Noname manuscript No.
(will be inserted by the editor)

Multi-Graph-View Subgraph Mining for Graph Classification

Jia Wu · Zhibin Hong · Shirui Pan · Xingquan Zhu · Zhihua Cai · Chengqi Zhang

Received: Dec 13, 2014 / Revised: Apr 08, 2015 / Accepted: May 21, 2015

Abstract In this paper, we formulate a new Multi-Graph-View learning task, where each object to be classified contains graphs from multiple graph-views. This problem setting is essentially different from traditional single-graph-view graph classification, where graphs are collected from one single feature view. To solve the problem, we propose a Cross Graph-View Subgraph Feature based Learning (*gCGVFL*) algorithm that explores an optimal set of subgraphs, across multiple graph-views, as features to represent graphs. Specifically, we derive an evaluation criterion to estimate the discriminative power and redundancy of subgraph features across all views, with a branch-and-bound algorithm being proposed to prune subgraph search space. Because graph-views may complement to each other and play different roles in a learning task, we assign each view with a weight value indicating its importance to the learning task, and further use an optimization process to find optimal weight values for each graph-view. The iteration between cross graph-view subgraph scoring and graph-view weight updating forms a closed loop to find optimal subgraphs to represent graphs for multi-graph-view learning. Experiments and comparisons on real-world tasks demonstrate the algorithm's superior performance.

Keywords Multi-Graph-View · Feature Selection · Subgraph Mining · Graph Classification

J. Wu
Quantum Computation & Intelligent Systems (QCIS) Centre, FEIT,
University of Technology Sydney, NSW 2007, Australia;
School of Computer Science, China University of Geosciences, Wuhan 430074, China
E-mail: jia.wu@student.uts.edu.au

Z. Hong, S. Pan and C. Zhang
Quantum Computation & Intelligent Systems (QCIS) Centre, FEIT,
University of Technology Sydney, NSW 2007, Australia
E-mail: {zhibin.hong@student.,shirui.pan@student.,chengqi.zhang@}uts.edu.au

Z. Cai (✉)
School of Computer Science, China University of Geosciences, Wuhan 430074, China,
E-mail: zhcai@cug.edu.cn

X. Zhu
Department of Computer & Electrical Engineering and Computer Science,
Florida Atlantic University, Boca Raton, FL, USA.
E-mail: xzhu3@fau.edu

1 Introduction

Graphs have been prevalently used in many applications to represent objects with complex structures, such as XML document categorization [2], bug identification in a computer program flow [5], and online product recommendation based on review [33]. In addition to simple graph learning, numerous researches have extended graph classification to various settings, such as multi-label graph classification [18], multi-graph classification [34], graph stream classification [1,25], *etc*. For all these methods, a common challenge is that graphs do not have vectorized features, so generic feature-based learning algorithms cannot be directly applied to graph classification.

One popular graph classification framework is to first mine some patterns (*i.e.,* frequent subgraphs [6,38], co-occurrent subgraphs [14], and subgraph distribution [8]) as features, and then transfer each graph into a single instance. Accordingly, the generic classification approaches can be directly applied for learning. This type of processing is similar to feature filtering methods which separate feature section and further classification into two sequential steps. Among them, frequent subgraphs have demonstrated good performance for graph classification [6]. In reality, the number of frequent subgraphs may increase exponentially with the size of training graphs, and some subgraph patterns may not have discriminative power for learning. Therefore, finding good subgraph features to represent graph data for learning is an essential challenge.

To address the above challenge, many approaches have been proposed for feature selection from graph data [41,17,8]. To date, all existing subgraph feature selection methods assume that training graphs are collected from a single feature view as shown in Figure 1. For example, in image retrieval, a graph can be used to denote an image with each node corresponding to a small region of the image and every two adjacent regions forming an edge [10]. By using colour histogram as a feature view to describe the node content, an image can be represented as a single-graph-view graph (*i.e.,* a colour histogram graph). Indeed, single-graph-view graphs are often limited to the underlying features and inadequate to fully describe the object content. For example, both colour and textures are two feature views commonly to represent images. By using colour histogram and textures to build two separate graphs for each image, we can obtain a multi-graph-view representation where

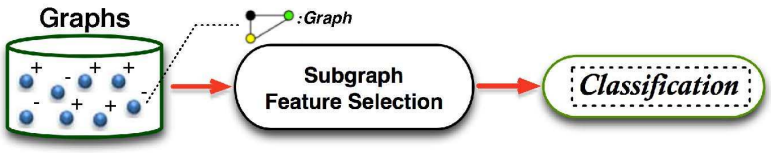


Fig. 1 Traditional subgraph feature based learning where graphs are collected from a single view.

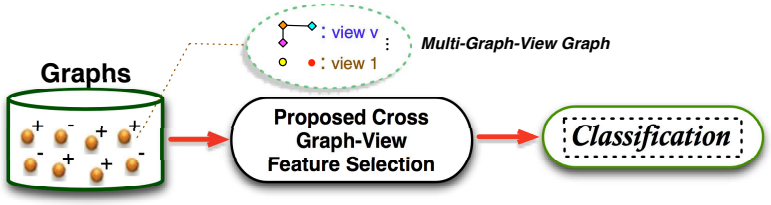


Fig. 2 Multi-Graph-View Learning where graphs are represented from different graph-views.

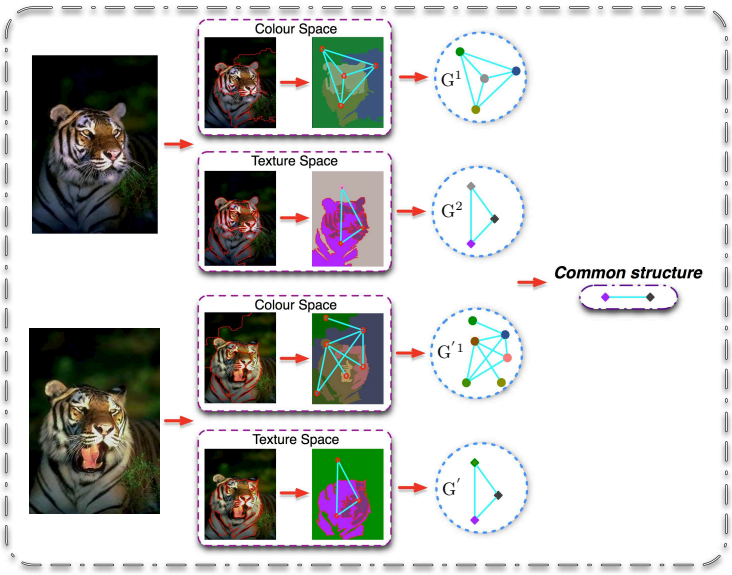


Fig. 3 An example of multi-graph-view representation for images where each image is represented as graphs with multiple graph-views (*e.g.,* colour view #1 and texture view #2). For the same objects, patterns shared by the objects may not exist in all feature views. For example, the two images have no common pattern in the colour space (*i.e.,* the graph G^1 and G'^1 in colour view), due to different lighting conditions, but they share the same patterns in the texture space (*i.e.,* the graph G^2 and G'^2 in texture view). Therefore, multi-graph-view representation is more comprehensive in representing the object content than single graph-view representation (the graph composition is detailed in Section 6.4).

each object is represented using multiple graphs, each of which is built from a single feature view. An example of the multi-graph-view representation is shown in Figure 3. Such a multi-graph-view representation essentially combines multiple feature views to describe the object, and is therefore potentially more comprehensive than traditional single-graph-view representation.

The above observations raise a novel problem setting: Multi-Graph-View Feature based Learning for Graph Classification. When learning from objects containing multi-view features, the importance of multiple feature views have already been well studied in the literature, where a handful of works have discussed the feature extraction problem from multi-view data [40,30]. Also, some works [7] proposed to use unlabeled data for unsupervised multi-view feature selection. All these exiting multi-view feature selection methods are designed for generic learning tasks where all samples have tabular instance-feature representations in vector space, so they cannot be directly applied to handle complex data objects. On the other hand, existing feature section algorithms on graph data cannot handle graphs with multi-graph-views.

To classify multi-graph-view graphs, a straightforward solution is to treat each view separately. Then, the Multi-Graph-View learning can be decomposed into multiple traditional single-graph-view graph classification tasks. However, such a trivial solution, *i.e.,* treating each graph-view separately and completely discarding other views, is clearly not an ideal solution because it will result in severe information loss. Another possible solution is to use a concatenation strategy to combine the most informative features selected from each single view as a new vector for learning. This intuitive method, however, cannot globally identify

the most informative subgraph features across all graph-views, because it only locally selects and combines features from each single graph-view. Not to mention that these methods normally require users to specify the number of features to be selected from each single view.

One slightly more intelligent design for multi-graph-view learning is to treat all graph-views equally and concatenate graphs from all graph-views as a complete graph set, and then apply traditional subgraph selection method to directly obtain subgraph features for graph classification. Nevertheless, this solution is still suboptimal mainly because (1) graph-views are not equally important for the learning tasks; and (2) the selected subgraph features may contain redundancy because there is no treatment to handle correlations and dependency between graph-views. Therefore, no effective method exists to automatically identify informative-irredundant subgraph features across multiple graph-views. To the best of our knowledge, our work is the first to explore the subgraph feature selection problem across multi-graph-view for graph classification.

Motivated by the above observations, we propose a cross graph-view subgraph feature based learning framework as shown in Figure 2, where the aim is to identity the most informative-irredundant subgraphs across all views for graph classification. To achieve the goal, the key challenge is threefold: 1) ***Subgraph Feature Evaluation.*** Conventional feature selection approaches cannot handle graph structured data, because they need the training data being represented as feature vectors. Meanwhile, extracting all subgraph features and then applying feature selection to extracted features is usually infeasible, mainly because that the number of subgraph features will grow exponentially with the graph size (*e.g.* number of nodes and edges). Furthermore, traditional feature selection on graph data is only suitable for single view, it is desired to design a new evaluation criterion for graphs with multi-graph-views; 2) ***Cross Graph-View Evaluation and Rating.*** The contribution of different feature views may vary significantly, where each single view has its unique statistical property. In addition, different feature views may also contain complementary or contradict information. Therefore, a new evaluation and rating method is required to assess all multi-graph-views; and 3) ***Cross Graph-View Redundancy.*** Considering the subgraph redundancy across different views, the corresponding redundancy constraint should be added to subgraph evaluation criteria to ensure that the selected subgraph features have minimum redundancy.

To address the above three challenges, the proposed optimization framework gCGVFL first designs a special measure to evaluate the importance of subgraphs in each graph-view (Challenge #1). By assigning proper weight values to different graph-views, gCGVFL can effectively adjust each graph-view's contribution to the cross graph-view feature selection process (Challenge #2). Furthermore, the proposed subgraph evaluation also considers the redundancy across all graph-views (Challenge #3). An optimization framework is then proposed to iteratively update the view weights and the informative-irredundant subgraphs across all graph-views. Experiments on two real-world learning tasks confirm the effectiveness of the proposed design.

2 Related Works

The proposed multi-graph-view learning for graph classification is essentially related to graph classification and multi-view learning.

2.1 Graph Classification

For graph classification, the main challenge, compared to the traditional supervised classification framework, is that graph objects do not have features. To this end, researchers have proposed many types of solutions, which can be broadly divided into the following two aspects: 1) graph based approaches, and 2) subgraph feature based methods. Because no feature representation is available for graph, one straightforward solution is to train some models directly on the graphs by using graph embedding or graph kernels. This type of approaches can perform well for graph classification problem, but they cannot indicate which part of graph makes the main contribution to the graph learning process. By contrast, subgraph feature based methods will explore some subgraphs from the given graph set as features to represent graphs in feature vectors for learning. In reality, because the subgraph searching space will increase exponentially with respect to the size of graphs, it is technically inefficient (or infeasible) to enumerate all subgraphs. To address the problem, some additional subgraph evaluation criteria are proposed. Among them, frequency is the most popular criterion. Yan and Han [39] proposed a Depth-First-Search (DFS) based subgraph mining strategy, namely gSpan, which first assigns a unique minimum DFS code to each graph, and then discovers all frequent subgraphs by a pre-order traversal of the tree. Some other frequent subgraph feature exploration approaches (*e.g.,* AGM [13], FSG [19], FFSM [12], MoFa [4] and Gaston [23]) have also be proposed. Nevertheless, these frequency strategies are all unsupervised, without utilizing the label information. To ensure the selected subgraph features to have high discriminative quality, supervised subgraph feature extraction methods have also been developed, such as LEAP [38], gPLS [27], COPK [29], etc. Moreover, Jin et al. [15] proposes an evolutionary strategy to mining discriminative subgraphs for graph classification in large databases. Recently, Kong *et al.* employs a dependence evaluation criteria HSIC (Hilbert-Schmidt Independence Criterion) [9] to solve the graph related tasks (*e.g.,* active subgraph feature mining problem [16] and multi-label subgraph feature selection issue [18]).

2.2 Multi-view Learning

As discussed above, all existing methods intend to extract subgraph features from graphs collected from a single-graph-view, which may not adequately describe the characteristics of the target object. Because objects may have different representative characteristics with respect to different feature spaces, exploring information from different graph-views can potentially help enhance the learning performance. For traditional multi-view feature based learning, the two essential problems include: 1) view evaluation, and 2) view combination [36]. Because views may have different contributions to the learning, how to evaluate these views and ensure their effectiveness has drawn many attentions. For example, Muslea *et al.* [26] proposed a decision tree based view validation approach to evaluation the view sufficiency. Considering that some views may inadequate for learning, Yan and Naphade [37] proposed a semi-supervised cross-feature learning by fully utilizing the information in the unlabeled dataset. Recently, Xia *et al.* [35] develop a novel multi-view spectral embedding, where each view will receive a weight value to indicate its importance.

For the sequential view combination issue, one common strategy is to concatenate all views to form a global view. By doing so, the multi-view problem can be translated to a single view learning. However, this type of view combination strategy may incur overfitting problems especially when the training samples are insufficient. Another popular view

combination strategy is based on Co-training [3], where a classifier is separately trained in each single view. After that, all classifiers are integrated for final prediction.

3 Problem Formulation

In this section, we define important notations used in the paper and formulate our problem definition.

Definition 1 *(Connected Graph)* A graph is represented as $G = (\mathcal{V}, E, \mathcal{L}, l)$ where $\mathcal{V} = \{v_1, \cdots, v_{n_v}\}$ is a set of vertices, $E \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges, and $\mathcal{L}$ is the set of symbols for the vertices and edges. $l : \mathcal{V} \cup E \rightarrow \mathcal{L}$ is the function assigning labels to the vertices and edges. A connected graph must has a path between any pair of vertices.

Definition 2 *(Graph-View)* A graph-view denotes a type of tuple $(\mathcal{V}, E, \mathcal{L}, l)$ used to represent an object as a graph. Similarly, multi-graph-view represents multiple types of tuples used to represent the same object.

Definition 3 *(Multi-Graph-View Graph)* $G_i = \{G_i^1, \cdots, G_i^k, \cdots, G_i^v\}$, a multi-graph-view graph, contains a number of graphs from different views, where G_i^k denotes a single-view graph (a connected graph) constructed from the k^{th} view. A graph G_i's label is denoted by $y_i \in \mathcal{Y}$, with $\mathcal{Y} = \{c_1, \cdots, c_L\}$. In this paper, we use G_i to denote a multi-graph-view graph, and use superscript or subscript k to denote a graph or a subgraph from the k^{th} view.

Definition 4 *(Subgraph)* Let $G = (\mathcal{V}, E, \mathcal{L}, l)$ and $g_i = (\mathcal{V}', E', \mathcal{L}', l')$ each denotes a connected graph. g_i is a subgraph of G, *i.e.*, $g_i \subseteq G$, iff there exists an injective function $\varphi : \mathcal{V}' \rightarrow \mathcal{V}$ s.t. (1)$\forall v \in \mathcal{V}', l'(v) = l(\varphi(v))$; (2) $\forall (u, v) \in E', (\varphi(u), \varphi(v)) \in E$ and $l'(u, v) = l(\varphi(u), \varphi(v))$. If g_i is a subgraph of G, then G is a supergraph of g_i. In this paper, subgraphs and subgraph features are equivalent terms.

Definition 5 *(Graph Feature Representation)* Let $\mathcal{S}^k = \{g_1, \cdots, g_{s_k}\}$ denote a set of subgraph features discovered from multi-graph-view graphs. For each graph G_i^k in the k^{th} view, we use a subgraph feature vector $\mathbf{x}_i^k = [(x_i^{g_1})^k, \cdots, (x_i^{g_{s_k}})^k]^\top$ to represent G_i^k in the feature space, where $(x_i^{g_e})^k = 1, 1 \leq e \leq s_k$, iff g_e is a subgraph of G_i^k (*i.e.,* $g_e \subseteq G_i^k$) and $(x_i^{g_e})^k = 0$ otherwise.

Given a multi-graph-view graph set $\mathcal{G} = \{\mathcal{G}^1, \cdots, \mathcal{G}^k, \cdots \mathcal{G}^v\}$ containing labeled graphs from v views, the **aim** of multi-graph-view learning is to find the optimal subgraph features from the training graph set $\mathcal{G}$ to train classification models, and predict previously unseen multi-graph-view graphs with a maximum accuracy.

4 Overall Framework of gCGVFL

Figure 4 lists the overall framework of the proposed cross graph-view subgraph feature based learning algorithm, which includes the following three major steps:

– **Subgraph Evaluation Criteria:** In order to explore the informative-irredundant subgraph feature across all graph-views, the subgraph discrimination criteria and subgraph redundancy criteria are proposed. These types of subgraph feature evaluation criteria will be unified to further obtain the cross graph-view subgraphs.

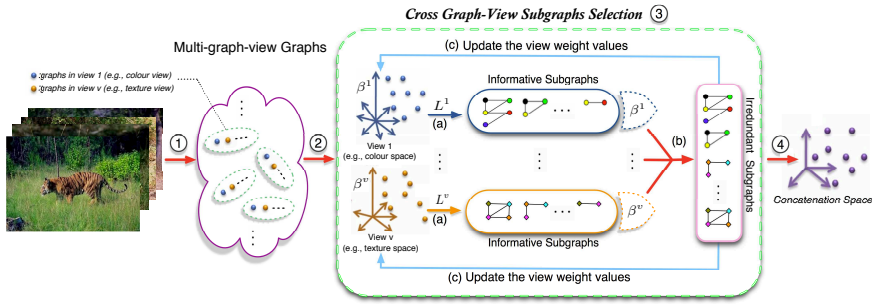


Fig. 4 A conceptual view of Cross Graph-View Subgraph Feature based Leaning (gCGVFL) for graph classification: After representing the original dataset (*e.g.,* images) into multi-graph-view graphs ①, gCGVFL intends to find optimal subgraph features to convert a multi-graph-view graph ② as an instance in the new feature space ④ by using proposed cross graph-view subgraph feature selection ③. More specifically, given a number of labeled multi-graph-view graphs with v views, gCGVFL starts from assigning equal weight values to each view. For each single graph-view, gCGVFL first builds a label distribution information embedded matrix L, *e.g.,* L^1 or L^v (a), to help discover informative subgraphs. By combining the view evaluation and redundancy checking, gCGVFL can obtain irredundant cross graph-view subgraphs (b). Because the initial view weights are imprecise to capture the importance of each view, an iterative weight updating process (detailed in Section 5.2) is proposed to update view weight value and refine the quality of selected cross graph-view subgraphs (c).

– **Cross Graph-View Subgraph Selection:** In order to properly identify the cross graph-view subgraph features, we assign an initial weight value to each view (*i.e.,* view evaluation). By doing so, we can build a bridge to evaluate the subgraph features cross all different views with the above subgraph evaluation criteria. However, the initial view evaluation is inaccurate. In this case, a weight updating strategy is proposed to ensure that high quality cross graph-view subgraphs are selected for further learning (detailed in Section 5.2).
– **Multi-Graph-View Graph Representation:** Based on the selected cross graph-view subgraph features, for each multi-graph-view graph $G_i = \{G_i^1, \cdots, G_i^k, \cdots, G_i^v\}$, we can obtain its feature representation in vector space for each single graph-view (*i.e.,* $\mathbf{x}_i^k$ for G_i^k in the k^{th} view). To represent G_i, a feature vector concatenation [33] is used to obtain the final vector representation across all graph-views $\mathbf{x}_i = [\mathbf{x}_i^1; \cdots \mathbf{x}_i^k \cdots; \mathbf{x}_i^v]$.

5 Multi-Graph-View Learning

5.1 Cross Graph-View Subgraph Exploration

Cross graph-view subgraph feature exploration for multi-graph-view learning intends to assess subgraph candidates and find a set of most informative subgraphs with min-redundancy crossing all graph-views to represent multi-graph-view graphs. In this section, we introduce the details of the optimization for cross graph-view subgraph feature selection.

Assume a set of graphs $\mathcal{G} = \{\mathcal{G}^1, \cdots, \mathcal{G}^k, \cdots \mathcal{G}^v\}$ are collected from v views, let $\mathcal{S} = \{\mathcal{S}^1, \cdots, \mathcal{S}^k, \cdots, \mathcal{S}^v\}$ denote the complete set of subgraphs discovered from $\mathcal{G}$, and $\mathbf{g} = \{\mathbf{g}^1, \cdots, \mathbf{g}^k, \cdots, \mathbf{g}^v\}$ is a set of cross graph-view subgraphs selected from $\mathcal{S}$ with $\mathbf{g}^k$ representing the subgraph set selected from the kth view. Our cross graph-view subgraph feature exploration **aims** to find a set of most informative subgraph features $\mathbf{g}$ ($\mathbf{g} \subseteq \mathcal{S}$), meanwhile with the minimum redundancy across all views. To this end, we define $\mathcal{I}(\mathbf{g})$

as an evaluation function to measure the informativeness of $\mathbf{g}$, with $\mathcal{R}(\mathbf{g})$ denoting the redundancy information in $\mathbf{g}$. Specifically, the objective of the subgraph feature exploration is defined in Eq. (1), where $|\cdot|$ represents the cardinality of the subgraph set, and m is the number of subgraphs to be selected from $\mathcal{S}$ across v views.

$$\begin{aligned} \mathbf{g}_\star = &\arg\max_{\mathbf{g}} \Big(\mathcal{I}(\mathbf{g})\Big) \\ &s.t.\ \mathcal{R}(\mathbf{g}) \leq \mathcal{R}(\mathbf{g}'), \mathbf{g} \subseteq \mathcal{S}, \mathbf{g}' \subseteq \mathcal{S}; \\ &|\mathbf{g}| = |\mathbf{g}'| = m; \end{aligned} \tag{1}$$

The optimal subgraph feature $\mathbf{g}_\star$ exploration problem across all views can be decomposed into two subproblems: cross graph-view max-discrimination (*i.e.*, $\max(\mathcal{I}(\mathbf{g}))$ and cross graph-view min-redundancy (*i.e.*, $\min(\mathcal{R}(\mathbf{g}))$ among all alternative graph-view subgraph feature sets).

5.1.1 Subgraph Discrimination Criterion

To address the first subproblem, subgraph exploration with maximum discrimination should consider the following two main challenges: (1) How to utilize graph labels to find informative subgraphs, and (2) How to tackle subgraph evaluation across different views, where every view has its specific characteristic, to find the optimal subgraphs.

To this end, we use a similarity based optimization to model each single view and carry out a cross graph-view optimization by assigning proper weight to each view. All selected subgraph features from different views are then unified to refine weight value of each view, through which the most discriminative subgraph features can be discovered.

For each single graph-view, to calculate the informativeness score of a subgraph feature set $\mathbf{g}^k$, *i.e.*, $\mathcal{I}(\mathbf{g}^k)$, we impose the class label information embedded constraints to graphs as follows: (a) *label information embedded must-link:* If two graphs G_i^k and G_i^k have the same class labels, we form a pairwise must-link constraint between G_i^k and G_j^k; and (b) *label information embedded cannot-link:* if G_i^k and G_j^k have different class labels, we form a cannot-link constraint between them. Because given the k^{th} view, each graph G_i^k is associated with a known class label, the subgraph features should ensure that graphs with the same label are highly similar to each other. For graphs with different class labels, subgraph features should also represent the disparity between them.

In order to derive solutions to find subgraphs as features, we first formally introduce notations as follows:

- $\mathcal{X}^k$: the matrix using subgraphs $\mathcal{S}^k$ to represent all graphs $\mathcal{G}^k$ in the kth view, $\mathcal{X}^k = [\mathbf{x}_1^k, \cdots, \mathbf{x}_{p_k}^k] = [\boldsymbol{f}_1^k, \cdots, \boldsymbol{f}_{s_k}^k]^\top \in \{0,1\}^{s_k \times p_k}$, with p_k denoting the size of graphs in the kth view. $\boldsymbol{f}_e^k$ ($1 \leq e \leq s_k$, $g_e^k \in S^k$), is an indicator vector of subgraph g_e^k with respect to all graphs in $\mathcal{G}^k$ in the kth view, *i.e.*, $\boldsymbol{f}_e^k = [f_e^{G_1^k}, \cdots, f_e^{G_{p_k}^k}]$, where $f_e^{G_i^k} = 1, 1 \leq i \leq p_k$ iff $g_e^k \subseteq G_i^k$ and $f_e^{G_i^k} = 0$ otherwise.
- A and B: $A = \{(i,j)|y_i \times y_j = 1\}$ denotes the *class label embedded must-link* pairwise constraint sets, while $B = \{(i,j)|y_i \times y_j = -1\}$ denotes the *class label embedded cannot-link* pairwise constraint sets.

Based on the above constraints, we derive a criterion to measure the cross graph-view informativeness score $\mathcal{I}(\mathbf{g})$ as follows:

$$\mathcal{I}(\mathbf{g}) = \sum_{k=1}^{v} \beta^k \mathcal{I}(\mathbf{g}^k) = \sum_{k=1}^{v} \beta^k \left(\frac{1}{2} \sum_{i,j} K_{\mathbf{g}}(G_i^k, G_j^k) Q_{i,j}^k \right) \tag{2}$$

where, $Q_{i,j}^k$ embeds class label information between two graphs G_i^k and G_j^k in the k^{th} view. β^k denotes the weight of the k^{th} view. $K_{\mathbf{g}}(G_i^k, G_j^k)$, which is defined in Eq. (3), denotes an L_2 norm distance between two graphs G_i^k and G_j^k in subgraph feature space based on the selected subgraph set $\mathbf{g}^k$,

$$\begin{aligned} K_{\mathbf{g}}(G_i^k, G_j^k) &=< \phi(G_i^k), \phi(G_j^k) >=< \mathcal{D}_{\mathbf{g}^k} \mathbf{x}_i^k, \mathcal{D}_{\mathbf{g}^k} \mathbf{x}_j^k > \\ &= \left\| \mathcal{D}_{\mathbf{g}^k} \mathbf{x}_i^k - \mathcal{D}_{\mathbf{g}^k} \mathbf{x}_j^k \right\|^2 \end{aligned} \tag{3}$$

where $\mathcal{D}_{\mathbf{g}^k} = diag(d(\mathbf{g}^k))$ is a diagonal matrix indicating which subgraph features $\mathbf{g}^k$ are selected from $\mathcal{S}^k$ to represent the graphs in the k^{th} view, $d(\mathbf{g}^k)_e = I(g_e^k \in \mathbf{g}^k)$ with $I(\cdot)$ being an indicator function.

Furthermore, in order to calculate $Q_{i,j}^k$ in Eq. (2), we adopt a kernel function, with respect to graphs' class labels, to measure $Q_{i,j}^k$ as $K(y_i, y_j) =< \varphi(y_i), \varphi(y_j) >$. In our experiments, $Q_{ij}^k = \{-1/|A|, y_i \times y_j = 1;\ 1/|B|, y_i \times y_j = -1\}$ is employed as the label kernel, through which the class label weight information is embedded in the matrix Q^k. For Eq. (2), if two graphs G_i^k and G_j^k have the same class labels (*i.e.,* $y_i \times y_j = 1$), then $Q_{i,j}$ will be -1 to ensure that maximizing $\mathcal{I}(\mathbf{g})$ is equivalent to minimizing the corresponding distance $K_{\mathbf{g}}(G_i^k, G_j^k)$.

Accordingly, Eq. (2) can be rewritten as follows,

$$\begin{aligned} \mathcal{I}(\mathbf{g}^k) &= \frac{1}{2} \sum\nolimits_{i,j} \left\| \mathcal{D}_{\mathbf{g}^k} \mathbf{x}_i^k - \mathcal{D}_{\mathbf{g}^k} \mathbf{x}_j^k \right\|^2 Q_{i,j}^k \\ &= tr\Big(\mathcal{D}_{\mathbf{g}^k}^\top \mathcal{X}^k \big(D^k - Q^k \big) \big(\mathcal{X}^k \big)^\top \mathcal{D}_{\mathbf{g}^k} \Big) \\ &= tr\Big(\mathcal{D}_{\mathbf{g}^k}^\top \mathcal{X}^k L^k \big(\mathcal{X}^k \big)^\top \mathcal{D}_{\mathbf{g}^k} \Big) \\ &= \sum\nolimits_{g_e^k \in \mathbf{g}^k} (\boldsymbol{f}_e^k)^\top L^k \boldsymbol{f}_e^k \end{aligned} \tag{4}$$

where $tr(\cdot)$ is the trace operator for a matrix, D^k is a diagonal matrix generated from Q^k, *i.e.,* $D_{i,i}^k = \sum_j Q_{ij}^k$. And $L^k = [L_{i,j}^k]^{p_k \times p_k} = D^k - Q^k$ is a Laplacian matrix. By denoting the function as $\hbar(g_e^k, L^k) = (\boldsymbol{f}_e^k)^\top L^k \boldsymbol{f}_e^k$, the Eq. (2) can be rewritten as

$$\mathcal{I}(\mathbf{g}) = \sum_{k=1}^{v} \mathcal{I}(\mathbf{g}^k) = \sum_{k=1}^{v} \sum_{g_e^k \in \mathbf{g}^k} \beta^k \hbar(g_e^k, L^k) \tag{5}$$

5.1.2 Subgraph Redundancy Criteria

According to the above cross graph-view subgraph discriminative criterion, a potential issue is that the selected informative subgraphs may have high discriminative scores, but may also contain high redundancy among them. For example, if a subgraph g has a high discriminative score, its supergraph and subgraph may also have a similar score. Then all the subgraphs

may be selected as the final subgraph features, which result in significant redundancy (and dependency) among the selected subgraph features. Given that the final feature set only contains m subgraph features, having highly redundant subgraph features may not only reduce the representation capability of the subgraphs, but also reduce the classification accuracy because dependent features always complicate learning algorithms' decisions.

In this subsection, we intend to evaluate the redundancy between subgraph features in order to find a subgraph feature set with minimum redundancy.

Definition 6 *(Cross Graph-View Redundancy)* Given two multi-graph-view subgraphs g^k and $g^{k'}$ ($1 \leq k \leq v, 1 \leq k' \leq v$) from two views, a multi-graph-view graph set $\mathcal{G} = \{\mathcal{G}^1, \cdots, \mathcal{G}^k, \cdots, \mathcal{G}^{k'}, \cdots, \mathcal{G}^v\}$ with its size N, and its corresponding view weight $\boldsymbol{\beta} = \{\beta^1, \cdots, \beta^k, \cdots, \beta^{k'}, \cdots, \beta^v\}, 0 \leq \beta^k \leq 1, \sum_{k=1}^{v} \beta^k = 1$, the subgraph redundancy $\psi(g^k, g^{k'})$ between g^k and $g^{k'}$ is defined by using the overlap over graph set $\mathcal{G}$ as:

$$\begin{aligned} \psi(g^k, g^{k'}) &= \frac{|\mathcal{G}(g^k) \cap \mathcal{G}(g^{k'})|_\beta}{|\mathcal{G}(g^k) \cup \mathcal{G}(g^{k'})|_\beta} = \frac{|\mathcal{G}_\cap|_\beta}{|\mathcal{G}_\cup|_\beta} \\ |\mathcal{G}_\cap|_\beta &= (\sum_{g^k \subseteq \mathcal{G}^k \in \mathcal{G}_\cap} \beta^k + \sum_{g^{k'} \subseteq \mathcal{G}^{k'} \in \mathcal{G}_\cap} \beta^{k'})/2; \\ |\mathcal{G}_\cup|_\beta &= |\mathcal{G}_\cap|_\beta + \sum_{g^k \subseteq \mathcal{G}^k \in \mathcal{G}^k_-} \beta^k + \sum_{g^{k'} \subseteq \mathcal{G}^{k'} \in \mathcal{G}^{k'}_-} \beta^{k'}; \end{aligned} \tag{6}$$

where $\mathcal{G}(g^k) = \{G_i^k | g^k \subseteq G_i^k \in \mathcal{G}^k\}$, similar with $\mathcal{G}(g'^k)$. $\mathcal{G}^k_- = \mathcal{G}(g^k) - \mathcal{G}_\cap$, and $\mathcal{G}^{k'}_- = \mathcal{G}(g^{k'}) - \mathcal{G}_\cap$. $\psi \in [0, 1]$. The higher the ψ value, the more redundancy g^k and $g^{k'}$ have. If $g^k = g^{k'}$, then $\mathcal{G}(g^k) = \mathcal{G}(g^{k'}) = \mathcal{G}_\cap$, and $\mathcal{G}^k_- = \mathcal{G}^{k'}_- = \emptyset$. So $\psi(g^k, g^{k'}) = 1$ means a maximum redundancy between two subgraphs.

In order to find the optimal subgraph feature set, we need to maintain the minimum redundancy in the selected informative subgraph set $\mathbf{g}$ (which contains m subgraphs)

$$\mathcal{R}(\mathbf{g}) = \frac{1}{m} \sum_{i=1; g_i \in \mathbf{g}}^{m} \mathbb{r}(g_i|\mathbf{g}) = \frac{1}{m \times m} \sum_{i=1; g_i \in \mathbf{g}}^{m} \sum_{j=1; g_j \in \mathbf{g}}^{m} \psi(g_i, g_j) \tag{7}$$

where g_i and g_j, $\{g_i, g_j\} \in \mathbf{g}$, are subgraphs from any graph-view, and $\mathbb{r}(g_i|\mathbf{g}) = 1/m \times \sum_{j=1; g_j \in \mathbf{g}}^{m} \psi(g_i, g_j)$ denotes the redundancy of g_i given $\mathbf{g}$.

5.2 Maximum Discriminative Score and Minimum Redundancy Criterion

To unify the maximum subgraph discriminative score (Sec. 5.1.1) and minimum subgraph redundancy (Sec. 5.1.2) criteria, Eq. (1) can be rewritten as

$$\begin{aligned} \mathbf{g}_\star &= \arg\max_{\mathbf{g} \subseteq \mathcal{S}} \sum_{k=1}^{v} \sum_{g_e^k \in \mathbf{g}^k} \beta^k \hbar(g_e^k, L^k) \\ s.t.\ & \sum\nolimits_{k=1}^{v} \beta^k = 1,\ 0 \leq \beta^k \leq 1,\ |\mathbf{g}| = m \\ & \mathcal{R}(\mathbf{g}) \leq \mathcal{R}(\mathbf{g}'), \mathbf{g}' \subseteq \mathcal{S}; |\mathbf{g}| = |\mathbf{g}'|; \end{aligned} \tag{8}$$

Noticing that, when we fix $\mathbf{g}$ to update $\boldsymbol{\beta}$, for each view k, $\mathcal{I}(\mathbf{g}^k)$ is also fixed. As a result, the solution to $\boldsymbol{\beta}$ is $\beta_k = 1$ corresponding to the maximum $\mathcal{I}(\mathbf{g}^k)$ over different

views, and β for other views will receive 0 weight otherwise. This trivial solution means that only one view is finally selected. Therefore, the performance of this method is equivalent to finding one single best view, which does not meet our objective of exploring complementary knowledge across multiple graph-views. To avoid such trivial solutions, we use $(\beta^k)^r$ (the r^{th} power of β^k) to replace β^k, then we have

$$\begin{aligned} \mathbf{g}_\star = \arg\max_{\mathbf{g}\subseteq\mathcal{S}} \sum_{k=1}^{v} \sum_{g_e^k\in\mathbf{g}^k} (\beta^k)^r \hbar(g_e^k, L^k) \\ s.t.\ \sum_{k=1}^{v} \beta^k = 1,\ 0 \le \beta^k \le 1,\ |\mathbf{g}| = m \\ \mathcal{R}(\mathbf{g}) \le \mathcal{R}(\mathbf{g}'), \mathbf{g}' \subseteq \mathcal{S}; |\mathbf{g}| = |\mathbf{g}'|; \end{aligned} \tag{9}$$

where $r \le -1$. By doing so, $\sum_{k=1}^{v}(\beta^k)^r$ achieves its maximum when $\beta^k = 1/v$ with respect to $\sum_{k=1}^{v}\beta^k = 1$ and $\beta^k \ge 0$. Similarly, β^k for different views can be obtained by setting $r \le -1$. In this case, each view will have its respective contribution to the final feature subset. Furthermore, the solution to Eq. (9) is a nonlinear−nonconvex optimization, which needs to simultaneously optimize $\mathbf{g}$ and $\boldsymbol{\beta} = \{\beta^1, \cdots, \beta^k, \cdots, \beta^v\}$. To the best of our knowledge, no solution exists to find its global optimal. In this paper, we derive an iterative algorithm by using the alternation optimization, which iteratively optimizes $\mathbf{g}$ and $\boldsymbol{\beta}$ in an alternating fashion, to obtain a near-optimal solution.

5.2.1 Optimizing $\mathbf{g}$ *with a Fixed* $\boldsymbol{\beta}$

When weight value of each view $\beta^k \in \boldsymbol{\beta}$ is fixed, the subproblem of maximizing $\mathcal{I}(\mathbf{g})$ in Eq. (1) is equal to finding a subset of subgraphs that can maximize the sum of $(\beta^k)^r\hbar(g_e^k, L^k)$.

Definition 7 *(dScore: Discriminative score)* For any of the graph-view (*e.g.* k^{th} view), given a label information embedded matrix Q^k, with L^k denoting a Laplacian matrix as $L^k = D^k - Q^k$. The informative score of a subgraph g_e^k can be measured by:

$$\mathbb{d}(g_e^k) = (\beta^k)^r\hbar(g_e^k, L^k) = (\beta^k)^r(\boldsymbol{f}_e^k)^\top L^k \boldsymbol{f}_e^k \tag{10}$$

Because the Laplacian matrix L^k is positive semi-definite [17], for any subgraph g_e^k in the k^{th} view, $(\beta^k)^r(\boldsymbol{f}_e^k)^\top L^k \boldsymbol{f}_e^k \ge 0$, *i.e.,* $\mathbb{d}(g_e^k) \ge 0$. In order to find the optimal cross graph-view subgraph set $\mathbf{g}$ which maximizes the criterion $\mathcal{I}(\mathbf{g})$ defined in Eq. (1), we can calculate dScore of each subgraph in $\mathcal{S} = \{g_1, \cdots, g_s\}$ and sort them in a descending order, *i.e.*, $\mathbb{d}(g_1) \ge \mathbb{d}(g_2) \cdots \ge \mathbb{d}(g_s)$ and then collect top-m informative subgraphs $\mathbf{g} = \{g_1, \cdots, g_m\}$ across all views.

Irredundant Subgraph Exploration: To discover subgraphs for evaluation, one straightforward solution is exhaustive enumeration, *i.e.*, all subgraphs in a graph dataset will be enumerated, with their dScore values being calculated for ranking. However, the number of subgraphs grows exponentially with respect to the size of graphs in each view, which makes the exhaustive enumeration approach impractical for real-world data. Furthermore, the redundancy between subgraphs should also be considered to mine subgraphs with maximum informativeness and minimum redundancy.

Alternatively, we employ a Depth-First-Search (DFS) algorithm gSpan [39] to iteratively enumerate subgraphs. The key idea of gSpan is to first assign a unique minimum DFS code to each graph, and then discover all frequent subgraphs by a pre-order traversal of the tree. Specifically, gSpan labels a subgraph with a DFS code and then produces child DFS codes

Algorithm 1 CGVSE: Cross Graph-View Subgraph Exploration

Input:
$\mathcal{G} = \{\mathcal{G}^1, \cdots, \mathcal{G}^k, \cdots, \mathcal{G}^v\}$: A multi-graph-view graph set;
$\boldsymbol{\beta} = \{\beta^1, \cdots, \beta^k, \cdots, \beta^v\}$: A graph-view weight set;
min_sup: The threshold of the frequent subgraph;
m: the number of subgraph features to be selected;
Output:
$\mathbf{g} = \{g_1, \cdots, g_m\}$: A set of cross graph-view subgraphs;
1: $\mathbf{g} = \emptyset, \tau = 0$;
2: **for all** each $\mathcal{G}^k$ in $\mathcal{G}$ **do**
3: **while** Recursively visit the DFS Code Tree in gSpan **do**
4: $g_e^k \leftarrow$ current visited subgraph in DFS Code Tree;
5: **if** $freq(g_e^k) < min\_sup$, **then**
6: **return**;
7: $\mathbf{g} \leftarrow \mathbf{g} \bigcup g_e^k$;
8: **if** $|\mathbf{g}| > m$, **then**
9: Compute the iScore $\mathbb{i}(g_i|\mathbf{g})$ for each $g_i \in \mathbf{g}$ with β^k;
10: $\mathbf{g} \leftarrow \mathbf{g}/\arg\min_{g_i \in \mathbf{g}} \mathbb{i}(g_i|\mathbf{g})$;
11: $\tau = \min_{g_i \in \mathbf{g}} \mathbb{i}(g_i|\mathbf{g})$;
12: **if** $\hat{\mathbb{i}}(g_e^k) \geq \tau$, **then**
13: Depth-first search the subtree rooted from node g_e^k;
14: **end while**
15: **end for**
16: **return** $\mathbf{g}$;

from the right-most path of the DFS Code Tree. If the child DFS code is a minimum DFS code, which is defined by a lexicographic order of the discovery time during the search process, the corresponding graph is processed (*i.e.* the DFS Code Tree, where each node is a subgraph, is obtained). By employing a depth first search strategy on the tree, gSpan can effectively enumerate all frequent subgraphs efficiently.

Definition 8 *(iScore: Irredundant dScore)* Given a cross graph-view subgraph set $\mathbf{g}$, the irredundant dScore of a subgraph g_i, $\mathbb{i}(g_i|\mathbf{g})$, is measured as

$$\mathbb{i}(g_i|\mathbf{g}) = \mathbb{d}(g_i)/\mathbb{r}(g_i|\mathbf{g}) \tag{11}$$

where the dScore $\mathbb{d}(g_i)$ is calculated via Eq. (10) with $\mathbb{r}(g_i|\mathbf{g})$ been calculated by Eq. (7).

In this paper, because subgraph search for each graph-view is independent, we combine discriminative score and redundancy to derive a upper bound for the iScore to prune the search space in the DFS-code tree, which is defined as follows:

Theorem 1 ***Upper bound of iScore:*** *Given two subgraphs g_e^k, $g_e'^k \in \mathcal{S}^k$ in the kth view, $g_e'^k$ is a supergraph of g_e^k (i.e., $g_e'^k \supseteq g_e^k$). The iScore of $g_e'^k$, $\mathbb{i}(g_e'^k|g_e^k)$ given g_e^k, is bounded by $\hat{\mathbb{i}}(g_e^k)$, i.e., $\mathbb{i}(g_e'^k|g_e^k) \leq \hat{\mathbb{i}}(g_e^k)$, with $\hat{\mathbb{i}}(g_e^k)$ been defined as:*

$$\hat{\mathbb{i}}(g_e^k) = (\boldsymbol{f}_e^k)^\top \hat{L}^k \boldsymbol{f}_e^k * (\boldsymbol{f}_e^k)^\top I \boldsymbol{f}_e^k \tag{12}$$

where the matrix $\hat{L}^k$ is defined as $\hat{L}_{ij}^k = max(0, L_{ij}^k)$.

Thus, for any $g_e'^k \supseteq g_e^k$, $\mathbb{i}(g_e'^k) \leq \hat{\mathbb{i}}(g_e^k)$ in the kth view. The corresponding proof is given in **Appendix**.

This upper bound is used to prune DFS-code tree in gSpan by using branch-and-bound pruning. Algorithm 1 lists the proposed dual criterion (*i.e.,* maximum discriminative scores and minimum redundancy) embedded cross graph-view subgraph feature exploration

method, which starts from an empty feature set $\mathbf{g}$ and the minimum iScore $\tau = 0$. The algorithm continuously enumerates subgraphs by recursively visiting the DFS Code Tree of each graph-view dataset, by using the gSpan algorithm. If a subgraph g_e^k in the k^{th} view is not a frequent subgraph, both g_e^k and its subtree will be pruned (line 5-6). Otherwise, if $\mathbf{g}$ has less than m subgraphs (*i.e.,* $\mathbf{g}$ is not full), g_e^k is added to the subgraph set $\mathbf{g}$ (lines 7).

When the size of $\mathbf{g}$ exceeds the predefined value m, the algorithm needs to remove one subgraph with the least discriminative power and also maintains the minimum redundancy in $\mathbf{g}$. Accordingly, we first calculate the iScore $\mathbb{i}(g_i|\mathbf{g})$ of each $g_i \in \mathbf{g}$, and then remove the subgraph with the least iScore value (lines 8-10). After that, the size $\mathbf{g}$ will be reduced to m again, and then we update the iScore for each $g_i \in \mathbf{g}$ to set the minimum iScore $\mathbb{i}(g_i|\mathbf{g})$ as the threshold τ for future process to prune the search space (line 11). Subsequently, the upper bound pruning module will check if $\hat{\mathbb{i}}(g_e^k)$ is less than the threshold τ. If so, it means that the iScore value of any supergraph $g_e'^k$ of g_e^k (*i.e.* $g_e'^k \supseteq g_e^k$) will not be greater than τ. Thus, we can safely prune subtrees rooted from $g_e'^k$ in the search space. If $\hat{\mathbb{i}}(g_e^k)$ is indeed greater than the threshold τ, the depth-first search will continue by following the children of g_e^k (line 12-13), until the mining process is completed.

5.2.2 Optimizing $\boldsymbol{\beta}$ with a Fixed $\mathbf{g}$

After the above process, we fix $\mathbf{g}$ to update $\boldsymbol{\beta}$. For Eq. (9), by using a Lagrange multiplier λ to take the constraint $\sum_{k=1}^{v} \beta^k = 1$ into consideration, we have the Lagrange function as follows,

$$f(\beta, \lambda) = \sum_{k=1}^{v} \sum_{g_e^k \in \mathbf{g}^k \subseteq \mathbf{g}} (\beta^k)^r \hbar(g_e^k, L^k) - \lambda\Big(\sum_{k=1}^{v} \beta^k - 1\Big) \tag{13}$$

By setting the derivative of $f(\beta, \lambda)$ with respect to β^k and λ to zero, we have

$$\begin{aligned} \frac{\partial f(\beta, \lambda)}{\partial \beta^k} &= r(\beta^k)^{r-1} \sum_{g_e^k \in \mathbf{g}^k \subseteq \mathbf{g}} \hbar(g_e^k, L^k) - \lambda = 0 \\ \frac{\partial f(\beta, \lambda)}{\partial \lambda} &= \sum_{k=1}^{v} \beta^k - 1 = 0 \end{aligned} \tag{14}$$

Then, we can obtain β^k accordingly,

$$\beta^k = \frac{\Big(1/\sum_{g_e^k \in \mathbf{g}^k \subseteq \mathbf{g}} \hbar(g_e^k, L^k)\Big)^{1/(r-1)}}{\sum_{k=1}^{v} \Big(1/\sum_{g_e^k \in \mathbf{g}^k \subseteq \mathbf{g}} \hbar(g_e^k, L^k)\Big)^{1/(r-1)}} \tag{15}$$

Because Laplacian matrix L^k is positive semi-definite, for any subgraph g_e^k, $\hbar(g_e^k, L^k) \geq 0$ [17], thus $\beta^k \geq 0$. According to Eq. (15), if $r \to -\infty$, β^k in all views will be close to each other. On the other hand, if $r \to 1$, only $\beta^k = 1$ corresponding to the maximum $\max \mathcal{I}(\mathbf{g})$ across different views, and $\beta^k = 0$ otherwise. Noticing that when $r = 0$, $(\beta^k)^r$ will always be 1. In this case, the view weights are useless. So, in our experiments, we set $r \leq -1$.

Notice that the global alignment objective function $\sum_{k=1}^{v} (\beta^k)^r \sum_{g_e^k \in \mathbf{g}^k \subseteq \mathbf{g}} \hbar(g_e^k, L^k)$ increases when the number of iterations increases, so the discovered $\mathbf{g}$ can always increase the objective function value with the fixed $\boldsymbol{\beta}$, and vice versa. As a result, gCGVFL aims to achieve a near-optimal solution.

Algorithm 2 gCGVFL: Cross Graph-View Feature based Learning

Input:
$\mathcal{G} = \{\mathcal{G}^1, \cdots, \mathcal{G}^k, \cdots, \mathcal{G}^v\}$: A multi-graph-view graph set;
min_sup: The threshold of the frequent subgraph;
m: the number of subgraph features to be selected;
Output:
The target class label y_t of a test multi-graph-view graph G_t;
// Training Phase:
1: $\boldsymbol{\beta} \leftarrow$ The β_k value is initially set to $1/v$.
2: **while not** convergence for $\boldsymbol{\beta}$ **do**
3: $\mathbf{g} \leftarrow CGVSE(\mathcal{G}, \boldsymbol{\beta}, min\_sup, m)$; // Alogirthm 1
4: $\boldsymbol{\beta} \leftarrow$ Apply $\mathbf{g}$ to update $\boldsymbol{\beta}$ via Eq. (15);
5: **end while**
6: $\mathbf{g}_* \leftarrow \mathbf{g}$ // Set of near-optimal cross graph-view features
7: $\mathcal{X}_* \leftarrow$ Apply $\mathbf{g}_*$ to $\mathcal{G}$ to obtain its feature representation.
8: $\mathcal{H} \leftarrow$ Apply $\mathcal{X}_*$ to built a classifier.
// Testing Phase:
9: $\mathbf{x}_t \leftarrow$ multi-graph-view graph representation for G_t.
10: $y_t \leftarrow h(\mathbf{x}_t|\mathcal{H})$.
11: **return** y_t.

Algorithm 2 lists the detailed procedures of using an iterative optimization process to obtain a near-optimal solution for cross graph-view subgraph exploration (lines 2-5 in Algorithm 2). By using the obtained subgraph set $\mathbf{g}$ to represent the graph in vector space, gCGVFL trains a classifier $\mathcal{H}$ (line 8) for graph classification. During the test phase, a multi-graph-view graph G_t is transferred into a feature vector by using $\mathbf{g}$ (line 9), and predicted by the classifier $\mathcal{H}$ to obtain its class label y_t (line 10).

6 Experiments

6.1 Experimental Settings

All reported results are based on 10 times 10-fold cross-validation with classification accuracy being used as the main performance metrics. For comparisons, the baselines and the proposed gCGVFL all use LibSVM, which is popularly used for classification, as the classifier during the learning process. In addition, the parameter r in gCGVFL is set to -1. Unless specified otherwise, the number of selected subgraph features is 100, and the minimum support threshold $min\_sup = 3\%$ for scientific publication dataset (Section 6.3) and $min\_sup = 2\%$ for content-based image retrieval dataset (Section 6.4). All experiments are conducted on a Linux cluster node with an Interl(R) Xeon(R) @3.33GHZ CPU and 3GB fixed memory size.

6.2 Baseline Methods

Because there is no existing multi-graph-view learning method available for graph classification, for comparison purposes we implement three baseline approaches and validate their performance by using three types of view combination strategies. For view combination, we compare the algorithm performance from both single graph-view and combined multiple graph-views, respectively.

6.2.1 Single Graph-View Approaches

In the first type of baseline approaches, we separate a multi-graph-view dataset into multiple graph datasets, each of which containing only one single graph-view. Our purpose is to demonstrate the performance of subgraph feature based learning on a single graph-view and further validate the performance gain/loss of the cross graph-view learning. For the single graph-view based subgraph feature learning baseline approaches, we implement the following methods:

IG: In this method, a set of frequent subgraphs are mined from all graphs in a given single graph-view. An Information Gain (IG) based feature selection criterion is used to select m subgraphs with the highest IG scores as the subgraph features to represent graphs for classification.

Topk: This method is similar to the IG based algorithm, but uses frequency as the measure to select top-k subgraphs with the highest frequency values as m subgraph features in each graph-view.

gHSIC: This is a state-of-the-art discriminative subgraph mining methods, which employs a dependence evaluation criterion named Hilbert-Schmidt Independence Criterion (HSIC). This method has been successfully applied to graph related classification tasks (*e.g.,* active feature selection [16]). Because this method is a recently developed subgraph exploration approach, it will help evaluate the proposed multi-graph-view graph learning algorithm. In our implementation, gHSIC employs the HSIC evaluation criterion to mining m subgraphs for each single graph-view, and uses selected subgraph features to represent graphs for classification.

Our proposed multi-graph-view learning method is denoted by gCGVFL. Meanwhile, in order to evaluate the redundancy checking module of gCGVFL, we also implement a rgCGVFL approach which is identical to gCGVFL except that it does not have redundancy checking. This baseline (rgCGVFL) will help study the impact of feature redundancy on the graph classification (detailed in Section 6.5.1).

6.2.2 Multiple Graph-View Approaches

For comparison purposes, we also implement the following three baseline strategies which concatenate subgraphs discovered from different graph-views for learning. The three types of cross graph-view subgraph feature combination strategies will serve as baselines to

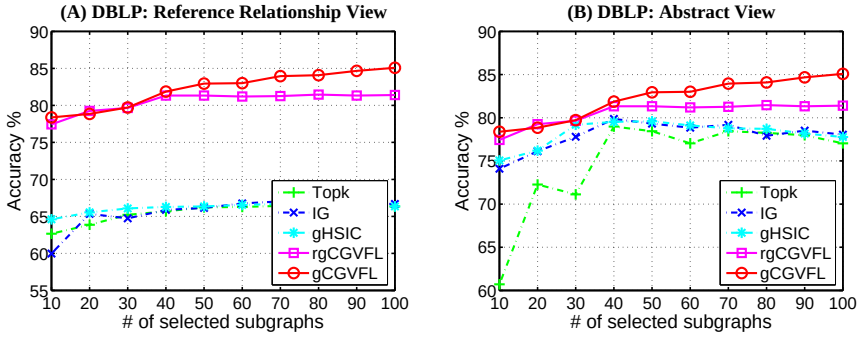


Fig. 5 Comparisons on *DBLP dataset* on each single view: (A) Reference Relationship (B) Abstract.

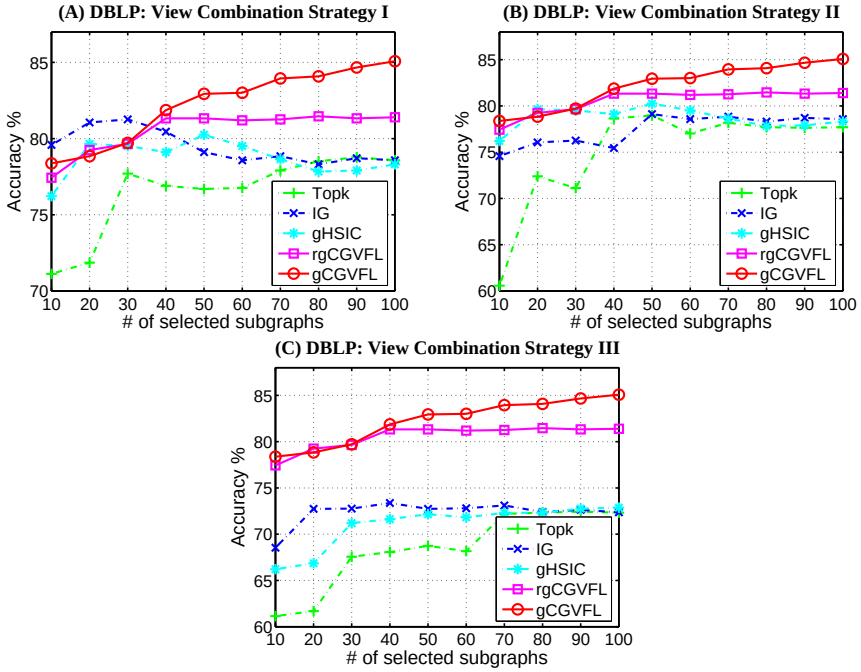


Fig. 6 Comparisons on *DBLP dataset* from multiple views via different view combination methods.

demonstrate whether concatenating subgraph features selected from each single graph-view can result in comparable performance as our proposed method.

View Combination Strategy I: It concatenates the top-m/v subgraph features selected from each view by using IG, Topk, or gHSIC to form concatenated m-dimensional cross graph-view subgraph features. This is similar to a state-of-the-art *filter* based feature selection approach for multi-view learning [28] with the number of features from each single view being pre-specified.

View Combination Strategy II: This type of baseline first concatenates frequent subgraphs enumerated in all graph-views (*i.e.,* the complete frequent subgraph set by treating each view equally) and then directly explores the m-dimensional features via IG, Topk, or gHSIC. This type of baseline does not need to specify the number of subgraph features for each single graph-view.

View Combination Strategy III: We further compare our model with a state-of-the-art view combination strategy [22]. Specifically, after obtaining the top-m/v features via subgraph selection methods, such as the above IG, Topk, or gHSIC, from each single graph-view, a LibSVM classifier is first trained independently for each view and then the classifiers are combined to predict class label of a test graph.

6.3 Scientific Publication Categorization Results

A scientific publication can be represented as a graph by using correlations of keywords in the abstract which contains rich content information [34]. This abstract graph will represent the content and the context information inside the abstract of each publication. In addition

to the abstract graph-view, we also build a second graph for each paper, where each node corresponds to the Paper ID or a keyword appearing in the paper title, and each edge denotes the citation relationship between papers or keyword relations in the title (similar to the approach used in [25]). As a result, each paper is represented by two graphs (*i.e.,* one from abstract graph-view and one from reference citation graph-view).

The Digital Bibliography & Library Project (DBLP) dataset (http://dblp.uni-trier.de/xml/) consists of bibliography data in computer science, where each record contains information such as abstract, authors, year, title, and references. To build a multi-graph-view graph, we select papers published in three main fields: Artificial Intelligence (AI: IJCAI, AAAI, NIPS, UAI, COLT, ACL, KR, ICML, ECML and IJCNN), Computer Vision (CV: ICCV, CVPR, ECCV, ICPR, ICIP, ACM Multimedia and ICME), and Database (DB: SIGMOD, PODS, VLDB, ICDE, CIKM, DASFAA, ICDT, SSD, DASFAA) to form a three class multi-graph-view graph classification task. The objective is to predict whether a paper belongs to the AI, CV or DB field by using the multiple graph-view representation. For each abstract in the abstract graph-view, a fuzzy cognitive map (E-FCM) [21] based approach is used to extract a number of keywords and correlations between keywords. In our experiments, we use keywords as nodes and correlations between two keywords as edge weight values to build a graph. A threshold (0.005) was used to remove edges whose correlation values are less than the threshold. At the last step, the graph is converted into an unweighted graph by setting the weight values of all remaining edges as "1". The same graph representation was also used in previous works [32, 34]. In our experiments, we choose 1500 papers with each class (AI, CV, or DB) containing 500 multi-graph-view graphs.

Figures 5(A) and 5(B) report the accuracy comparisons on each single graph-view (*i.e.,* reference relationship graph-view and abstract graph-view) on the DBLP dataset. Clearly, the proposed gCGVFL achieves the best performance in all case on the reference relationship view. For the abstract view, although IG and gHSIC based approaches have similar accuracies, they are both inferior to the proposed gCGVFL. Moreover, the results in Figures 6(A)-6(C) show the performance comparisons using cross graph-view subgraphs with different view combination strategies. In some cases, we noticed that gCGVFL is less accurate than IG based approach with view combination strategy I, when the number of selected subgraphs is small (*i.e.,* ≤ 30). Nevertheless gCGVFL can achieve a much better performance than the baselines when the number of cross graph-view subgraphs increase. This is mainly because that when the size of the subgraph feature set increases, it becomes easier for the feature set $\mathbf{g}$ to attract redundant subgraph features. As a result, the redundant subgraph

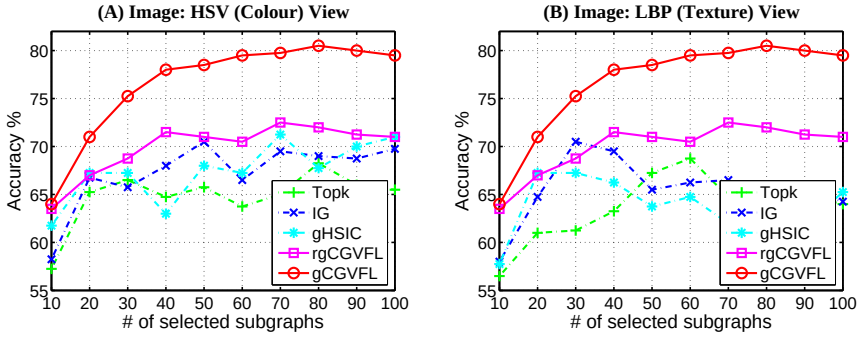


Fig. 7 Comparisons on *Image dataset* on each single view: (A) HSV (Colour) (B) LBP (Texture).

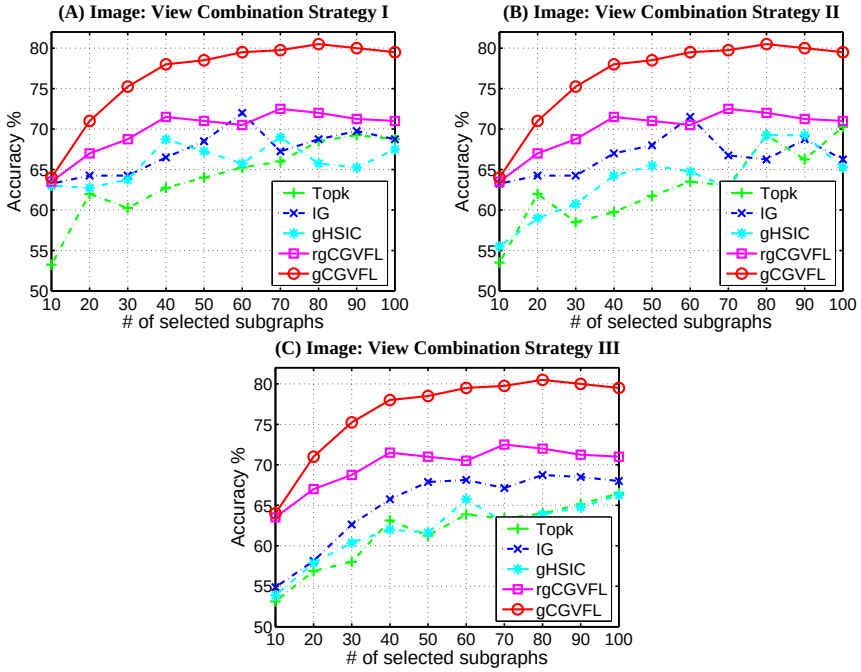


Fig. 8 Comparisons on *Image dataset* from multiple views via different view combination methods.

features will reduce the classification accuracies. By using the informative-irredundant dual subgraph evaluation criterion in gCGVFL, the quality of the cross graph-view subgraphs can be improved.

In Section 6.5.1, we will carry out irredundancy analysis to assess the quality of the selected subgraph set $\mathbf{g}$ from the informativeness scores and the redundancy perspectives.

6.4 Content-based Image Retrieval Results

In this section, we report gCGVFL's performance for content based image retrieval. The original images [20] from Corel dataset[1] are preprocessed and segmented using Statistical Region Merging (SRM) [24], which has achieved a better performance on the Berkeley Segmentation Database compared to state-of-the-art techniques. In this case, each image is considered as a graph consisting of a set of nodes and edges which correspond to image regions and the adjacency between two regions, respectively.

In order to build multiple graph-view representation for each image, we employ different types of features [31, 11], including Hue-Saturation-Value (HSV) in colour space and Local Binary Patterns (LBP) in the texture space, to construct graphs with different feature views. Specifically, HSV is a commonly used colour model, where HSV stands for hue, saturation and intensity, and LBP is a model to represent texture in a local region. For HSV feature, we first extract a 3-channel HSV feature for each pixel. The extracted HSV representations are fed to a k-means clustering algorithm to construct a 256-dimensional codebook. After

[1] https://sites.google.com/site/dctresearch/Home/content-based-image-retrieval

that, a one-dimensional code is assigned to each pixel based on the similarity between the pixel representation and the cluster centers. The HSV-based representation for a region is constructed as a 256-dimensional histogram-based vector by computing the statistics of the occurrences of the codes. For the LBP representation, we adopt the uniform LBP and generate a 59-bin code for each pixel, where each pixel is assigned to one bin according to the local texture pattern. Subsequently, a 59-dimensional histogram representation is constructed for each region encoding its statistics.

Furthermore, we collect categories "Tiger", "Lizard", "Eagle", and "Porp" to form a four-class classification problem containing 400 multi-graph-view graphs (100 for each class). Figures 7 and 8 report the accuracy performance comparison on each single graph-view and multiple graph-views with selected subgraph features varying from 10 to 100, respectively. Overall, the IG related baselines show the best performance compared to all other baselines. However, they are all consistently inferior to the proposed gCGVFL, which explores most informative subgraph features with minimum redundancy for classification.

6.5 Detailed Algorithm Performance Studies

6.5.1 Effectiveness of Irredundancy checking in gCGVFL

One of the unique features of the proposed gCGVFL is that it uses a minimum redundancy criterion to ensure the quality of the selected discriminative subgraphs. In order to understand the actual role of the minimum redundancy criterion in the gCGVFL, we implement a rgCGVFL approach which is identical to gCGVFL, except that it does not have redundancy checking module. More specifically, rgCGVFL selects the optimal subgraph set by directly using the discriminative evaluation criteria (detailed in Section 5.1.1). We empirically compare gCGVFL and rgCGVFL throughout the whole experiments, so we can clearly understand the efficiency of the redundancy checking module. The experimental results on both DBLP dataset (as shown in Figures 5 and 6) and Image dataset (as shown in Figures 7 and 8) show that gCGVFL significantly outperforms rgCGVFL, which demonstrates that redundancy has a significant impact on the learning performance. Without considering the redundancy among the selected subgraph features, rgCGVFL may be inferior to the baseline approaches.

6.5.2 Algorithm Convergence Study

In order to obtain the final informative-irredundant subgraph feature set $\mathbf{g}$, gCGVFL employs a first-order incremental search strategy. Specifically, suppose the subgraph set $\mathbf{g}$ is full (*i.e.,* containing m subgraphs), gCGVFL needs to update the subgraph set by using the next explored subgraph candidate g_i. It first adds g_i to $\mathbf{g}$, and calculates the iScore $\mathtt{i}(g_i|\mathbf{g}) = \mathtt{d}(g_i)/\mathtt{r}(g_i|\mathbf{g})$ for each subgraph. After that, the subgraph with smallest iScore is removed in order to ensure $\mathbf{g}$ only contains m subgraphs. By doing so, we expect that the final set $\mathbf{g}$ will achieve the near-optimal solutions with maximum discriminative scores and minimum redundancy. In order to verify that gCGVFL's greedy search strategy can indeed result in continuously improved subgraph feature set with an increasing discriminative scores and decreasing redundancy (*i.e.*, gCGVFL can reach the convergence), we further report the detailed discriminative scores and the redundancy of the subgraph set $\mathbf{g}$ with respect to the number of explored subgraph candidates in Figure 9(a). In our experiments, we record the order of the subgraph candidates explored by the subgraph mining process (*i.e.* the g_e^k

outputs from Step 4 in Algorithm 1), and report the discriminative scores (left $y-$axis) and the redundancy (right $y-$axis) of $\mathbf{g}$ with respect to the order of the subgraph candidates ($x-$ axis). In addition, we also report the discriminative score of g_e^k in Figure 9(b) where the $x-$axis has the same meaning as Figure 9(a). In other words, for each g_e^k from Step 4 in Algorithm 1, we immediately calculate its discriminative scores (Figure 9(b)), and also report the corresponding subgraph feature set $\mathbf{g}$'s discriminative scores and redundancy in Figure 9(a).

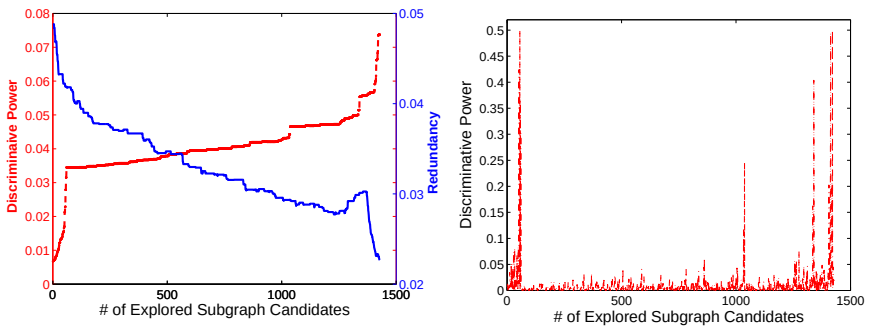
The results from Figure 9(a) confirm that with the number of subgraph candidates increase, the discriminative power of $\mathbf{g}$ can be continuously enhanced, and the corresponding redundancy is also reduced, which empirically confirms the convergence of the algorithm.

One interesting finding is that when comparing results from Figures 9(a) and 9(b), we can find that informative subgraphs do appear in clusters. In other words, if a subgraph has a high discriminative score, its descendent or sibling may also have a high discriminative score (shown as group of spikes in Figure 9(b)). By employing the redundancy checking module, gCGVFL can effectively include the good subgraphs and reduce the redundancy when multiple correlated good subgraphs appear (shown as sharp increase of the discriminative scores and the dramatic decrease of the redundancy in Figure 9(a)). This observation not only demonstrates gCGVFL's convergence but also confirms its effectiveness in finding high quality cross graph-view subgraph features for graph classification.

6.5.3 Efficiency of the Pruning Strategy

In this subsection, we evaluate the efficiency of the proposed pruning module for searching subgraphs in gCGVFL as described in Section 5.2.1. For comparisons, we implement a UgCGVFL approach with no pruning module and compare its runtime performance with gCGVFL, through which we can demonstrate the efficiency of the pruning module. In our implementation, UgCGVFL first uses gSpan to find a frequent subgraph set, and then selects the optimal subgraph set by using the same evaluation criteria as gCGVFL.

In Figures 10(A) and 10(B), we report the average CPU runtime performance with respect to different minimum support min_sup values, with the number of selected subgraphs being fixed to 100 (with the default settings as in Section 6.1), on the *DBLP* and *Image* datasets, respectively. The results show that as the min_sup values decrease, the runtime



(a) (b)

Fig. 9 (a) Discriminative capability of selected subgraphs and the corresponding redundancy, and (b) Discriminative capability for the subgraph updating candidate, respectively.

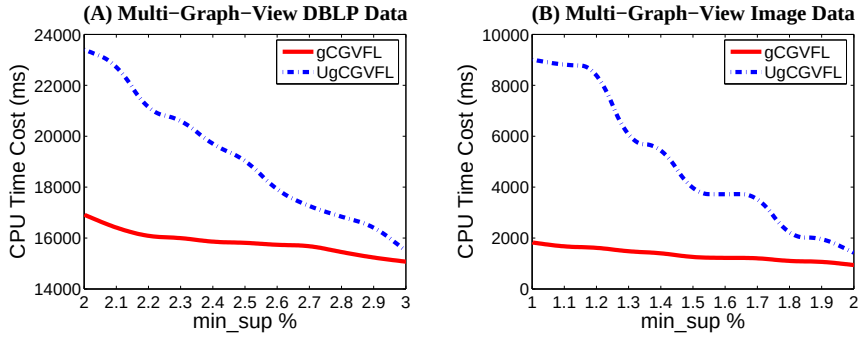


Fig. 10 Average CPU runtime comparison between gCGVFL *v.s.* unbounded gCGVFL (UgCGVFL) with respect to different min_sup values on *DBLP* (A) and *Image* (B) dataset, respectively.

of unbounded UgCGVFL increase dramatically, mainly because a smaller min_sup value will substantially increase the number of candidates for validation. By using pruning strategy (*i.e.,* the upper bound $\tau = \min_{g_i \in \mathbf{g}} \mathtt{i}(g_i|\mathbf{g})$ for each multi-graph-view as shown in Algorithm 1), gCGVFL's runtime performance is relatively stable with respect to different support min_sup values. Overall, gCGVFL demonstrates clear advantage compared to unbounded UgCGVFL, especially when the minimum support value min_sup is small. For example, on multi-graph-view Image dataset as shown in Figure 10(B), UgCGVFL needs about 9000ms to mine the discriminative subgraphs, whereas by using the upper bound pruning gCGVFL only takes about 2000ms, which results in a remarkable efficiency gain for gCGVFL.

6.5.4 Study of the Effect of Parameter r

Table 1 reports the performance of gCGVFL with respect to different r values. The results show that when $r = -1$ and -2, gCGVFL has the best performance on the Image dataset, and the best performance on DBLP dataset is achieved when $r = -10$. As discussed in the last paragraph of Section 5.2: r values are directly related to the view correlations. Our experiments indicate that the above two datasets have similar complementary information across different views. Our experiments show that the above two datasets have similar complementary information among different views. Overall, the accuracy for DBLP dataset is relatively less stable for different r values, indicating that the impact of the r values for DBLP dataset is larger than that for Image dataset.

Table 1 Comparison of gCGVFL *w.r.t.* different r: %.

Data	r=-1	r=-2	r=-3	r=-4	r=-5	r=-6	r=-7	r=-8	r=-9	r=-10
Image	**79.0**	**79.0**	78.8	78.8	78.8	78.8	78.8	78.8	78.8	78.8
DBLP	85.0	84.5	84.8	84.7	84.6	84.6	84.6	84.6	84.6	**85.2**

7 Conclusion

This paper investigated a new multi-graph-view learning task for graph classification, where the object is a graph containing multiple graph-views. We argued that many real-world applications involve graph structured data with multiple graph-views, where each graph-view provides valuable information to represent the content and structure relationships of the object. To build a learning model for multi-graph-view classification, we first proposed a dual criterion to find the cross graph-view subgraph set with maximum discriminative scores and minimum redundancy. After that, an iterative optimization framework was proposed to improve the quality of the selected informative-irredundent subgraphs across all views for graph classification. Experiments and comparisons on real-world tasks showed that the proposed cross graph-view gCGVFL approach significantly outperforms baseline methods.

Acknowledgments

The work was supported by the Key Project of the Natural Science Foundation of Hubei Province, China (Grant No. 2013CFA004), and the National Scholarship for Building High Level Universities, China Scholarship Council (No. 201206410056), and National Natural Science Foundation of China (Grant No. 61403351 and 61370025), the Chinese National "111" Project hosted by SA Centre for Big Data Research in Renmin University of China. It is also partially supported by the Australian Research Council Discovery Projects under Grant No. DP140100545 and DP140102206.

Appendix: Proof of the Theorem 1

Theorem 1 ***Upper bound of iScore:*** *Given two subgraphs g_e^k, $g_e'^k \in \mathcal{S}^k$ in the kth view, $g_e'^k$ is a supergraph of g_e^k (i.e., $g_e'^k \supseteq g_e^k$). The iScore of $g_e'^k$, $\mathbb{i}(g_e'^k|g_e^k)$ given g_e^k, is bounded by $\hat{\mathbb{i}}(g_e^k)$, i.e., $\mathbb{i}(g_e'^k|g_e^k) \leq \hat{\mathbb{i}}(g_e^k)$, with $\hat{\mathbb{i}}(g_e^k)$ been defined as:*

$$\hat{\mathbb{i}}(g_e^k) = (\boldsymbol{f}_e^k)^\top \hat{L}^k \boldsymbol{f}_e^k * (\boldsymbol{f}_e^k)^\top I \boldsymbol{f}_e^k \tag{16}$$

where the matrix $\hat{L}^k$ is defined as $\hat{L}_{ij}^k = max(0, L_{ij}^k)$.

Proof

$$\mathbb{i}(g_e'^k|g_e^k) = \mathbb{d}(g_e'^k)/\mathbb{r}(g_e'^k|g_e^k) \tag{17}$$

For $\mathbb{d}(g_e'^k)$, $0 \leq \beta^k \leq 1$, we have

$$\mathbb{d}(g_e'^k) \leq (\boldsymbol{f}_e'^k)^\top L^k \boldsymbol{f}_e'^k = \sum_{i,j:G_i^k, G_j^k \in \mathcal{G}(g_e'^k)} L_{ij}^k \tag{18}$$

where $\mathcal{G}(g_e'^k) = \{G_i^k | g_e'^k \subseteq G_i^k, 1 \leq i \leq p^k\}$. $\boldsymbol{f}_e'^k$, an indicator vector of subgraph $g_e'^k$ with respect to all graphs in $\mathcal{G}^k$ in the kth view, is similar with $\boldsymbol{f}_e^k$ denoted as above. Because $g_e'^k$ is the supergraph of g_e^k (*i.e.* $g_e'^k \supseteq g_e^k$), according to the anti-monotonic property, we have $\mathcal{G}(g_e'^k) \subseteq \mathcal{G}(g_e^k)$. Besides, and $\hat{L}_{ij}^k = max(0, L_{ij}^k)$, so $\hat{L}_{ij}^k \geq L_{ij}^k$ and $\hat{L}_{ij}^k \geq 0$.

Thus, Eq. (18) could be rewritten as

$$\begin{aligned}\mathbb{d}(g_e'^k) &= \sum_{i,j:G_i^k,G_j^k \in \mathcal{G}(g_e'^k)} L_{ij}^k \leq \sum_{i,j:G_i^k,G_j^k \in \mathcal{G}(g_e'^k)} \hat{L}_{ij}^k \\ &\leq \sum_{i,j:G_i^k,G_j^k \in \mathcal{G}(g_e^k)} \hat{L}_{ij}^k = (\boldsymbol{f}_e^k)^\top \hat{L}^k \boldsymbol{f}_e^k\end{aligned} \tag{19}$$

Moreover, for $1/\mathbb{r}(g_e'^k|g_e^k)$ in $\mathbb{i}(g_e'^k|g_e^k)$, because $g_e'^k \supseteq g_e^k$, we have $\mathcal{G}(g_e'^k) \subseteq \mathcal{G}(g_e^k)$, so $\mathcal{G}(g_e'^k) \cap \mathcal{G}(g_e^k) = \mathcal{G}_\cap = \mathcal{G}(g_e'^k)$ and $\mathcal{G}(g_e'^k) \cup \mathcal{G}(g_e^k) = \mathcal{G}_\cup = \mathcal{G}(g_e^k)$. Meanwhile, $\mathcal{G}(g_e'^k)$ and $\mathcal{G}(g_e^k)$ are not empty, because in our irredundant subgraph exploration only frequent subgraph will be considered for further selection. According to the Eqs. (6) and (7), we have

$$\begin{aligned}1/\mathbb{r}(g_e'^k|g_e^k) &= 1 \Big/ \left(\frac{\sum_{g^k \subseteq \mathcal{G}^k \in \mathcal{G}(g_e'^k)} \beta^k}{\sum_{g^k \subseteq \mathcal{G}^k \in \mathcal{G}(g_e^k)} \beta^k} \right) \\ &= \sum_{g^k \subseteq \mathcal{G}^k \in \mathcal{G}(g_e^k)} 1 \Big/ \sum_{g^k \subseteq \mathcal{G}^k \in \mathcal{G}(g_e'^k)} 1 \\ &\leq \sum_{g^k \subseteq \mathcal{G}^k \in \mathcal{G}(g_e^k)} 1 = (\boldsymbol{f}_e^k)^\top I \boldsymbol{f}_e^k;\end{aligned} \tag{20}$$

By unifying the numerator $\mathbb{d}(g_e'^k)$ in Eq. (19) and denominator $\mathbb{r}(g_e'^k|g_e^k)$ in Eq. (20) of $\hat{\mathbb{i}}(g_e^k)$, we have

$$\mathbb{i}(g_e'^k|g_e^k) \leq (\boldsymbol{f}_e^k)^\top \hat{L}^k \boldsymbol{f}_e^k * (\boldsymbol{f}_e^k)^\top I \boldsymbol{f}_e^k = \hat{\mathbb{i}}(g_e^k) \tag{21}$$

Thus, for any $g_e'^k \supseteq g_e^k$, $\mathbb{i}(g_e'^k) \leq \hat{\mathbb{i}}(g_e^k)$ in the kth view.

References

1. Aggarwal, C.C.: On classification of graph streams. In: Proceedings of the SIAM International Conference on Data Mining, SDM, pp. 652–663 (2011)
2. Aggarwal, C.C., Ta, N., Wang, J., Feng, J., Zaki, M.: Xproj: A framework for projected structural clustering of xml documents. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, pp. 46–55 (2007)
3. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th Annual Conference on Computational Learning Theory, COLT, pp. 92–100 (1998)
4. Borgelt, C., Berthold, M.: Mining molecular fragments: Finding relevant substructures of molecules. In: Proceedings of the IEEE International Conference on Data Mining, ICDM, pp. 51–58 (2002)
5. Cheng, H., Lo, D., Zhou, Y., Wang, X., Yan, X.: Identifying bug signatures using discriminative graph mining. In: Proceedings of the 18th International Symposium on Software Testing and Analysis, ISSTA, pp. 141–152 (2009)
6. Deshpande, M., Kuramochi, M., Wale, N., Karypis, G.: Frequent substructure-based approaches for classifying chemical compounds. Knowledge and Data Engineering, IEEE Transactions on. **17**, 1036–1050 (2005)
7. Fang, Z., Zhang, Z.M.: Discriminative feature selection for multi-view cross-domain learning. In: Proceedings of the 22Nd ACM International Conference on Conference on Information and Knowledge Management, CIKM, pp. 1321–1330 (2013)
8. Fei, H., Huan, J.: Structure feature selection for graph classification. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM, pp. 991–1000 (2008)
9. Gretton, A., Bousquet, O., Smola, A., Schölkopf, B.: Measuring statistical dependence with hilbert-schmidt norms. In: Proceedings of the 16th International Conference on Algorithmic Learning Theory, ALT, pp. 63–77 (2005)
10. Harchaoui, Z., Bach, F.: Image classification with segmentation graph kernels. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 1–8 (2007)

11. Hong, Z., Wang, C., Mei, X., Prokhorov, D., Tao, D.: Tracking using multilevel quantizations. In: Proceedings of the European Conference on Computer Vision, ECCV, pp. 155–171 (2014)
12. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraphs in the presence of isomorphism. In: Proceedings of the IEEE International Conference on Data Mining, ICDM, p. 549552 (2003)
13. Inokuchi, A., Washio, T., Motoda, H.: An apriori-based algorithm for mining frequent substructures from graph data. In: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, PKDD, pp. 13–23 (2000)
14. Jin, N., Young, C., Wang, W.: Graph classification based on pattern co-occurrence. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM, pp. 573–582 (2009)
15. Jin, N., Young, C., Wang, W.: Gaia: graph classification using evolutionary computation. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD, pp. 879–890 (2010)
16. Kong, X., Fan, W., Yu, P.S.: Dual active feature and sample selection for graph classification. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, pp. 654–662 (2011)
17. Kong, X., Yu, P.: Semi-supervised feature selection for graph classification. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, pp. 793–802 (2010)
18. Kong, X., Yu, P.S.: gmlc: A multi-label feature selection framework for graph classification. Knowledge and Information Systems **31**(2), 281–305 (2012)
19. Kuramochi, M., Karypis, G.: Frequent subgraph discovery. In: Proceedings of the IEEE International Conference on Data Mining, ICDM, pp. 313–320 (2001)
20. Li, J., Wang, J.Z.: Real-time computerized annotation of pictures. Pattern Analysis and Machine Intelligence, IEEE Transactions on **30**, 985–1002 (2008)
21. Luo, X., Z.X., Yu, J., Chen, X.: Building association link network for semantic link on web resources. Automation Science and Engineering, IEEE Transactions on **8**(3) (2011)
22. Mayo, M., Frank, E.: Experiments with multi-view multi-instance learning for supervised image classification. In: IVCNZ, pp. 363–369 (2011)
23. Nijssen, S., Kok, J.: A quickstart in frequent structure mining can make a difference. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, pp. 647–652 (2004)
24. Nock, R., Nielsen, F.: Statistical region merging. Pattern Analysis and Machine Intelligence, IEEE Transactions on **26**(11) (2004)
25. Pan, S., Wu, J., Zhu, X., Zhang, C.: Graph ensemble boosting for imbalanced noisy graph stream classification. Cybernetics, IEEE Transactions on **45**(5), 940–954 (2015)
26. Phillips, W., Riloff, E.: Exploiting strong syntactic heuristics and co-training to learn semantic lexicons. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, EMNLP, pp. 125–132 (2002)
27. Saigo, H., Krämer, N., Tsuda, K.: Partial least squares regression for graph mining. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, pp. 578–586 (2008)
28. Tang, J., Hu, X., Gao, H., Liu, H.: Unsupervised feature selection for multi-view data in social media. In: Proceedings of the 13th SIAM International Conference on Data Mining, SDM, pp. 270–278 (2013)
29. Thoma, M., Cheng, H., Gretton, A., Han, J., Kriegel, H., Smola, A., Song, L., Yu, P., Yan, X., Borgwardt, K.: Near-optimal supervised feature selection among frequent subgraphs. In: Proceedings of the SIAM International Conference on Data Mining, SDM, pp. 1075–1086 (2009)
30. Wu, J., Hong, Z., Pan, S., Zhu, X., Cai, Z., Zhang, C.: Multi-graph-view learning for graph classification. In: Proceedings of the 14th IEEE International Conference on Data Mining, ICDM, pp. 590–599 (2014)
31. Wu, J., Hong, Z., Pan, S., Zhu, X., Zhang, C., Cai, Z.: Multi-graph learning with positive and unlabeled bags. In: Proceedings of the SIAM International Conference on Data Mining, SDM, pp. 217–225 (2014)
32. Wu, J., Pan, S., Zhu, X., Cai, Z.: Boosting for multi-graph classification. Cybernetics, IEEE Transactions on **45**(3), 430–443 (2015)
33. Wu, J., Zhu, X., Zhang, C., Cai, Z.: Multi-instance multi-graph dual embedding learning. In: Proceedings of the 13th IEEE International Conference on Data Mining, ICDM, pp. 827–836 (2013)
34. Wu, J., Zhu, X., Zhang, C., Yu, P.: Bag constrained structure pattern mining for multi-graph classification. Knowledge and Data Engineering, IEEE Transactions on **26**(10), 2382–2396 (2014)
35. Xia, T., Tao, D., Mei, T., Zhang, Y.: Multiview spectral embedding. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **40**(6), 1438–1446 (2010)
36. Xu, C., Tao, D., Xu, C.: A survey on multi-view learning. arXiv preprint arXiv:1304.5634 (2013)
37. Yan, R., Naphade, M.: Semi-supervised cross feature learning for semantic concept detection in videos. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, pp. 657–663 (2005)

38. Yan, X., Cheng, H., Han, J., Yu, P.S.: Mining significant graph patterns by leap search. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD, pp. 433–444 (2008)

39. Yan, X., Han, J.: gspan: Graph-based substructure pattern mining. In: Proceedings of the IEEE International Conference on Data Mining, ICDM, pp. 721–724 (2002)

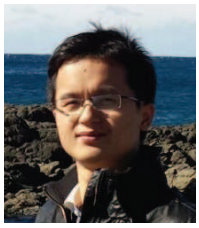
40. Yu, J., Liu, D., Tao, D., Seah, H.S.: On combining multiple features for cartoon character retrieval and clip synthesis. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **42**(5), 1413–1427 (2012)

41. Zhao, X., Xiao, C., Zhang, W., Lin, X., Tang, J.: Improving performance of graph similarity joins using selected substructures. In: Proceedings of the 19th International Conference on Database Systems for Advanced Applications, DASFAA, pp. 156–172 (2014)

Author Biographies

Jia Wu received his bachelor degree in computer science from China University of Geosciences (CUG), Wuhan, China, in 2009. Since September 2009, he has been working toward the PhD degree under the Master-Doctor combined program in computer science from CUG. Besides, he is also pursuing a PhD degree in the Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney (UTS), Australia. His research focuses on data mining and machine learning.

Zhibin Hong received the Bachelors degree in Electronic Engineering from South China University of Technology (SCUT), China, in 2010. He has been a PhD candidate in Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney (UTS), Australia, since 2011. His research interests include computer vision, machine learning and data mining. He is a student member of the IEEE.



Shirui Pan received his master degree in computer science from Northwest A&F University, Yangling, Shaanxi, China, in 2011. Since September 2011, he has been working toward the PhD degree in the Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney (UTS). His research focuses on data mining and machine learning.

Xingquan Zhu received the PhD degree in computer science from Fudan University, Shanghai, China. He is an associate professor in the Department of Computer & Electrical Engineering and Computer Science, Florida Atlantic University. Prior to that, he was with the Centre for Quantum Computation & Intelligent Systems, University of Technology, Sydney, Australia. His research interests mainly include data mining, machine learning, and multimedia systems. Since 2000, he has published more than 180 refereed journal and conference papers in these areas, including two Best Paper Awards and one Best Student Paper Award. Dr. Zhu is an associate editor of the IEEE *Transactions on Knowledge and Data Engineering* (2008-2012, 2014-date), and is currently serving on the Editor Board of International Journal of Social Network Analysis and Mining SNAM (2010-date) and Network Modeling Analysis in Health Informatics and Bioinformatics Journal (2014-date). He served or is serving as a program committee co-chair for BIBE-2014, GRC-2013, ICTAI-2011, and ICMLA-2010. He also served as a conference co-chair for ICMLA-2012. He is a senior member of the IEEE.

Zhihua Cai received the B.Sc. degree from Wuhan University, Wuhan, China, in 1986, the M.Sc. degree from Beijing University of Technology, Beijing, China, in 1992, and the Ph.D. degree from China University of Geosciences, Wuhan, in 2003. He is currently a faculty member with the School of Computer Science, China University of Geosciences. He has published over 50 research papers in journals and international conferences, such as IEEE *Transactions on Knowledge and Data Engineering*, IEEE *Transactions on Cybernetics*, *Applied Soft Computing*, *Information Sciences*, *Knowledge-Based Systems*, *Knowledge and Information Systems*, etc. His main research areas include data mining, machine learning, evolutionary computation, and their applications.

Chengqi Zhang received the Ph.D. degree from University of Queensland, Brisbane, Australia, in 1991 and the D.Sc. degree (higher doctorate) from Deakin University, Geelong, Australia, in 2002. Since December 2001, he has been a Professor of Information Technology with the University of Technology, Sydney (UTS), Sydney, Australia, where he has been the Director of the UTS Priority Investment Research Centre for Quantum Computation and Intelligent Systems since April 2008. Since November 2005, he has been the Chairman of the Australian Computer Society National Committee for Artificial Intelligence. He has published more than 200 research papers, including several in first-class international journals, such as the *Artificial Intelligence*, *IEEE*, and *ACM Transactions*. He has published six monographs and edited 16 books, and has attracted 11 Australian Research Council grants. His research interests mainly focus on data mining and its applications. He has been serving as an Associate Editor for three international journals, including IEEE *Transactions on Knowledge and Data Engineering* (2005-2008); and he served as General Chair, PC Chair, or Organising Chair for five international Conferences including ICDM 2010 and WI/IAT 2008. He is also General Co-Chair of KDD 2015 in Sydney and the Local Arrangements Chair of IJCAI-2017 in Melbourne. He is a Fellow of the Australian Computer Society and a Senior Member of the IEEE.