs, respectively.

The baseline methods are MGSSL-M and gMGSL. In MGSSL-M, We first mine top-m frequent subgraphs from the labeled multi-graphs. Then, based on these mined frequent subgraphs and Definition 6, every multi-graphs are represented by binary vectors. Lastly, an existing semi-supervised classification method is utilized to build a classifier for predicting the class labels of unseen multi-graphs. gMGSL is a multi-graph classification algorithm proposed by Wu et al. [1], where the optimal subgraph features are mined from the labeled multi-graphs.

All experiments are run on a single machine with two 3.1 HZ CPUs, 8 GB Memory, 500GB hard disk and Redhat 4.4.4–13 operation system. In this paper, the default value of every parameter is employed in all algorithms. Every experiment is conducted three times, and the average of the three experimental results are reported, including accuracy and subgraph feature mining time.

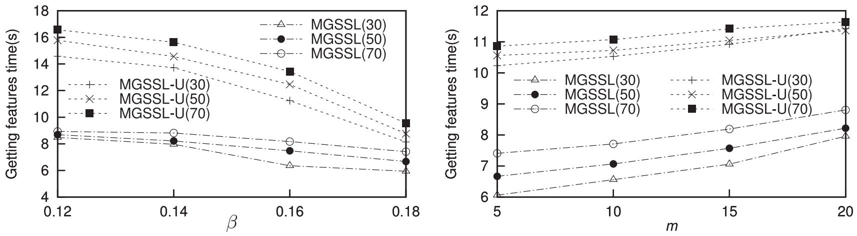
5.1. Upper bound pruning strategy

In this section, we compare the accuracy and the subgraph feature mining time of MGSSL, i.e., MGSSL with the upper bound pruning strategy, with MGSSL-U, i.e., MGSSL without the upper bound pruning strategy, on the DBLP and the SYN datasets, respectively. Experimental results are shown in Table 2 and Figs. 4 and 5.

Table 2 shows that both MGSSL and MGSSL-U have the same classification accuracy when they run on DBLP or SYN dataset with 30 labeled multi-graphs. Because we only prune some subgraphs, which must not be the optimal subgraph features, by our upper bound pruning strategy. In other words, the usage of the upper bound pruning condition does not change the final selected optimal subgraph features, but speeds up the optimal subgraph feature selection. Thus, both MGSSL and MGSSL-U have the uniform accuracy under the identical conditions.

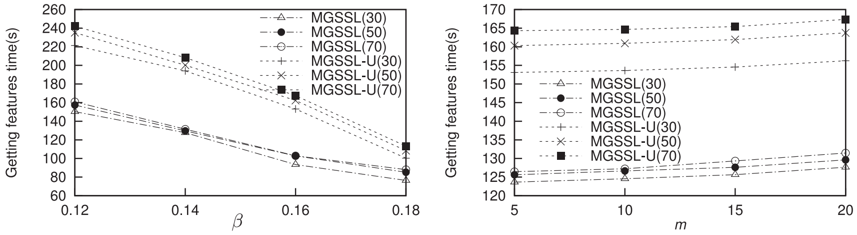
Fig. 4(a) shows that, compared to MGSSL-U, MGSSL costs less subgraph feature mining time under the same conditions on the DBLP dataset containing 30, 50 or 70 labeled multi-graphs. That is because the upper bound pruning strategy in MGSSL effectively

Please cite this article as: J. Pang et al., Semi-supervised multi-graph classification using optimal feature selection and extreme learning machine, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.01.114



(a) Different β's (m=20) (b) Different m's (β=0.14)

Fig. 4. Subgraph feature mining time of MGSSL and MGSSL-U over DBLP dataset with 30, 50 or 70 labeled multi-graphs.



(a) Different β's (m=20) (b) Different m's (β=0.14)

Fig. 5. Subgraph feature mining time of MGSSL and MGSSL-U over SYN dataset with 30, 50 or 70 labeled multi-graphs.

reduces search space and promotes the subgraph feature mining. MGSSL(i) and MGSSL-U(i) denote that MGSSL and MGSSL-U run over a dataset with i labeled mutli-grpahs, respectively($i = 30$, 50, or 70). Meanwhile, with the increase of the support threshold(β), the subgraph feature mining time of both MGSSL and MGSSL-U reduces. In addition, with the increase of β, the reducing speed of the running time in MGSSL-U is faster than that in MGSSL. Because the main time cost of MGSSL-U is used to mine frequent subgraph. Existing efficient frequent subgraph mining algorithms, such as gSpan and FSG, are sensitive to the support threshold, i.e., the running time increases rapidly with the decrease of the support threshold. MGSSL is less sensitive to the support threshold than MGSSL-U, since MGSSL not only uses the threshold filtering, but also uses the upper bound pruning strategy.

Fig. 4(b) shows that the subgraph feature mining time of both MGSSL and MGSSL-U increases with the growth of the number of the subgraph features(m) on the DBLP dataset containing 30, 50 or 70 labeled multi-graphs. Meanwhile, the subgraph feature mining time of MGSSL is shorter than that of MGSSL-U under the same conditions. Because MGSSL adopts the upper pruning strategy to effectively reduce the search space.

Fig. 5 compares the time of the subgraph feature mining of MGSSL and MGSSL-U on the SYN dataset. The experimental results show that MGSSL uses less time than MGSSL-U on the SYN dataset because of the adoption of the upper bound pruning strategy.

5.2. Accuracy comparison with the baseline methods

In this section, the accuracies of MGSSL and the baseline methods are compared by varying the number of the subgraph features on the DBLP and the SYN dataset. Fig. 6 reports the accuracies of MGSSL and the baseline methods using SS-SVM, SS-Bayes or ES-ELM on the DBLP dataset with 30 labeled multi-graphs.

Fig. 6(a) shows that for the same number of the subgraph features, the accuracy of the MGSSL is higher than other algorithms. Because MGSSL adopts our proposed scoring function of the subgraph feature, which not only considers the contribution of the unlabeled multi-graphs, but also takes the two levels constraint property into account. Additionally, the accuracies of MGSSL and the baseline methods increase with the increase of the number of the subgraph features. Because the number of the subgraph feature become large, the number of the valuable subgraphs may grow. Experimental results reported in Fig. 6(b) and(c) show that our MGSSL achieves a higher classification accuracy than baseline methods. That is because our MGSSL designs an novel optimal subgraph feature selection metric. Our metric selects a few features, which are more valuable than ones mined by the metrics of MGSSL-M and gMGSL.

Fig. 7 reports the accuracies of MGSSL and the baseline methods using SS-SVM, SS-Bayes or ESELM on SYN dataset with 30 labeled multi-graphs. Fig. 7 displays that MGSSL acquires significantly higher accuracy than MGSSL-M and gMGSL over SYN dataset.

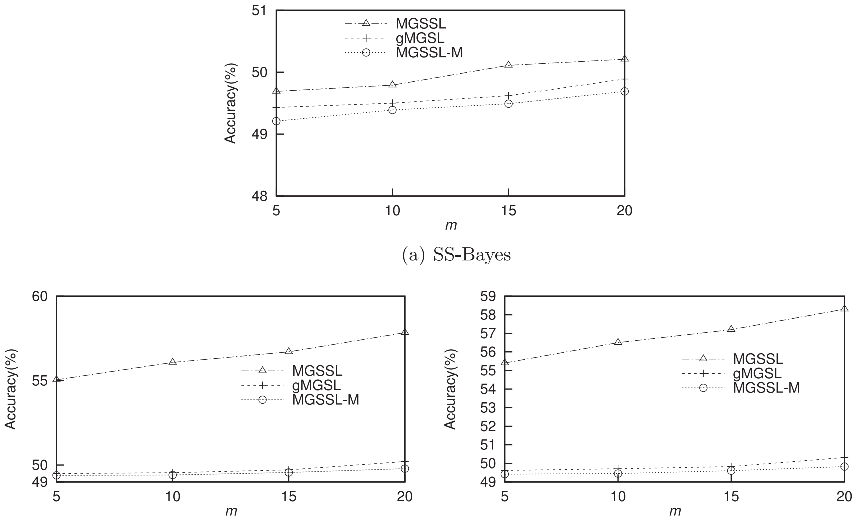
Fig. 8 shows that the accuracy of MGSSL with ESELM is higher than the accuracy of MGSSL with SS-SVM or SS-Bayes, which can verify the superiority of ESELM over SS-Bayes and SS-SVM in term of accuracy on DBLP and SYN dataset. MGSSL+SS-Bayes, MGSSL+SS-SVM and MGSSL+ESELM denote MGSSL using SS-Bayes, SS-SVM or ESELM algorithm, respectively.

5.3. Accuracy with different support thresholds

In this section, we compare the accuracies of MGSSL and the baseline methods with different support thresholds(β's) over the DBLP and SYN datasets containing 30 labeled multi-graphs. The experimental results are shown in Figs. 9 and 10.

Fig. 9(a) shows that the accuracies of all algorithms using SS-Bayes decrease with the increase of the support threshold over DBLP dataset. Because the informative subgraphs may not always be frequent but also may be infrequent. If the support threshold increases, we may omit more and more valuable subgraphs, whose supports are less than the support threshold. It is observed that the accuracy of our MGSSL obviously reduces with the growth of β over DBLP dataset in Fig. 9(b) and (c). Because MGSSL, which adopts the threshold pruning strategy, may ignore a few informative subgraph features with the growth of β. Fig. 10 shows that the accuracies of all algorithms decrease with the increase of the support threshold over SYN dataset. It is observed that the accuracy

Please cite this article as: J. Pang et al., Semi-supervised multi-graph classification using optimal feature selection and extreme learning machine, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.01.114

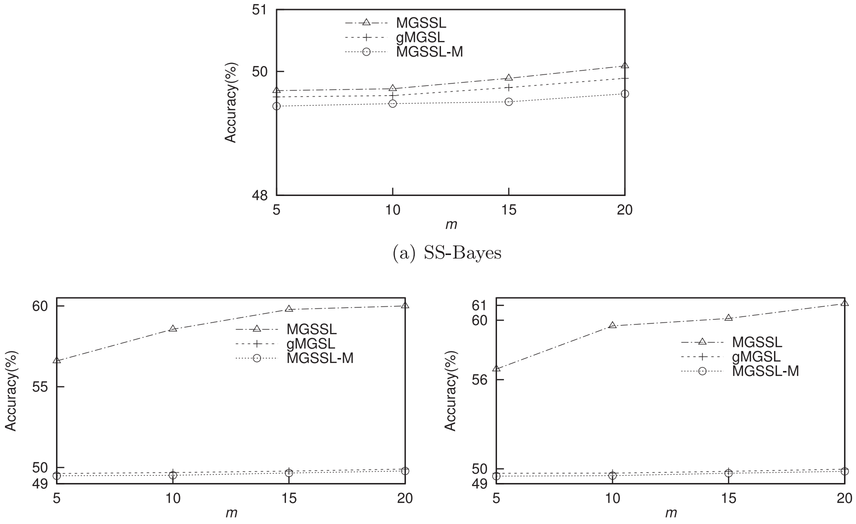


(a) SS-Bayes

(b) SS-SVM

(c) ESELM

Fig. 6. Accuracy of different algorithms using SS-Bayes, SS-SVM or ESELM by varying *m*'s on DBLP dataset with 30 labeled multi-graphs (β=0.14).

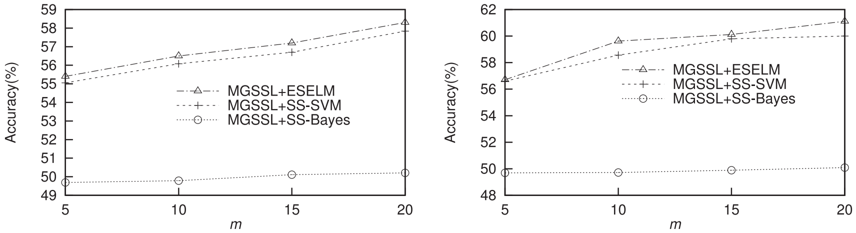


(a) SS-Bayes

(b) SS-SVM

(c) ESELM

Fig. 7. Accuracy of different algorithms using SS-Bayes, SS-SVM or ESELM by varying *m*'s on SYN dataset with 30 labeled multi-graphs (β=0.14).



(a) DBLP

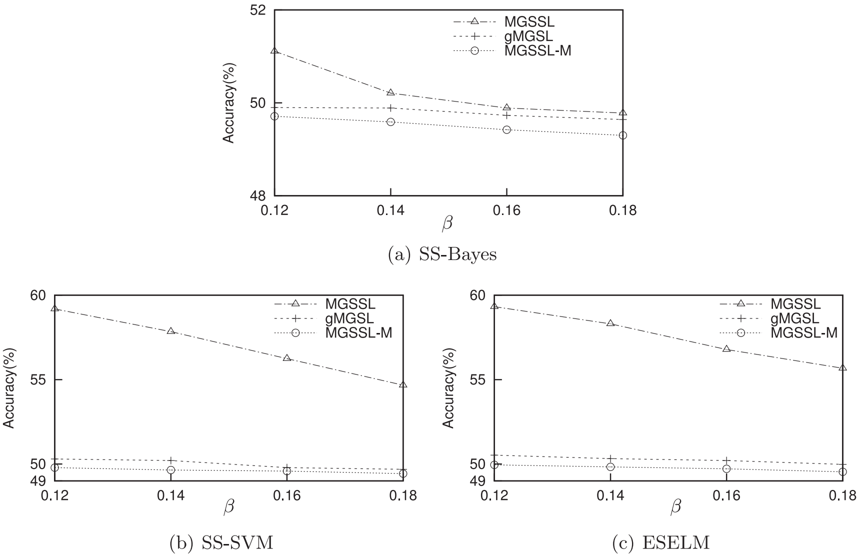
(b) SYN

Fig. 8. Accuracy of MGSSL using SS-Bayes, SS-SVM or ESELM by varying *m*'s on DBLP dataset and SYN dataset (β=0.14).

of MGSSL slowly decreases with the increase of β on SYN dataset in Fig. 10. That is because the subgraph frequency distributions of DBLP and SYN datasets are different.

6. Conclusions

In this paper, we propose an novel algorithm MGSSL to solve the semi-supervised multi-graph classification problem. For selecting the optimal subgraph features to improve the classification accuracy of our proposal, we propose an evaluation criterion, which not only takes the unlabeled multi-graphs into account, but also takes the constraint properties between the multi-graph level and the graph level into account. Then, the optimal subgraph feature selection problem is equivalently translated into the *m* most valuable subgraph feature mining problem, which is solved utilizing our proposed OSFS algorithm. Moreover, an upper bound pruning

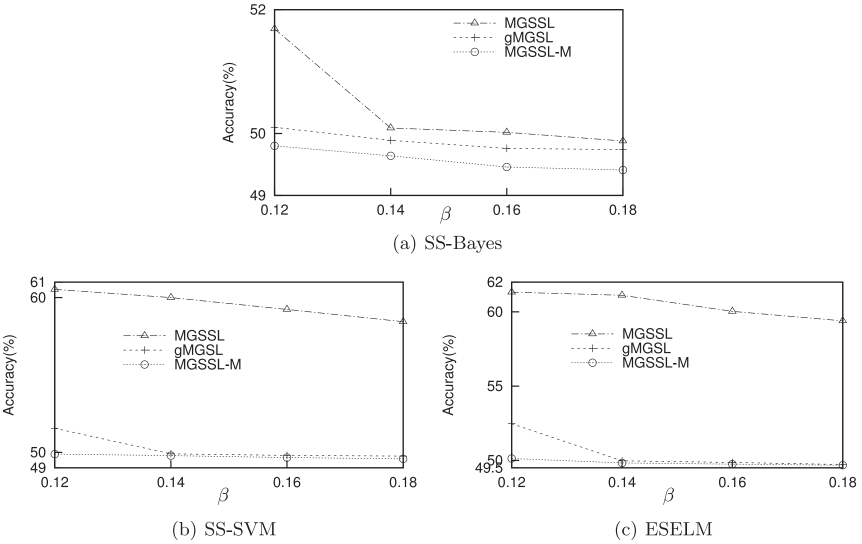


(a) SS-Bayes

(b) SS-SVM

(c) ESELM

Fig. 9. Accuracy of different algorithms using SS-Bayes, SS-SVM or ESELM by varying β's on DBLP dataset with 30 labeled multi-graphs (m=20).



(a) SS-Bayes

(b) SS-SVM

(c) ESELM

Fig. 10. Accuracy of different algorithms using SS-Bayes, SS-SVM or ESELM by varying β's on SYN dataset with 30 labeled multi-graphs (m=20).

strategy is proposed to reduce the search space of OSFS. Specially, semi-supervised ELM is adopted to further improve the classification accuracy. Extensive experimental results on both real and synthetic datasets show that our MGSSL costs less subgraph feature mining time and apparently outperforms the baseline methods in terms of the classification accuracy.

Acknowledgments

The work is partially supported by the National Natural Science Foundation of China (No. 61702381, No. 61472071, No. 61272179), the Scientific Research Foundation of Wuhan University of Science and Technology (2017xz015). Jia Xu is supported by the National Natural Science Foundation of China (No. 61402494), the Guangxi Natural Science Foundation (No. 2015GXNSFBA139243) and the Scientific Research Foundation of GuangXi University (No.XGZ141182). On besides, we really appreciate Shenglan Liu for sharing the source codes of ESELM.

References

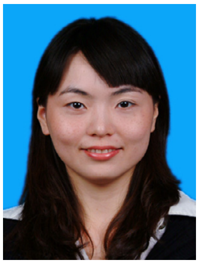
[1] J. Wu, X. Zhu, C. Zhang, Bag constrained structure pattern mining for multi–graph classification, TKDE 26 (2014) 2382–2396.
[2] J. Wu, S. Pan, X. Zhu, Boosting for multi-graph classification, Trans. Cybern. 45 (2015) 430–443.
[3] X. Kong, P. Yu, Semi-supervised feature selection for graph classification, in: Proceedings of the KDD, 2010, pp. 793–802.
[4] B. Liu, in: Web Data Mining–exploring Hyperlinks, Contents, And Usage Data, Springer, Verlag Berlin Heidelberg, 2011, pp. 171–201.
[5] J. Wu, Z. Hong, S. Pan, Multi-graph learning with positive and unlabeled bags, in: Proceedings of the SDM, 2014, pp. 217–225.
[6] S. Liu, L. Feng, H. Wang, Extend semi-supervised ELM and a frame work, Neural Comput. Appl. 27 (1) (2014) 1–9.
[7] X. Yan, H. Cheng, J. Han, Mining significant graph patterns by leap search, in: Proceedings of the SIGMOD, 2008, p. 433C-444.
[8] S. Ranu, A. Singh, Graphsig: a scalable approach to mining significant subgraphs in large graph databases, in: Proceedings of the ICDE, 2009, pp. 844–855.
[9] N. Jin, W. Wang, LTS: discriminative subgraph mining by learning from search history, in: Proceedings of the ICDE, 2011, pp. 207–218.
[10] Y. Zhu, J. Yu, H. Cheng, Graph classification: a diversified discriminative feature selection approach, in: Proceedings of the CIKM, 2012, pp. 205–214.

Please cite this article as: J. Pang et al., Semi-supervised multi-graph classification using optimal feature selection and extreme learning machine, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.01.114

[11] G. Huang, Q. Zhu, C. K.Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: Proceedings of the IJCNN, 2004, pp. 985–990.
[12] G. Huang, N. Liang, H. Rong, On-line sequential extreme learning machine, in: Proceedings of the IASTED, 2005, pp. 232–237.
[13] G. Huang, Q. Zhu, C. K.Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.
[14] G. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (16–18) (2007) 3056–3062.
[15] G. Huang, L. Chen, Enhanced random search based incremental extreme learning machine, Neurocomputing 71 (16–18) (2008) 3460–3468.
[16] G. Huang, X. Ding, H. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing 74 (1–3) (2010) 155–163.
[17] G. Huang, D. Wang, Y. Lan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cybern. 2 (2) (2011) 107–122.
[18] G. Huang, H. Zhou, X. Ding, Extreme learning machine for regression and multiclass classification, Trans. Syst. Man Cybern. Part B Cybern. 42 (2) (2012) 513–529.
[19] G. Huang, D. Wang, Advances in extreme learning machines(ELM2011), Neurocomputing 102 (2013) 1–2.
[20] G. Huang, An insight into extreme learning machines: random neurons, random features and kernels, Cogn. Comput. 6 (3) (2014) 376–390.
[21] G. Huang, X. Bai, L. L. C.Kasun, Local receptive fields based extreme learning machine, Comput. Intell. Mag. 10 (2) (2015) 18–29.
[22] J. Xin, Z. Wang, C. Chen, ELM*: Distributed extreme learning machine with mapreduce, World Wide Web 17 (5) (2014) 1189–1204.
[23] J. Xin, Z. Wang, L. Qu, Elastic extreme learning machine for big data classification, Neurocomputing 149 (Part A) (2015) 464–471.
[24] X. Bi, X. Zhao, G. Wang, Distributed extreme learning machine with kernels based on mapreduce, Neurocomputing 149 (2015) 456–463.
[25] B. Wang, S. Huang, J. Qiu, Parallel online sequential extreme learning machine based on mapreduce, Neurocomputing 149 (2015) 224–232.
[26] G. Wang, Y. Zhao, D. Wang, A protein secondary structure prediction framework based on the extreme learning machine, Neurocomputing 72 (1–3) (2008) 262–268.
[27] Y. Sun, Y. Yuan, G. Wang, An OS-ELM based distributed ensemble classification framework in p2p networks, Neurocomputing 74 (16) (2011) 2438–2443.
[28] X. Zhao, G. Wang, X. Bin, XML document classification based on ELM, Neurocomputing 74 (16) (2011) 2444–2451.
[29] Z. Wang, Y. Zhao, G. Wang, On extending extreme learning machine to non-redundant synergy pattern based graph classification, Neurocomputing 149 (2015) 330–339.
[30] J. Pang, Y. Gu, J. Xu, Parallel multi-graph classification using extreme learning machine and mapreduce, ELM (2015) 77–92.
[31] J. Liu, Y. Chen, M. Liu, SELM: Semi-supervised ELM with application in sparse calibrated location estimation, Neurocomputing 74 (16) (2011) 2566–2572.
[32] K. Li, J. Zhang, H. Xu, A semi-supervised extreme learning machine method based on co-training, J. Comput. Inf. Syst. 9 (1) (2013) 207–214.
[33] G. Huang, S. Song, JND, Semi-supervised and unsupervised extreme learning machines, IEEE Trans. Cybern. 44 (12) (2014) 2405–2417.
[34] R. Xia, OpenPR-NBEM: a C++ implementation of expectation-maximization (EM) algorithm with naive bayes for unsupervised and semi-supervised learning, http://www.mloss.org/software/view/357/.
[35] T. Joachims, Making large-scale svm learning practical, Advances in kernel methods - support vector learning, in: B. Schölkopf, C. Burges, A. Smola (Eds.), MIT-Press, 1999.
[36] M. Grbovic, C. Dance, S. Vucetic, Sparse principal component analysis with constraints, in: Proceedings of the AAAI, 2012, pp. 935–941.
[37] X. Yan, J. Han, gspan: Graph-based substructure pattern mining, in: Proceedings of the ICDM, 2002. 721C-724
[38] F. Wang, J. Wang, C. Zhang, Semisupervised classification using linear neighborhood propagation, in: Proceedings of the CVPR, 2006, pp. 160–167.
[39] S. Yan, H. Wang, Semi-supervised learning by sparse representation, in: Proceedings of the SIAM International Conference on Data Mining, 2009, pp. 792–801.
[40] K. Perusich, M. Senior, Using fuzzy connitive maps for knowledge management in a conflict environment, Trans. System Man Cybern. Part C 36 (6) (2006) 810–821.
[41] X. Luo, Z. Xu, J. Yu, Building association link network for semantic link on web resources, Trans. Autom. Sci. Eng. 8 (3) (2011) 482–494.
[42] Pubchem, http://pubchem.ncbi.nlm.nih.gov.

Jun Pang was born in 1984. He received his Ph.D. degree in Computer Science from Northeastern University in 2016. Currently, he is a Lecturer in College of Computer Science and Technology, Wuhan University of Science and Technology. His research interests include cloud computing, similarity join and graph data management.

Yu Gu was born in 1981. He received the Ph.D. degree in Computer Science from Northeastern University in 2010. Currently, he is a Professor and Ph.D. Supervisor in School of Computer Science and Engineering, Northeastern University. His research interests include spatial data management and graph data management.



Jia Xu was born in 1984. She received the Ph.D. degree in Computer Science from Northeastern University in 2013. She is currently an Associate Professor at Department of Computer, Electronics and Information from Guangxi University. Her research interests include RFID data management, data privacy protection and big data management.

Ge Yu was born in 1962. He received his Ph.D. degree in Computer Science from Kyushu University of Japan in 1996. Now he is a Professor and Ph.D. Supervisor at Northeastern University, the senior member of CCF, and the member of IEEE and ACM. His research interests include data base theory and technology.

Please cite this article as: J. Pang et al., Semi-supervised multi-graph classification using optimal feature selection and extreme learning machine, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.01.114