Multiple Structure-View Learning for Graph Classification

Jia Wu, *Member, IEEE*, Shirui Pan, *Member, IEEE*, Xingquan Zhu, *Senior Member, IEEE*, Chengqi Zhang, *Senior Member, IEEE*, and Philip S. Yu, *Fellow, IEEE*

***Abstract*— Many applications involve objects containing structure and rich content information, each describing different feature aspects of the object. Graph learning and classification is a common tool for handling such objects. To date, existing graph classification has been limited to the single-graph setting with each object being represented as one graph from a single structure-view. This inherently limits its use to the classification of complicated objects containing complex structures and uncertain labels. In this paper, we advance graph classification to handle multigraph learning for complicated objects from multiple structure views, where each object is represented as a bag containing several graphs and the label is only available for each graph bag but not individual graphs inside the bag. To learn such graph classification models, we propose a multistructure-view bag constrained learning (MSVBL) algorithm, which aims to explore substructure features across multiple structure views for learning. By enabling joint regularization across multiple structure views and enforcing labeling constraints at the bag and graph levels, MSVBL is able to discover the most effective substructure features across all structure views. Experiments and comparisons on real-world data sets validate and demonstrate the superior performance of MSVBL in representing complicated objects as multigraph for classification, e.g., MSVBL outperforms the state-of-the-art multiview graph classification and multiview multi-instance learning approaches.**

***Index Terms*— Graph, graph classification, multiview learning, subgraph mining.**

I. Introduction

MANY real-world objects, such as chemical compounds in biopharmacy and proteins in molecular biology [1], images in Web pages [2], brain regions in brain networks [3], and users in social networks [4], contain rich features and structure information. In many cases, these objects are represented by using features in the vector space, such as amino acid sequences to represent a protein, bag-of-words to represent a document, and color histogram to represent an image. In practice, simple feature-vector representations inherently discard the structure information of the object, such as the chemical bounds that regulate the attraction of atoms for chemical compounds, the spatial correlations of regions inside an image [5], and the contextual correlation of keywords for a document [6]. Alternatively, a structural-representation (e.g., graph) can be used to preserve the structure information.

When representing the structure of objects for learning, existing methods often use graphs constructed from a single feature view. For example, an image (i.e., an object) can be represented as a single structure-view graph by using color histogram as features, with each node denoting a small region and adjacent regions being connected through an edge [2], as shown in Fig. 1(a). Nevertheless, using graphs from an individual structure-view may not adequately describe the object's content. For instance, color and texture have different visual characteristics, and are both commonly utilized to describe images. Therefore, using graphs constructed from multiple feature views can accurately represent the structure and the content of the object, and an example is shown in Fig. 2. The multiple structure-view settings can be generalized to many *other domains*, such as brain network analysis, where a brain network can be represented by graphs from different properties, encoding correlations between the functional activities of brain regions [3]. In this paper, we refer to graphs constructed from multiple structure views as *multistructure-view* (MSV) graphs.

Real-world objects often have complicated characteristics, depending on how they are assessed and characterized. For example, an image may be labeled as "leopard/tiger," because it contains a leopard/tiger inside the image. Arguably, not all regions of the image are relevant to the object and background regions may not be directly related to the label of the image, as shown in Fig. 1(b). This representation and learning complication is known as *multi-instance* learning [8]. The uniqueness of handling the label ambiguity (i.e., the label information is not required for each single instance) makes the multi-instance representation applicable to plenty of real-world practical applications.

Most existing multi-instance studies focus on instances with feature vectors. An alternative way to preserve the structure of the object is to represent the object (e.g., an image) as a bag of graphs, as shown in Fig. 1(c), with each graph representing

Manuscript received May 21, 2016; revised October 24, 2016; accepted December 31, 2016. This work was supported in part by the Australian Research Council Discovery Projects under Grant DP140100545 and Grant DP140102206, and in part by the U.S. National Science Foundation under Grant III-1526499, Grant CNS-1115234, and Grant CNS-1626432. *(Corresponding author: Shirui Pan.)*

J. Wu is with Department of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, NSW 2019, Australia (e-mail: jia.wu@mq.edu.au).

S. Pan and C. Zhang are with the Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: shirui.pan@uts.edu.au; chengqi.zhang@uts.edu.au).

X. Zhu is with the Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431 USA (e-mail: xzhu3@fau.edu).

P. S. Yu is with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607 USA (e-mail: psyu@cs.uic.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2703832

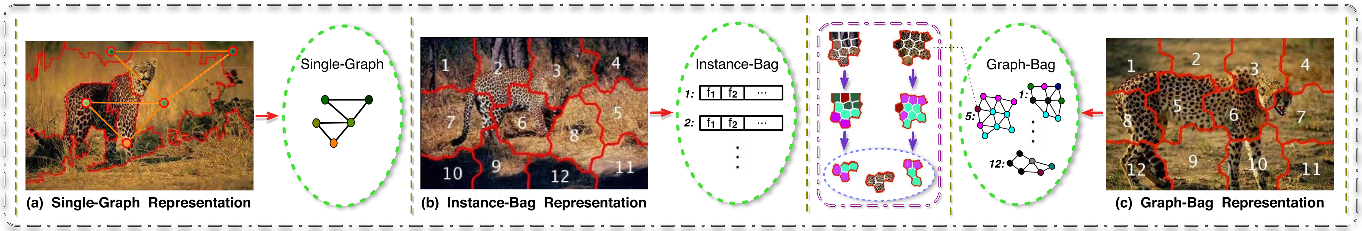


Fig. 1. Illustration of multigraph (i.e., graph-bag) representation derived from single-graph and multi-instance (i.e., instance-bag) representation. (a) Single-graph representation. A graph is used to denote an image with each node corresponding to a small region of the image and adjacent regions being connected by an edge [2], [5]. Single-graph representation can lose important local structure information, because image segmentation algorithms often separate a meaningful semantic object into multiple subregions (e.g., body or head of an animal). (b) Instance-bag representation. An image is represented as a bag of instances where each region inside the image corresponds to an instance represented in the vector space [7]. If a region contains an object of interest (e.g., a leopard), the image is labeled as positive. For traditional instance-bag representation, region #2 is represented as a single instance by using visual features. In other words, although region #2 contains multiple subregions (i.e., tree, grass, and leopard) with special structures and layout, existing instance-bag representation approaches discard the structure information and only consider the visual features of the whole region for learning. (c) Graph-bag representation. A more effective graph representation explicitly explores complex relationships among the data and uses effective data structures, such as graphs, to represent data for learning. As shown in the rectangle between (b) and (c), region #2 in (b) and region #5 in (c) share a common structure representing a meaningful object (e.g., the leopard). In this case, a region of a given image can be naturally represented as a graph in order to preserve and represent local structure information inside the region. This representation is more accurate than simply treating the whole region as one single instance, and it can be applied to other real-world applications (e.g., a biopharmaceutical activity test via a group/bag of molecules).

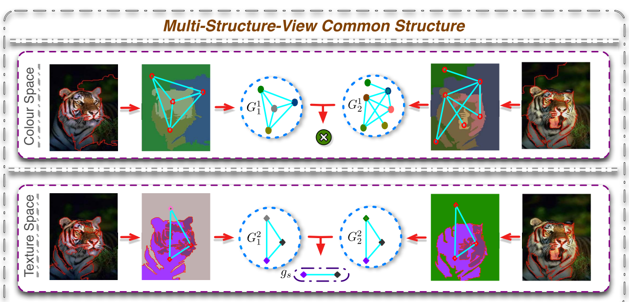


Fig. 2. MSV learning in which graphs are constructed from different structure views (e.g., the color view and the texture view). Existing graph classification research on images [2], [5] focuses on exploring common structures from single feature view graphs (such as the color view) as features for graph representation and learning. In some circumstances, no common structure exists in color space between two given graphs (e.g., G_1^1 and G_2^1), as shown in the first row. Instead, common structures may exist in other feature views (i.e., the texture view). For example, subgraph g_s is discovered from graphs G_1^2 and G_2^2 constructed from the texture view of the same objects.

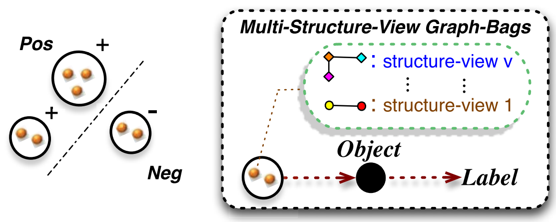


Fig. 3. Proposed MSVBL aims to separate objects into different classes (left) where the object is a bag of graphs constructed from multiple structure views (right).

and preserving the structure information of a portion of the object [9], [10]. If, for a region, the image contains any object-of-interest (e.g., a leopard/tiger), the bag will be labeled as positive. If no regions inside the image contain an object-of-interest, the bag will be labeled as negative. This bag constrained graph representation can also be applied to other practical application fields, such as drug activity prediction and scientific publication categorization. For the former, it is time-consuming and expensive to label each individual molecule (graph representation). In order to reduce prediction costs, the molecular group could be utilized to investigate the activities of a group (i.e., graph bag) of molecules. For the latter application, each scientific paper can be represented as a graph that considers the keyword correlations in the Abstract. Therefore, a scientific paper and all references cited in the paper form a graph bag.

The above-mentioned observations result in the novel bag constrained multiple structure-view learning paradigms described in Fig. 3, where the object is represented as a graph-bag consisting of graphs collected from multiple structure views. To build an effective learning model, the technical challenge is twofold: 1) multiple structure-view representations: how to find effective substructure features for different structure views and 2) graph-bag-based MSV learning: how to integrate bag constraints, where the class label is only available for a graph-bag, for further learning.

Intuitively, when objects are represented as a bag of MSV graphs, a straightforward solution to enable learning is to propagate the bag label to each graph inside the bag. In this case, the learning issue is downgraded to an up-to-date *multigraph-view graph classification* problem [11]. Unfortunately, due to the bag constraint that not all graphs inside a positive bag are positive, simple bag label propagation may cause some negative graphs to be mislabeled and deteriorate the learning accuracy. Alternatively, frequent subgraphs can first be explored to represent MSV graphs in vector space, so that the problem is downgraded to the latest *multiview multi-instance learning* [12]. However, this is still suboptimal, mainly, because simple frequent subgraph features do not have sufficient discriminative ability for learning, unless subgraph features are carefully explored and assessed across different structure views.

To solve the above-mentioned challenges, we propose an MSV bag constrained learning (MSVBL) algorithm, with emphasis on cross structure-view substructure feature explo-

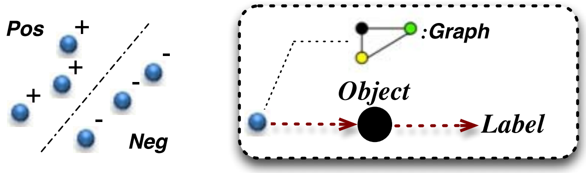


Fig. 4. Traditional graph classification intends to separate objects into different classes (left), where each object is represented as a single graph from a single structure-view (right).

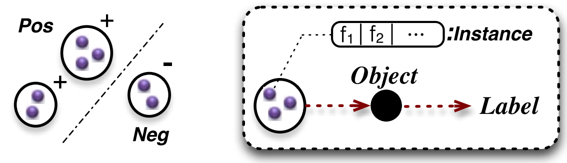


Fig. 5. Traditional multi-instance classification intends to separate a bag of instances into different classes (left), where the object for classification is a bag containing multiple instances with each instance being represented as a feature vector (right).

ration for accurate graph classification. A unique feature of MSVBL is that it progressively selects the most discriminative subgraph across different structure views under graph bag constraints, so it not only achieves maximum margins between labeled graph bags (positive versus negative), but also has minimum loss on the graphs in negative bags. The key contribution of this paper is threefold.

1) We formulate a new bag constrained graph classification problem, in which the learning object is a bag of graphs (i.e., graph-bag) with multiple structure views.
2) MSVBL integrates multiple structure-view substructure exploration and learning into a unified framework. This is inherently different from many common subgraph-based graph mining methods, which treat subgraph exploration and subsequent model learning as separate processes.
3) An upper bound score for each substructure is derived to effectively prune the substructure search space.

The rest of this paper is structured as follows. The related works are reviewed in Section II. Preliminaries and the problem statement are addressed in Section III. Section IV outlines the proposed MSV bag constrained graph learning framework MSVBL, and is followed by experiments in Section V. We conclude this paper in Section VI.

II. Related Works

Our problem is inspired by multi-instance learning on graphs with multiple structure views. Thus, in this section, we review works related to graph classification, multi-instance learning, and multiview bag/graph learning.

A. Graph Classification

Learning from graphs is a challenging task, mainly because graphs only have structured data (node and edge) but no feature representation, as shown in Fig. 4. Therefore, traditional feature-based approaches [13] (e.g., Bayesian networks, decision trees, and instance-based learning) cannot be directly applied for learning. Motivated by the similarity strategy in instance-based learning, a straightforward method is to directly calculate the graph similarity in the structure space. To this end, graph kernels [14], [15] have been proposed to make use of graph properties (e.g., node degree distribution [16]) to calculate the similarity between graphs. These methods share the same principle in their design: they enumerate graph structures, in terms of paths or walks, and so on, and compare the similarity between graphs using such structures. Because graph structures are potentially infinite, these methods often cannot identify which substructures (i.e., parts of the object graph) are mostly discriminative for distinguishing graphs from different class labels (i.e., enabling discriminative graph learning and classification).

Methods also exist to find good subgraphs that transfer the graph structure learning problem into a traditional supervised learning issue. In this case, majority learning approaches (e.g., support vector machines) can be directly used for classification. Nevertheless, if we enumerate all the subgraph candidates, the corresponding search space increases exponentially with respect to the number of graphs. To solve this issue, a commonly used subgraph estimation criterion (i.e., discovering all frequent subgraphs) is proposed by Yan and Han [17]. Other subgraph excavation methods (e.g., FFSM [18] and PSFS [19]) have also been proposed to find frequent subgraph features for further learning.

The above-mentioned frequency-based methods are mainly unsupervised, and do not utilize the label information. Supervised subgraph feature extraction methods have also been proposed to find discriminative subgraph features for different classes, such as LEAP [20], gPLS [21], COPK [22], and GAIA [23]. Kong and Yu [1] proposed a gSSC method to explore subgraphs (i.e., discriminative features) for semisupervised graph classification. Kong *et al.* recently proposed tackling graph learning issues (e.g., active graph classification [24], uncertain graph [25], and multilabel graph classification [26]) by employing the Hilbert–Schmidt independence criterion (HSIC) [27]. There are also a number of complex graph classification tasks, such as positive and unlabeled graph classification [28], graph stream classification [29], and multitask graph classification [30]. In addition, there is another stream of work, which explores the subgraph in multiplex networks [31], [32], which contain multiple types or edges. Although multiplex networks do not address the same multiple structure-view learning problems, they are potentially useful to solve similar problems, such as the image data set.

B. Multi-Instance Learning

Multi-instance learning was motivated by drug activity learning [33] where if a molecule group is active, at least one molecule is active. For inactive groups, all molecules inside the group are inactive. Such observations led to a novel multi-instance learning task, as shown in Fig. 5, in which the training data are instance-bags, with the label only available for each bag (but not for the instances inside the bag).

To support multi-instance learning, most existing methods attempt to upgrade the traditional supervised learning approaches. For example, Wang [34] proposed a lazy learning k-nearest neighbor algorithm, citation-KNN. Other approaches include tree-based multi-instance learning [35], multi-instance rule-based learning mi-DS algorithm [36], multi-instance kernel machines [7], and multi-instance-bag dissimilarity-based learning [37], [38]. Researchers have also attempted to adapt other popular single-instance learning algorithms to the multi-instance setting, such as multi-instance neural networks (e.g., BP-MIP [39] and RBF-MIP [40]) and MIBoost [41] (a variation of AdaBoost [42]).

The above-mentioned methods mainly focus on upgrading traditional supervised learning approaches for the multi-instance setting. On the other hand, transferring multi-instance issues to a classical single-instance setting can also work well. One simple and effective method is to transform the original multi-instance data into a single-instance data format by representing each bag as one instance, which is called SimpleMI [43]. Alternatively, [44] and [45] proposed an instance selection method using a feature mapping strategy based on the selected instances from training bags. Some algorithms are specially designed for multi-instance tasks, and examples include: maximum margin [46], scalable multi-instance learning based on the vector of locally aggregated descriptors, and MIL based on the Fisher vector [47].

C. Multiview Bag/Graph Learning

Multiple feature view learning [48], [49] has recently drawn much attention, and extensive research has shown that learning from multiple feature views is potentially more accurate than relying on a single feature view. Most of the existing feature-based learning approaches under multiple views are constructed on general studies, in which the label is allocated for a single instance with feature-vector representation. Nevertheless, feature-based learning approaches are unable to handle structure data and cannot be directly applied for the instance-bag learning tasks, where the learning object is the instance-bag and the label is only available for the instance-bag but not for the individual instance.

To explore informative features across multiple views in multi-instance learning, one intuitive solution is to first handle the single-view informative features by separating the views [50], and using concatenation methods [51] to combine all the selected features to represent bags for further classification. Nevertheless, this type of intuitive approach is unable to *globally* excavate the most informative features from different feature views to benefit the subsequent learning, mainly because they only locally explore and concatenates the features from each individual view. A contrasting approach is to concatenate all the feature views as one complete view, so that existing multiple instance feature learning approaches can be directly employed on the concatenated view (i.e., the whole feature space) for further learning [52]. One recent method uses a cotraining-based approach to deal with multi-instance data under different feature views [12].

The substructures features (i.e., subgraphs) mined from single structure-view graphs cannot adequately describe the learning object characteristics [53] in single structure view classification, whereas excavating rich information from different structure views benefits graph learning performance, mainly because an object may present various properties as for different feature spaces. A key problem for multiple structure-view feature-based learning is the view combination addressed in our previous multigraph-view learning for single graph classification [11]. One popular structure-view combination approach is to concatenate all individual structure views into a whole structure-view. The MSV learning task can then be transferred to a single structure-view learning problem. Nevertheless, such a structure-view combination can incur overfitting issues, especially when there are insufficient training graph data sets. Another cotraining structure-view method, which integrates all graph classifiers in each substructure-view to carry out the final target object classification, is also very common. In these structure-view combination approaches, the object for learning is the individual graph, so these approaches cannot be directly applied to a multigraph setting in which the object to be classified is a graph bag (i.e., a graph set). The classification object in existing multi-instance learning techniques is in the feature-vector space, so these methods cannot be used for graphs. This naturally raises the requirement to design new methods to handle bags that contain graphs under multiple structure views.

III. Definitions and Problem Statement

This section first introduces important notations and definitions, and then states our research problem.

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

Definition 2 (Subgraph/Substructure): Let $G = (\mathcal{V}, E, \mathcal{L}, l)$ and $g_i = (\mathcal{V}', E', \mathcal{L}', l')$ be two graphs. g_i is a subgraph/substructure of G, i.e., $g_i \subseteq G$, iff there exists an injective function $\varphi : \mathcal{V}' \to \mathcal{V}$ 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.

Definition 3 (Structure-View): A structure-view is denoted as a tuple $(\mathcal{V}, E, \mathcal{L}, l)$, which represents the structure of an object as a graph from a single structure-view, such as a single relationship or a single feature. Similarly, MSV denotes multiple types of tuples, which describe the structure variants of an object from different structure views.

Definition 4 (Multistructure-View Graph-Bag): An MSV graph-bag $B_i = \{B_i^1, \ldots, B_i^k, \ldots, B_i^v\}$ consists of many graph bags, where B_i^k denotes a single-structure-view graph bag from the kth structure-view, and each B_i^k contains many graphs $G_j^k \in B_i^k$ constructed from the kth structure-view. The class label of the graph bag B_i is represented by $Y_i \in \mathcal{Y}$, with $\mathcal{Y} = \{-1, +1\}$.

The set of all graph bags under all structure views is denoted by $\mathcal{B}$, with $\mathcal{B}^-$ and $\mathcal{B}^+$ denoting all negative and all positive

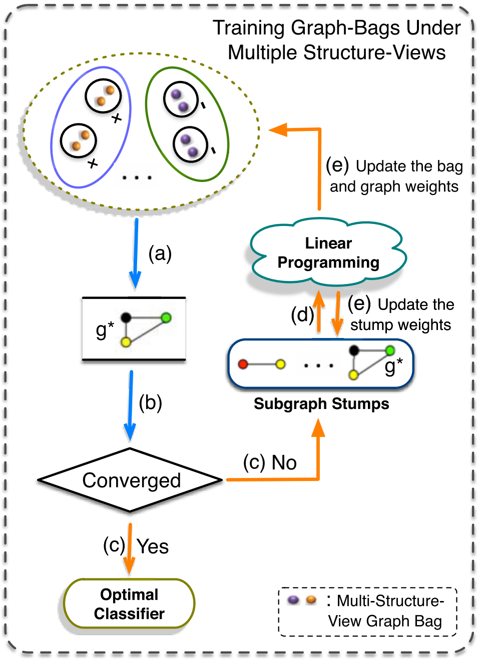


Fig. 6. Conceptual view of the proposed MSV learning for graph-bag classification (MSVBL). In each iteration, MSVBL selects an optimal subgraph g_* (step a). If the algorithm does not meet the stopping condition, g_* will be added to the subgraph set $\mathbf{g}$ (step d) or will otherwise terminate. During the loop, MSVBL solves a linear programming to update the weights for training graph-bags and graphs. The weights are continuously updated until the optimal classifier is obtained.

graph bags, respectively. The aggregation of all graphs in negative bags is denoted by $\mathcal{G}^-$. In addition, we use G_j to denote a graph generated from multiple structure views, with superscript k denoting the kth structure-view.

Definition 5 (Subgraph Representation for Graph): Given a subgraph set $\mathbf{g} = \{g_1, \ldots, g_m\}$ discovered from graphs under multiple structure views, where $g_s \in \mathbf{g}$ could be mined from any structure-view. Accordingly, each graph G_j can be represented as a subgraph feature vector $\mathbf{x}_j^G = [f_1^{G_j}, \ldots, f_m^{G_j}]^\top \in \{0,1\}^m$, where $f_s^{G_j} = 1, 1 \le s \le m$, iff g_s is a subgraph of G_j (i.e., $\exists G_j^k \in G_j \wedge g_s \subseteq G_j^k$) and $f_s^{G_j} = 0$ otherwise.

Definition 6 (Subgraph Representation for Graph-Bag): For subgraph set $\mathbf{g} = \{g_1, \ldots, g_m\}$ mentioned previously, an MSV bag B_i can be represented by a feature vector $\mathbf{x}_i^B = [f_1^{B_i}, \ldots, f_m^{B_i}]^\top \in \{0,1\}^m$, where $f_s^{B_i} = 1$, iff g_s is a subgraph of any graph G_j in bag B_i (i.e., $\exists G_j \in B_i^k \in B_i \wedge g_s \subseteq G_j$) and $f_s^{B_i} = 0$ otherwise.

Given a set of bags $\mathcal{B} = \{\mathcal{B}^1, \ldots, \mathcal{B}^k, \ldots \mathcal{B}^v\}$ containing labeled graph-bags from v structure views, the **aim** of MSV learning for bag constrained graph classification is to build a prediction model by exploring optimal subgraphs from the training graph bag set $\mathcal{B}$, and accurately predict the labels of previously unseen MSV graph bags.

IV. Multistructure-View Bag Learning

Our proposed MSV bag constrained graph classification framework is shown in Fig. 6. It consists of three major steps.

1) *Optimal Subgraph Exploration:* In each iteration, MSVBL explores a discriminative subgraph to improve the discriminative capability of the graph feature set $\mathbf{g}$.
2) *Bag Margin Maximization:* Based on the currently selected subgraphs $\mathbf{g}$, a linear programming is solved to achieve maximum bag margin for graph bag classification.
3) *Updating Bag and Graph Weights:* After the linear programming has been solved, the weight values for the training bags and graphs are updated until the algorithm converges.

A. Maximum Bag Margin Formulation

In graph-bag constrained learning, bag labels are asymmetric in the sense that every graph inside a negative graph-bag has a negative label, whereas at least one graph is positive in a positive graph-bag. Accordingly, we can aggregate the linear constraints from two levels (bag- and graph-levels) as

$$\begin{aligned}
\min_{\mathbf{w},\boldsymbol{\xi},\boldsymbol{\eta}} \; & \sum_k \sum_s^{m_k} w_s^k + C_1 \sum_{i:B_i \in \mathcal{B}} \xi_i + C_2 \sum_{j:G_j \in \mathcal{G}^-} \eta_j \\
\text{s.t.} \; & Y_i \sum_k \sum_{s=1}^{m_k} (w_s^B)^k h_{g_s}(B_i^k) \ge 1 - \xi_i, \quad i = 1, \ldots, |\mathcal{B}| \\
& \sum_k \sum_{s=1}^{m_k} (w_s^G)^k h_{g_s}(G_j^k) \le -1 + \eta_j, \quad j = 1, \ldots, |\mathcal{G}^-| \\
& \mathbf{w}^B \ge 0; \quad \mathbf{w}^G \ge 0; \quad \boldsymbol{\xi} \ge 0; \quad \boldsymbol{\eta} \ge 0
\end{aligned} \tag{1}$$

where $w_s^k = (w_s^B)^k + (w_s^G)^k$, ξ_i and η_j are the evaluation of the misclassification. C_1 and C_2 are misclassification tradeoff hyperplane margin and errors, which are both set to 1 in our experiment. Because bag labels are known, the weighted errors are $C_1 \sum_{i:B_i \in \mathcal{B}} \xi_i$. In addition, graphs in the negative bags are known as negative. Therefore, the weighted errors at the graph level are $C_2 \sum_{j:G_j \in \mathcal{B}^-} \eta_j$.

In (1), $h_{g_s}(B_i^k)$ is a weak subgraph classifier, which outputs the class label of the bag B_i^k in the kth view based on subgraph g_s, and $h_{g_s}(G_j^k)$ is a weak subgraph classifier for the graph G_j^k in the kth structure-view based on subgraph g_s. We can use a subgraph g_s as a decision stump classifier for a graph or bag in the kth structure-view as

$$\begin{cases} h_{g_s}(B_i^k) = (\psi_s^B)^k \left(2I(g_s \subseteq B_i^k) - 1\right) \\ h_{g_s}(G_j^k) = (\psi_s^G)^k \left(2I(g_s \subseteq G_j^k) - 1\right) \end{cases} \tag{2}$$

where $g_s \subseteq B_i^k$ iff g_s is a subgraph of any graph G in bag B_i^k, i.e., $\exists G \in B_i^k \wedge g_s \subseteq G$. $(\psi_s^B)^k$ and $(\psi_s^G)^k$ ($\psi_s^B, \psi_s^G \in \Psi = \{-1, +1\}$) are parameters controlling the label of the classifiers, with $I(\cdot)$ being an indicator function. $(w_s^B)^k$ and $(w_s^G)^k$ denote the weights of the bag and graph in the kth structure-view, respectively. For a subgraph set with size $m = \sum_k m_k$, the prediction rule for a graph bag B_i is a linear structure-view combination of the corresponding weak classifiers as

$$\mathcal{H}(B_i) = \operatorname{sign}\left(\sum_k \sum_{s=1}^{m_k} (w_s^B)^k h_{g_s}(B_i^k)\right). \tag{3}$$

B. Linear Programming Optimization

To support multiple structure-view bag constrained graph classifications, a set of subgraph features $\mathbf{g} = \{g_1, \ldots, g_s, \ldots, g_m\}$ is required. One straightforward solution is an exhaustive enumeration strategy, which enumerates all subgraphs to find the best ones for learning. Nevertheless, the number of subgraph candidates increases exponentially, and the huge amount of time consumed makes this type of greedy subgraph search method impractical for real-world learning tasks. This problem can be solved by a column generation technique [54], which works on the Lagrangian dual problem with respect to (1). Starting from an empty subgraph feature set $\mathbf{g}$, column generation iteratively adds one subgraph g_s to $\mathbf{g}$ which violates the constraint under the dual learning problem. Each time the subgraph set $\mathbf{g}$ is updated, column generation resolves the primal problem in (1) by solving the restricted dual problem. This process keeps running until convergence, which can be formulated as

$$\begin{aligned}
&\max_{\boldsymbol{\gamma},\boldsymbol{\mu}} \sum_{i:B_i\in\mathcal{B}} \gamma_i - \sum_{j:G_j\in\mathcal{G}^-} \mu_j \\
&\text{s.t. } 0 \le \gamma_i \le C_1, \quad i = 1, \ldots, |\mathcal{B}| \\
&\qquad 0 \le \mu_j \le C_2, \quad j = 1, \ldots, |\mathcal{G}^-| \\
&\qquad \sum_k \left(\sum_{i:B_i\in\mathcal{B}} \gamma_i Y_i h_{g_s}\big(B_i^k\big) - \sum_{j:G_j\in\mathcal{G}^-} \mu_j h_{g_s}\big(G_j^k\big) \right) \le 2\upsilon
\end{aligned} \tag{4}$$

where γ_i and μ_j are Lagrange multipliers, with $\sum_k 1 = \upsilon$. Note that the related dual problem has a small number of variables, but many constraints. Among them, each constraint $\zeta_{g_s} = \sum_k \ (\sum_{i:B_i\in\mathcal{B}} \gamma_i Y_i h_{g_s}(B_i^k) - \sum_{j:G_j\in\mathcal{G}^-} \mu_j h_{g_s}] (G_j^k)) \le 2\upsilon$ indicates a subgraph feature g_s over all graph-bags $\mathcal{B}$, with the first and second terms of the left of constraint being the gain on the labeled graph-bags and graphs in negative bags, respectively. Intuitively, this constraint provides a metric to access the bag constraint-based discriminative power of a given subgraph g_s.

C. Bag Constrained Criteria

In addition to favoring the subgraph in the feature set $\mathbf{g}$ which has a high discriminative score, we also want to make sure that the selected subgraph g_s has the capability to identify positive graphs in positive bags. The selected subgraph set $\mathbf{g} = \{g_1, \ldots, g_m\} \ni g_s$ should ensure the following constraints.

1) *Graph-Bag Must-Link:* Because bag labels are known in advance, the selected subgraph features for graph-bags B_i and B_j should ensure that graph-bags with the same label are close to one another.
2) *Graph-Bag Cannot-Link:* The selected subgraphs should ensure the disparity of graph bags with different class labels by taking into account the data distributions inside each graph-bag.
3) *Graph Must-Link:* In our graph-bag setting, every graph inside the negative bags is negative, and thus, the subgraph feature representation should encourage negative graphs to be close to one another.
4) *Graph Separability:* The corresponding genuine labels for graphs in positive graph bags are unavailable. To this end, we adopt the assumption of principal component analysis, i.e., exploring the component with the largest possible variance, to preserve the diversity in positive bags.

Based on the above-mentioned discussion, the subgraph feature estimation $\ell(\mathbf{g})$ can be formulated as follows:

$$\begin{aligned}
\ell_{\mathbf{g}} = \ell_{\mathbf{g}}^B + \ell_{\mathbf{g}}^G &= \frac{1}{2}\sum\nolimits_{Y_i,Y_j} K_{\mathbf{g}}^B(B_i, B_j) Q_{i,j}^B \\
&\quad + \frac{1}{2}\sum\nolimits_{G_i,G_j} K_{\mathbf{g}}^G(G_i, G_j) Q_{i,j}^G
\end{aligned} \tag{5}$$

where $\ell_{\mathbf{g}}^B$ denotes the similarity between two graph-bags via bag level criteria 1) and 2), with $\ell_{\mathbf{g}}^G$ representing the graph level criteria 3) and 4). $Q_{ij}^B = \{-1/|A|, Y_iY_j = 1; 1/|B|, Y_iY_j = -1\}$, with $A = \sum_{Y_iY_j=-1} 1$, and $B = \sum_{Y_iY_j=1} 1$ representing the total bag pairwise constraints. $Q_{ij}^G = \{-1/|C|, \forall G_i, G_j \in \mathcal{B}^-;\ 1/|D|, \forall G_i, G_j \in \mathcal{B}^+\}$, with $C = \sum_{G_i,G_j\in\mathcal{B}^-} 1$ and $D = \sum_{G_i,G_j\in\mathcal{B}^+} 1$ denote graph pairwise constraints. $K_{\mathbf{g}}^B(B_i, B_j)$ and $K_{\mathbf{g}}^G(G_i, G_j)$ denote the distance between two bags or graphs in the feature vector space under the explored subgraph set $\mathbf{g}$ using an L_2 norm measure.

Accordingly, for bag level $\ell_{\mathbf{g}}^B$, we have

$$\begin{aligned}
\ell_{\mathbf{g}}^B &= \frac{1}{2}\sum\nolimits_{Y_i,Y_j} \left\| \mathbf{x}_i^B - \mathbf{x}_j^B \right\|^2 Q_{i,j}^B \\
&= \sum\nolimits_{Y_i,Y_j} \left(\mathbf{x}_i^B\right)^\top \mathbf{x}_i^B Q_{i,j}^B - \sum\nolimits_{Y_i,Y_j} \left(\mathbf{x}_i^B\right)^\top \mathbf{x}_j^B Q_{i,j}^B \\
&= \sum\nolimits_{Y_i} \left(\mathbf{x}_i^B\right)^\top \mathbf{x}_i^B \sum\nolimits_{Y_j} Q_{i,j}^B - \sum\nolimits_{Y_i,Y_j} \left(\mathbf{x}_i^B\right)^\top \mathbf{x}_j^B Q_{i,j}^B \\
&= \sum\nolimits_{Y_i} \left(\mathbf{x}_i^B\right)^\top \mathbf{x}_i^B D_{i,i}^B - \sum\nolimits_{Y_i,Y_j} \left(\mathbf{x}_i^B\right)^\top \mathbf{x}_j^B Q_{i,j}^B \\
&= tr\left(\mathcal{X}_B D_B \mathcal{X}_B^\top\right) - tr\left(\mathcal{X}_B Q_B \mathcal{X}_B^\top\right) \\
&= tr\left(\mathcal{X}_B (D_B - Q_B) \mathcal{X}_B^\top\right) = tr\left(\mathcal{X}_B L_B \mathcal{X}_B^\top\right) \\
&= \sum\nolimits_{g_s\in\mathbf{g}} \left(\boldsymbol{f}_s^B\right)^\top L_B \boldsymbol{f}_s^B
\end{aligned} \tag{6}$$

where $tr(\cdot)$ denotes the matrix trace operator, $\mathcal{X}_B = [\mathbf{x}_1^B, \ldots, \mathbf{x}_p^B] = [\boldsymbol{f}_1^B, \ldots, \boldsymbol{f}_m^B]^\top \in \{0, 1\}^{m\times p}$, with p denoting the size of bags. $\boldsymbol{f}_s^B$ $(1 \le s \le m,\ g_s \in \mathbf{g})$ is regarded as a vector indicator of subgraph g_s, with respect to all graph bags, i.e., $\boldsymbol{f}_s^B = [f_s^{B_1}, \ldots, f_s^{B_p}]^\top$, where $f_s^{B_i} = 1, 1 \le i \le p$ iff $\exists G \in B_i^k \in B_i \wedge g_s \subseteq G$ and $f_s^{B_i} = 0$ otherwise. D_B, as a diagonal matrix, is generated from Q_B, where $D_{i,i}^B = \sum_j Q_{ij}^B$. L_B is a Laplacian matrix, denoted by $L_B = [L_{i,j}^B]^{p\times p} = D_B - Q_B$. Similarly, the graph level $\ell_{\mathbf{g}}^G$ in (5) can also be derived as a matrix format, which joins with graph level $\ell_{\mathbf{g}}^B$ to rewrite (5) as

$$\ell_{\mathbf{g}} = \sum_{g_s\in\mathbf{g}} \left(\left(\boldsymbol{f}_s^B\right)^\top L_B \boldsymbol{f}_s^B + \left(\boldsymbol{f}_s^G\right)^\top L_G \boldsymbol{f}_s^G\right) = \sum_{g_s\in\mathbf{g}} \boldsymbol{f}_s^\top L \boldsymbol{f}_s \tag{7}$$

where

$$\boldsymbol{f}_s = \begin{bmatrix} \boldsymbol{f}_s^B \\ \boldsymbol{f}_s^G \end{bmatrix}, \quad L = \begin{bmatrix} L_B & 0 \\ 0 & L_G \end{bmatrix} \tag{8}$$

where f_s is a vector indicator of subgraph g_s with respect to the data combined with bags and graphs. In this case, each subgraph g_s will have an independent discrimination criterion $\ell_{g_s} = \boldsymbol{f}_s^\top L \boldsymbol{f}_s$, because $\ell_{\mathbf{g}} = \sum_{g_s \in \mathbf{g}} \ell_{g_s}$.

Definition 7 (mgScore): Given a graph-bag set $\mathcal{B}$ containing multiple structure-view graphs, the informative score for a subgraph g_s can be measured by

$$£_{g_s} = \sum_k \left(\sum_{B_i \in \mathcal{B}} \gamma_i Y_i h_{g_s}\big(B_i^k\big) - \sum_{G_j \in \mathcal{G}^-} \mu_j h_{g_s}\big(G_j^k\big) \right) + \boldsymbol{f}_s^\top L \boldsymbol{f}_s. \tag{9}$$

To construct the MSV bag constraining model, the most informative subgraph feature considering each training bag weight and graph weight in negative bags across all structure views needs to be explored for bag constrained graph classification.

D. Optimal Subgraph Exploration

To discover subgraphs for validation, an intuitive solution for exploring an informative subgraph set is to employ an exhaustive enumeration strategy, which needs to enumerate all subgraphs and uses their mgScore values for ranking. Nevertheless, the number of subgraph candidates increases exponentially with respect to the size of the search space (i.e., the graph set collected from each structure-view). The huge time consumption makes this type of greedy subgraph search method infeasible for real-world learning tasks. Instead, we apply gSpan [17], which is an efficient subgraph mining approach based on the depth-first search (DFS) strategy, to find the subgraph feature candidates. The core concept of gSpan is that it establishes a lexicographic order to encode each graph, through which all frequent subgraphs are discovered efficiently. In MSV scenarios, we derive an upper bound for mgScore to prune the DFS-code tree (i.e., reduce the search space) as follows:

Theorem 1 (mgScore Upper Bound): Given two subgraphs $g_s, g_s' \in \mathbf{g}$, where g_s' is a supergraph of g_s (i.e., g_s is a subgraph of g_s' with $g_s' \supseteq g_s$). The mgScore of g_s', $£_{g_s'}$ is bounded by $\hat{£}_{g_s}$, i.e., $£_{g_s'} \le \hat{£}_{g_s}$, with $\hat{£}_{g_s}$ being defined as

$$\hat{£}_{g_s} = \max\big(\zeta_{g_s}^-, \zeta_{g_s}^+\big) + \boldsymbol{f}_s^\top \hat{L} \boldsymbol{f}_s \tag{10}$$

where, $\hat{L}$ is conducted by $\hat{L}_{i,j} = \max(0, L_{i,j})$, and

$$\zeta_{g_s}^- = 2\sum_k \left(\sum_{i:Y_i=-1, g_s \in B_i^k} \gamma_i + \sum_{j:g_s \in G_j^k} \mu_j \right) + \upsilon \sum_{i:B_i \in \mathcal{B}} \gamma_i Y_i \tag{11}$$

$$\zeta_{g_s}^+ = 2\sum_k \sum_{i:Y_i=+1, g_s \in B_i^k} \gamma_i - \upsilon \left(\sum_{i:B_i \in \mathcal{B}} \gamma_i Y_i - \sum_{j:G_j \in \mathcal{G}^-} \mu_j \right). \tag{12}$$

For any subgraph $g_s' \supseteq g_s$, $£_{g_s'} \le \hat{£}_{g_s}$ (i.e., the mgScore of subgraph g_s', $£_{g_s'}$ is bounded by $\hat{£}_{g_s}$). The proof is detailed in the following three components: 1) $\zeta_{g_s} \le \zeta_{g_s}^-$ in Appendix A; 2) $\zeta_{g_s} \le \zeta_{g_s}^+$ in Appendix B; and 3) $\ell_{g_s'} \le \boldsymbol{f}_s^\top \hat{L} \boldsymbol{f}_s$ in Appendix C. In this case, the $\max(£_{g_s}^-, £_{g_s}^+) + \boldsymbol{f}_s^\top \hat{L} \boldsymbol{f}_s$ will be selected as the upper bound. When a subgraph g_s is generated, all its supergraphs are upper bounded by $\hat{£}_{g_s}$. Therefore, this theorem will help to reduce the search space efficiently.

Algorithm 1 Informative Subgraph Exploration

Input:
$\mathcal{B} = \{\mathcal{B}^1, \ldots, \mathcal{B}^k, \ldots, \mathcal{B}^\upsilon\}$: A multi-structure-view bag set with υ structure-views;
$\boldsymbol{\gamma} = \{\gamma_1, \ldots, \gamma_{|\mathcal{B}|}\}$: A bag weight set;
$\boldsymbol{\mu} = \{\mu_1, \ldots, \mu_{|\mathcal{G}^-|}\}$: A negative graph weight set;
min_sup: The threshold of the frequent subgraph;
Output:
g_*: The most discriminative subgraph;
1: $g_* = \emptyset$;
2: $\mathcal{G} = \{\mathcal{G}^1, \ldots, \mathcal{G}^k, \ldots, \mathcal{G}^\upsilon\} \leftarrow$ Aggregate all graphs in $\mathcal{B}$;
3: **for all** structure-views $\mathcal{G}^k, k = 1, \ldots, \upsilon$ in $\mathcal{G}$ **do**
4: **while** Recursively visit the DFS Code Tree in gSpan **do**
5: $g_s^k \leftarrow$ current visited subgraph in DFS Code Tree;
6: **if** $freq(g_s^k) < min\_sup$, **then**
7: **return**;
8: Compute the mgScore $£_{g_s^k}$ for subgraph g_s^k using Eq. (10);
9: **if** $£_{g_s^k} \ge £_{g_*}$ **or** $g_* == \emptyset$, **then**
10: $g_* \leftarrow g_s^k$;
11: **if** $\hat{£}_{g_s^k} \ge £_{g_*}$, **then**
12: Depth-first search the subtree rooted from node g_s^k;
13: **end while**
14: **end for**
15: **return** g_*;

The above-mentioned upper bound can be used to prune the DFS code search tree in gSpan via the branch-and-bound pruning strategy; the complete subgraph feature exploration approach is listed in Algorithm 1. The algorithm enumerates subgraph features by searching the whole DFS code tree for each structure-view. If a current subgraph g_s^k in the kth view is infrequent, both g_s^k and its related subtree need to be discarded (lines 6 and 7). If not, the mgScore of g_s^k (i.e., $£_{g_s^k}$) will be calculated (line 8). If $£_{g_s^k}$ is greater than the current optimal mgScore $£_{g_*}$ or the optimal subgraph $£_{g_*}$ is empty (i.e., in the first iteration), $£_{g_s^k}$ will be regarded as the current optimal item $£_{g_*}$ (lines 9 and 10). Subsequently, the upper bound pruning module will check whether $\hat{£}_{g_s^k}$ is less than $£_{g_*}$; if so, this means that the mgScore value of any supergraph $g_s^{k'}$ of g_s^k (i.e., $g_s^{k'} \supseteq g_s^k$) will not be greater than $£_{g_*}$. Thus, the subtree rooted from g_s^k is safely pruned. If $\hat{£}_{g_s^k}$ is indeed greater than the mgScore of g_*, the search process will sequentially visit nodes from the subtree of g_s^k (lines 11 and 12).

E. MSVBL

The complete procedures of the proposed MSVBL framework MSVBL are listed in Algorithm 2, which iteratively extracts informative subgraphs across different structure views

Algorithm 2 MSVBL

Input:

$\mathcal{B} = \{\mathcal{B}^1, \ldots, \mathcal{B}^k, \ldots, \mathcal{B}^v\}$: A multi-structure-view graph bag set;

min_sup: The threshold of the frequent subgraph;

m: the maximum number of iteration;

Output:

The target label Y_c of a test multi-structure-view bag B_c;

// Training Phase:

1: $\mathbf{g} \leftarrow \emptyset$;
2: $t \leftarrow 0$;
3: **while** $t \le m$ **do**
4: $g_* \leftarrow$ Apply $\mathcal{B}$ and min_sup to obtain the most informative subgraph; // Alogirthm 1
5: **if** $\zeta_{g_*}/2v \le 1 + \epsilon$ **then**
6: **break**;
7: $\mathbf{g} \leftarrow \mathbf{g} \cup g_*$;
8: Solve Eq. (1) based on $\mathbf{g}$ to get $\boldsymbol{w}^B$ and $\boldsymbol{w}^G$, and the Lagrange multipliers of Eq. (4) $\boldsymbol{\gamma}$ and $\boldsymbol{\mu}$;
9: $t \leftarrow t + 1$;
10: **end while**

// Testing Phase:

11: $Y_c \leftarrow sign\Big(\sum_k \sum_{g_s \in \mathbf{g}} \big(w_s^B\big)^k h_{g_s}\big(B_c^k\big)\Big)$.
12: **return** Y_c.

to expand the candidate subgraph set $\mathbf{g}$, by using mgScore. After m iterations, MSVBL will boost the generated m weak classifiers for final prediction.

MSVBL starts from an empty subgraph set $\mathbf{g} = \emptyset$ (line 1), and iteratively chooses the most informative subgraph feature g_* in each round (line 4) according to Algorithm 1. If the current optimal subgraph no longer violates the constraint, the iteration process terminates (lines 5 and 6). Because the difference between the optimal values in the last few iterations is relatively small, a threshold ϵ is used to relax the stopping condition (i.e., we set $\epsilon = 0.05$ in our experiments). After that MSVBL solves the linear programming problem by using the current optimal subgraph set $\mathbf{g}$ to recalculate two groups of weight values: 1) $\boldsymbol{w}^B$ and $\boldsymbol{w}^G$: the weights for bag-level and graph-level weak subgraph decision stumps, respectively and 2) $\boldsymbol{\gamma}$ and $\boldsymbol{\mu}$: the weights of training bags and graphs in negative bags for optimal subgraph feature exploration in the next iteration, which can be calculated from the Lagrange multipliers in the primal issue (line 8). If the learning framework converges or the maximum number of iterations is achieved, the training phase of MSVBL is terminated. During the testing phase, the label Y_c of a test bag B_c is determined by the final classifier $\text{sign}(\sum_k \sum_{g_s \in \mathbf{g}} (w_s^B)^k h_{g_s}(B_c^k))$.

V. Experiments

A. Benchmark Graph Bag Data Sets

1) Scientific Publication Multistructure-View Graph Bags: The information from the Abstract content and the paper citation relationship naturally form two structure views. Each scientific paper is converted into an Abstract content view graph by utilizing the contextual correlations (edges in graphs) of keywords (nodes in graphs) in the Abstract. Using linked keyword relationships (e.g., cooccurrence of keywords in different sentences) to form a graph representation for each paper (as shown in Fig. 7 to be explained later) has shown better performance than simple bag-of-words representation [6], because one or multiple independent keywords/attributes is insufficient to describe the content of a paper. For a paper citation relationship view graph, each graph node represents a paper ID with edges representing the citation relationships among papers (detailed in [29]). With graphs built from the paper and the references cited in the paper, a paper can be represented as a graph bag containing multiple graphs in two structure views (i.e., Abstract view versus citation relationship view). For example, assume paper A cites papers A_1, A_2, and A_3, and the label of A is "Positive." For each view, we will first generate one graph from A, A_1, A_2, and A_3, respectively. After that we put all four graphs in one bag, and label the bag as "Positive." Thus, each paper corresponds to a *graph bag* with two structure views (Abstract content view versus paper citation relationship view).

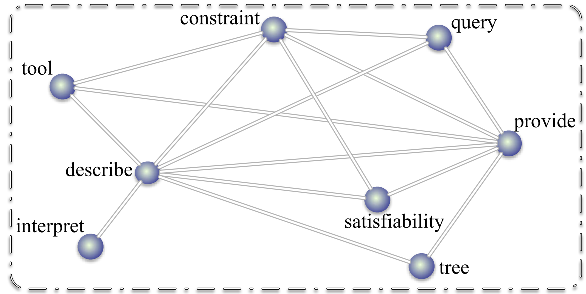


Fig. 7. Graph representation of the Abstract in a paper entitled "Static Analysis in Datalog Extensions." Each node (i.e., a circle) denotes a keyword in the Abstract. The weight values between nodes indicate the correlations between keywords. By using a threshold (e.g., 0.005), an Abstract can be converted into an unweighted graph.

The Digital Bibliography and Library Project (DBLP) data set[1] consists of bibliography in computer science, with each record containing information, such as Abstract, authors, year, venue, title, and references. We select papers published in Artificial Intelligence (AI: IJCAI, AAAI, NIPS, UAI, COLT, ACL, KR, ICML, ECML, and IJCNN) as positive bags, and Database (DB: SIGMOD, PODS, VLDB, ICDE, CIKM, DASFAA, ICDT, and SSD) as negative bags to form an MSV learning task. The objective is to predict whether a scientific publication is part of the artificial intelligence (positive) or database (negative) field by using the graph representations with the above structure views. The two research fields overlap in many aspects, e.g., data mining, information retrieval, and pattern recognition, which help create a challenging MSV learning task.

In the Abstract structure-view, an element fuzzy cognitive map (E-FCM) [55] is utilized for each abstract to explore keywords as nodes, and correlations between keywords are used to form the edges of each graph, as shown in Fig. 7.

[1] http://dblp.uni-trier.de/xml/

The same graph representation for scientific publication can be found in our previous work [51]. In the experiments, we choose 600 papers in total (corresponding to 600 multiple structure-view bags) to form positive (AI) bags (300 bags with 1756 graphs) and negative (DB) bags (300 bags with 1738 graphs).

2) Content-Based Image Multistructure-View Graph Bags: The original images [56] collected from the "Corel" data set[2] are preprocessed using VLFeat segmentation.[3] Each image is segmented into multiple regions, with each region corresponding to one graph. For an individual region simple linear iterative clustering [57], a state-of-the-art superpixel-based method is applied to obtain graph representation. Each node indicates one superpixel and each edge denotes the adjacency relationship between two superpixels.

Two types of feature [58], hue–saturation–value (HSV) in the color space and local binary patterns (LBPs) in the texture space, are naturally related to two structure views. HSV is a common cylindrical-coordinate representation applied for constructing a color model, and LBP is a well-known texture spectrum descriptor for capturing local texture features. We first extract a three-channel HSV feature on each pixel for the HSV representation. A 256-D codebook is constructed via k-means clustering on the explored HSV cylindrical-coordinate representations. Each pixel is transferred to a 1-D code by calculating the distance between the pixel color and the prior cluster centers. We then assign a 256-D histogram-based vector to each superpixel (i.e., HSV-based superpixel representation) using the code occurrence statistics. The uniform LBP is used to generate a 59-bin code on each pixel, which is assigned to 1 bin based on the local texture pattern. A 59-D histogram representation can be constructed to encode the statistics of each superpixel. Similar graph representation can be found in our previous work [59]. In this image related experimental data set, the superclass "Cats" has three subclasses "Tiger," "Lion," and "Leopard," which are used as positive images (300 bags with 2679 graphs). In addition, 300 images of other animals are randomly selected as negative bags, including 2668 segments (i.e., graphs) in negative bags.

B. Experimental Settings

All experimental results and comparisons are reported on 10 times tenfold cross-validation. Unless specified otherwise, we set the minimum support threshold min_sup $= 3\%$ for scientific publication data (Section V-A1) and min_sup $= 2\%$ for content-based image data (Section V-A2). All experiments are conducted on a Linux cluster 16 processors [Interl(R) Xeon(R) at 3.47-GHz CPU] and 128-GB memory size.

C. Baseline Methods

To the best of our knowledge, this is the first work to consider the multiple structure-view bag constrained graph classification problems. The contribution of this paper is to design an effective graph classification framework under multiple structure views to advance the fundamental graph classification technique, not a new algorithm in a special domain (e.g., image or text, or other domains in which the proposed framework can be applied) to compare with other type of technique, e.g., deep learning and extreme learning machines. As a result, all baseline methods belong to the graph classification family.

To comparatively study the performance of the proposed MSVBL method, we first use two types of baseline (bag level and graph level) for single structure-view evaluation, and then implement three different structure-view combination strategies for comparison studies. Bag-level approaches first discover informative subgraphs at bag level to represent graphs in the bag set (i.e., transferring a graph-bag set to an instance-bag set) for classification. By contrast, graph-level approaches propagate graph bag labels to all graphs in the bag, through which the informative subgraphs can be explored to represent bag-of-graphs to bag-of-instances in the feature vector space.

1) Subgraph Evaluation Criterion: To explore informative subgraphs for comparison purposes, we implement the following four different types of subgraph feature evaluation criteria.

a) Frequency-based approach: For the purpose of selecting subgraph features from graphs, the Top-k [60] approach adopts the frequency criteria to select the highest frequent subgraphs as features. In the graph-bag setting, the bag-level frequency is measured with respect to bags (i.e., the occurrence of the subgraph is counted as 1 if a subgraph is contained in one or more graphs inside a bag, or 0 otherwise). By contrast, the graph-level frequency setting directly calculates the frequency with respect to graphs.

b) Information theory-based approach: Information gain (IG), which is used in selecting feature nodes for decision tree construction, is commonly used for subgraph estimation in graph classification [24], [29]. When dealing with graph bags, bag-level IG tries to select subgraphs with the highest IG based on subgraph feature representation for graph bags, as given in Definition 6. Graph-level IG calculates the IG score on graphs based on Definition 5.

c) Discrimination-based approach: A novel discriminative subgraph selection criterion, gSSC [1], has demonstrated strong performance in tackling graph structure data. The basic idea is to select informative subgraphs such that graphs with different labels in the subgraph feature space are distinct from each other. Accordingly, the bag- and graph-level gSSC apply the gSSC discriminative measures to bags (graph-bags with bag labels) and graphs (graph objects and the labels via inheriting the bag labels), respectively.

d) Dependence-based approach: The HSIC, which measures the dependence between two variables in a specially designed kernel space, has recently been proposed to maximize the dependence between subgraphs for graph objects. This state-of-the-art subgraph dependence evaluation criterion has been successfully employed in many graph learning tasks, such as traditional graph classification [24], uncertain graphs [25], and multilabel graphs [26]. The bag-level gHSIC adopts the

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

[3]http://www.vlfeat.org/

HSIC criterion to explore subgraphs using the proposed bag representation for learning, and graph-level gHSIC simply works on graphs by propagating the bag label to graphs inside each bag.

2) Multistructure-View Combination: For comparison purposes, the following three structure-view combination strategies across different structure views are also implemented for learning.

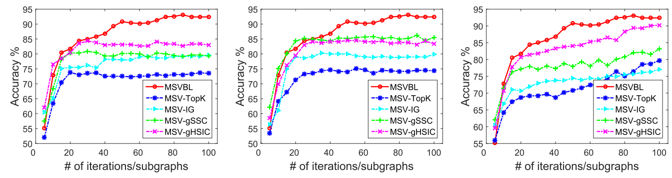
a) Local MSV: Similar to the view combination in [51], the local structure-view combination strategy adopts a concatenation mechanism to obtain MSV subgraphs from different structure views. The above-mentioned subgraph evaluation criterion (e.g., gSSC or gHSIC) is used for each single structure view to select m_k subgraph features, which will be concatenated as final subgraphs to represent graphs as feature vectors. A multi-instance learner (e.g., MIBoost [41]) will then be used for classification.

b) Global MSV: The global view combination strategy concatenates heterogeneous feature spaces into one homogeneous feature space. Single-view feature selection methods are applied to the concatenated features for learning [52]. Because there is no feature space in the graph domain, this baseline approach first concatenates all the frequent subgraph features discovered from all structure views (i.e., constructing the entire subgraph feature space), and then utilizes the Top-k, IG, gSSC, or gHSIC evaluation criteria to directly explore the m subgraphs from all structure views for graph classification.

c) Ensemble MSV: We also compare our proposed method MSVBL with a state-of-the-art *multi-instance-view combination strategy* [12]. A number of informative subgraphs are excavated for each single structure view via Top-k, IG, gSSC, or gHSIC evaluation criteria. By representing each graph as an instance in the feature vector space, this structure-view combination baseline trains a multi-instance classifier (e.g., MIBoost [41]) by treating each view independently and integrates classifiers across all structure views for prediction.

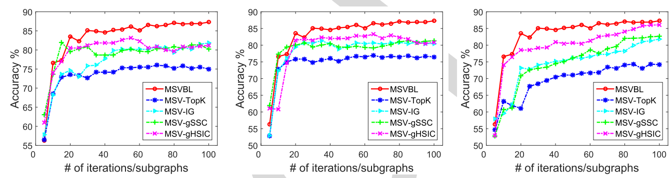
To sum up, we first carry out comparisons in our experiment via the above-mentioned three structure-view combination strategies based on the graph- or bag-level subgraph evaluation criterion.

3) Latest Graph Classification Advances: By directly propagating bag labels to graphs inside each bag, the problem in this paper can be transferred to the state-of-the-art graph learning task with multiple structure views (MSVGL [11]), which will also be used as a type of baseline (detailed in Section V-D3). We also implement a bMSVBL approach (i.e., MSVBL without using the graph level constraint) as a baseline to explore the efficiency of the unified two level (bag- and graph-level) framework. A baseline dMSVBL approach [53], which does not consider the bag constrained criteria, is also implemented to demonstrate the distinct performance of the proposed MSVBL (detailed in Section V-D4). An unbounded MSVBL (uMSVBL) approach with no pruning module as described in Section V-D is implemented to evaluate the efficiency of the pruning strategy used in MSVBL (detailed in Section V-D7).



(a) (b) (c)

Fig. 8. Bag-level comparisons on *DBLP graph bag data set* with different structure-view combination approaches. (a) Local MSV. (b) Global MSV. (c) Ensemble MSV.



(a) (b) (c)

Fig. 9. Bag-level comparisons on *image graph bag data set* with different structure-view combination approaches. (a) Local MSV. (b) Global MSV. (c) Ensemble MSV.

D. Experimental Results

1) Comparison With Bag-Level Evaluation Criteria: Figs. 8 and 9 report the results of the diverse bag-level subgraph feature estimation criteria (i.e., TopK, IG, gSSC, or gHSIC) under the proposed three multiple structure-view combination strategies on DBLP and Image bag constrained graph data sets, respectively. It can be seen that MSVBL consistently performs better than baseline approaches when the number of selected subgraph features is 20 or more. When the number of selected subgraph features is less than 10, the performance of all algorithms is comparable, mainly because a small number of subgraph stumps (i.e., weak classifiers) leads to inferior classification accuracy in early iterations.

Although the generally worst-performing MSV-TopK obtains slightly better performance when the number of subgraph candidates is sufficiently large (e.g., $\geq$80) under the ensemble structure-view combination strategy, as shown in Fig. 8(c), its subgraph evaluation measure relies on frequency and is not suitable for graph-bag learning with multiple structure views. This is mainly because their frequent subgraphs are not selected toward the distinction of complicated objects in positive and negative graph bags.

Most of the time, the information theory-based MSV-IG and discrimination-based MSV-gSSC subgraph evaluations are comparable, as shown in Figs. 8(a) and 9(a)–(c). However, gSSC-based approach significantly outperforms IG-based MSVBL on the DBLP graph data, as shown in Fig. 8(b) and (c), which can be attributed to the discriminative criterion used in MSV-gSSC. Of the baselines, HSIC-based MSV-gHSIC shows the best performance, except in comparison with MSV-IG under the global structure-view combination strategy on the DBLP graph-bag data in Fig. 8(b). Although MSV-gHSIC obtains high accuracy during the last few iterations, as shown in Figs. 8(c) and 9(c), this baseline

TABLE I

BAG-LEVEL t-TEST RESULTS. A, B, C, AND D DENOTE MSVBL, LOCAL MSV, GLOBAL MSV, AND ENSEMBLE MSV, RESPECTIVELY

	DBLP Graph Bag Data			Image Graph Bag Data		
	A-B	A-C	A-D	A-B	A-C	A-D
MSV-TopK	1.84E-11	6.79E-12	9.09E-12	5.66E-12	5.55E-12	4.67E-13
MSV-IG	3.87E-09	5.83E-09	8.05E-10	6.22E-10	1.85E-11	1.93E-09
MSV-gSSC	1.79E-07	2.60E-03	1.84E-07	4.18E-05	2.32E-05	2.86E-09
MSV-gHSIC	1.66E-04	2.11E-06	3.99E-06	8.44E-06	1.70E-04	1.04E-08

TABLE II

GRAPH-LEVEL t-TEST RESULTS. A, B, C, AND D DENOTE MSVBL, LOCAL MSV, GLOBAL MSV, AND ENSEMBLE MSV, RESPECTIVELY

	DBLP Graph Bag Data			Image Graph Bag Data		
	A-B	A-C	A-D	A-B	A-C	A-D
MSV-TopK	1.30E-10	2.41E-12	2.86E-08	7.87E-13	1.38E-12	4.54E-11
MSV-IG	1.05E-09	2.37E-09	1.97E-09	5.83E-10	1.73E-10	2.21E-11
MSV-gSSC	7.77E-09	3.18E-07	4.30E-08	2.14E-09	2.48E-11	4.10E-11
MSV-gHSIC	6.87E-10	4.28E-10	5.21E-06	5.95E-09	3.12E-10	5.57E-08

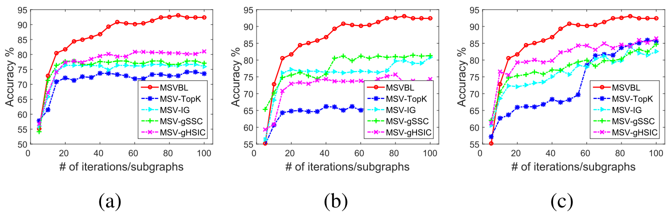


Fig. 10. Graph-level comparisons on *DBLP bag constrained graph data set* with different structure-view combination approaches. (a) Local MSV. (b) Global MSV. (c) Ensemble MSV.

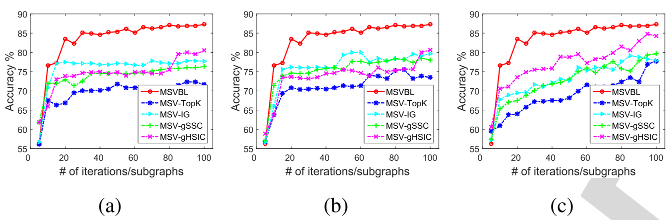


Fig. 11. Graph-level comparisons on *image bag constrained graph data set* with different structure-view combination approaches. (a) Local MSV. (b) Global MSV. (c) Ensemble MSV.

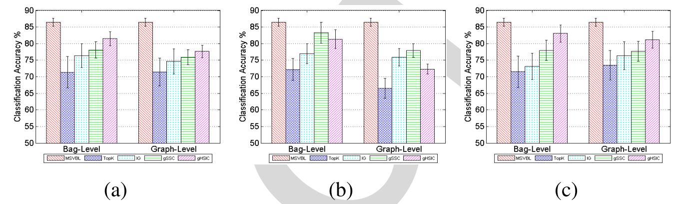


Fig. 12. Average results on *DBLP graph bag data set* with different structure-view combination approaches at bag and graph levels. (a) Local MSV. (b) Global MSV. (c) Ensemble MSV.

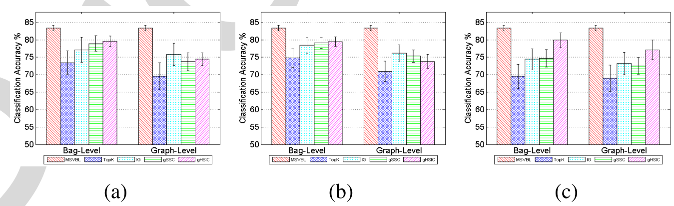


Fig. 13. Average results on *image graph bag data set* with different structure-view combination approaches at bag and graph levels. (a) Local MSV. (b) Global MSV. (c) Ensemble MSV.

still cannot outperform the best achievement of the proposed MSVBL.

To further demonstrate that MSVBL is indeed statistically superior to the bag-level MSV baselines, we report the pairwise t-test (with confidence level $\alpha = 0.05$) to validate the statistical significance in Table I, where each entry (value) denotes the p-value for a t-test between two algorithms, and a p-value less than $\alpha = 0.05$ indicates that the difference is statistically significant. The results in Table I on both bag constrained graph data sets confirm that MSVBL statistically outperforms local, global, and ensemble MSV in all cases.

2) Comparison With Graph-Level Evaluation Criteria: The results in each subfigure of Figs. 10 and 11 report the comparison with graph-level evaluation criteria under a special structure-view combination strategy. As expected, all graph-level subgraph evaluation criteria under any structure-view combination strategy are inferior to the proposed MSVBL, which should contribute to the dual bag- and graph-level mechanisms. In Table II, we report the pairwise t-test with confidence level $\alpha = 0.05$ to demonstrate the statistical performance of the proposed MSVBL. The p-values (less than 0.05) in each entry assert that MSVBL statistically and significantly outperforms graph-level MSV-based learning methods MSV-TopK, MSV-IG, MSV-gSSC, and MSV-gHSIC under all three structure-view combination strategies.

When the number of subgraph features is sufficiently large (e.g., more than 90), all baselines achieve similar performance. The information theory-based approach MSV-IG performs better than the approach at bag level, which is inferior to other discriminative approaches in Section V-D1. For instance, MSV-IG achieves better performance than discriminative MSV-gSSC on the image graph bag data set, as shown in Fig. 11(a)–(c). Moreover, MSV-IG is superior to the best bag-level baseline MSV-gHSIC under the local MSV strategy on the image graph bag data set [Fig. 11(a)], and the global structure-view combination strategy on both data sets [Figs. 10(b) and 11(b)]. This is possibly because the graphs at graph level may provide more information than bags. The graph-level methods directly propagate bag labels to graphs inside each bag. This can lead to a situation in which some graphs in the positive graph-bags may have incorrect labels, which results in performance degradation for graph-level MSV-gSSC and MSV-gHSIC (both need to utilize the label information).

For the purpose of comparing the same subgraph evaluation criteria under different estimation levels, we report the average accuracy in Figs. 12 and 13, where each subfigure [e.g., Fig. 13(a)] corresponds to a specific structure-view combination strategy (e.g., local strategy), summarizing both graph- and bag-level subgraph evaluation criteria.

TABLE III

BEST ACCURACY RESULT OF MSVBL VERSUS DIFFERENT BAG- OR GRAPH-LEVEL SUBGRAPH EVALUATION CRITERIA UNDER DIFFERENT STRUCTURE-VIEW COMBINATION STRATEGIES, OVER ITERATIONS (SUBGRAPHS) VARYING FROM 1 TO 100 ON DBLP BAG CONSTRAINED GRAPH DATA

Accuracy %	DBLP Bag Constrained Graph Data					
MI Classfiers	Different Multi-Structure-View (MSV) Combination Strategies					
	Local MSV		Global MSV		Ensemble MSV	
	B-Level	*G-Level*	*B-Level*	*G-Level*	*B-Level*	*G-Level*
MSVBL	**93.17±1.02**	**93.17±1.02**	**93.17±1.02**	**93.17±1.02**	**93.17±1.02**	**93.17±1.02**
MSV-TopK	73.83±3.92	74.17±4.32	75.17±3.25	72.50±3.51	79.83±4.33	86.00±4.06
MSV-IG	79.67±3.04	76.67±3.37	80.33±3.10	80.83±2.64	77.00±3.08	84.00±3.12
MSV-gSSC	80.83±2.47	77.83±1.84	86.17±2.88	81.50±2.45	83.17±2.02	84.67±2.35
MSV-gHSIC	84.33±1.89	81.00±1.93	84.50±1.84	75.67±2.07	90.33±2.23	86.50±2.36

TABLE IV

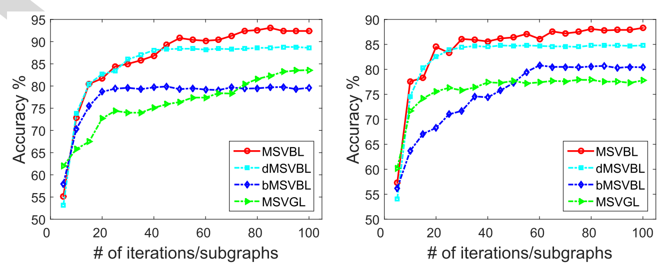
BEST ACCURACY RESULT OF MSVBL VERSUS DIFFERENT BAG- OR GRAPH-LEVEL SUBGRAPH EVALUATION CRITERIA UNDER DIFFERENT STRUCTURE-VIEW COMBINATION STRATEGIES, OVER ITERATIONS (SUBGRAPHS) VARYING FROM 1 TO 100 ON IMAGE BAG CONSTRAINED GRAPH DATA

Accuracy %	Image Bag Constrained Graph Data					
MI Classfiers	Different Multi-Structure-View (MSV) Combination Strategies					
	Local MSV		Global MSV		Ensemble MSV	
	B-Level	*G-Level*	*B-Level*	*G-Level*	*B-Level*	*G-Level*
MSVBL	**87.33±0.83**	**87.33±0.83**	**87.33±0.83**	**87.33±0.83**	**87.33±0.83**	**87.33±0.83**
MSV-TopK	76.00±3.71	72.33±3.96	77.00±3.53	75.50±4.03	74.33±3.74	77.67±3.85
MSV-IG	81.83±3.06	77.83±3.22	81.33±2.92	80.00±3.35	81.83±3.66	79.17±3.48
MSV-gSSC	82.00±2.57	76.33±2.64	81.33±2.22	78.50±2.35	82.67±2.35	79.67±2.86
MSV-gHSIC	83.17±1.86	80.50±2.04	83.33±1.57	80.67±1.92	86.17±2.03	84.83±2.19

In most cases, the subgraph evaluation criteria at bag-level are approximately 5% more accurate on both the DBLP and Image graph bag data sets. The only exception in Fig. 12(c) is that the graph-level TopK and IG approaches, under the ensemble structure-view combination strategy, perform 2% better than the related bag-level versions. By comparing the best accuracy over 100 iterations or subgraphs in Tables III and IV, we find that the bag-level subgraph evaluation criterion shows more improvement over graph-level baselines.

3) Internal Performance Analysis in MSVBL: The above-mentioned comparison results with the bag- and graph-level baselines have demonstrated the superiority of the proposed MSVBL. Indeed, because MSVBL includes two relatively independent components: *1) dual bag- and graph-level mechanism* and *2) discriminative subgraph candidate generation*, we want to carry out an internal performance study to better understand the actual role of each component. To investigate the efficiency of the dual level (unified bag- and graph-level) framework used in MSVBL, we implement an MSVBL version without using the graph level constraint, namely, bMSVBL. In consideration of the discriminative subgraph search used in MSVBL, another type of baseline dMSVBL approach that does not utilize the bag constrained discriminative score for subgraph candidate generation is also implemented to further demonstrate the distinct performance of MSVBL.

The detailed experimental results are reported in Fig. 14(a) and (b) for both the DBLP and Image graph bag data sets. dMSVBL is inferior to MSVBL when the subgraphs are relatively adequate (i.e., ≥40). On the other hand, MSVBL constantly outperforms bMSVBL without using the graph-level constraint. The results also show that when the number of subgraphs is less than 40, dMSVBL without bag constrained discriminative subgraph selection achieves comparable performance to the proposed MSVBL, which indicates that effective discriminative subgraph features cannot be identified with an insufficient number of subgraphs. This observation is consistent with the bag constrained subgraph quality analysis in Section V-D4.



(a) (b)

Fig. 14. Experimental results for MSVBL on (a) DBLP and (b) image graph bag data set.

In addition, graph-level approaches directly propagate bag labels to graphs. This transfers the problem to an *up-to-date graph learning task with multiple structure views* [11], where the learning approach MSVGL is also used for comparison

TABLE V

PAIRWISE t-TEST RESULTS. A DENOTES THE PROPOSED MSVBL, AND B, C, AND D DENOTE dMSVBL, bMSVBL, AND MSVGL, RESPECTIVELY

DBLP			Image		
A-B	A-C	A-D	A-B	A-C	A-D
1.40E-03	5.47E-08	3.01E-09	2.91E-06	4.55E-10	2.64E-10

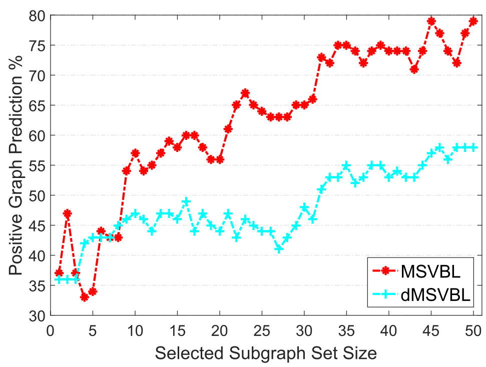


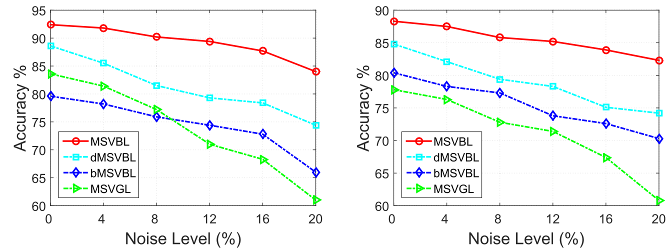
Fig. 15. Bag constrained subgraph quality on image graph bag data set.

with the proposed MSVBL. MSVGL first explores an optimal set of subgraphs as features to transfer MSV graphs into feature-vectors, with an AdaBoost [42] classifier being trained for final prediction. The results in Fig. 14(a) and (b) show that, in spite of the acceptable performance MSVGL obtains, it cannot reach the best performance achieved by MSVBL.

In Table V, we report the pairwise t-test with confidence level $\alpha = 0.05$. The p-values (less than 0.05) in each entry confirm that MSVBL statistically significantly outperforms bMSVBL, dMSVBL, and the state-of-the-art MSVGL.

4) Bag Constrained Subgraph Quality Analysis: To validate the quality of the selected subgraph set, and check whether the informative subgraphs chosen by the proposed MSVBL can identify genuinely positive patterns, we report the results of the Image graph bag data in Fig. 15. In this figure, the x-axis denotes selected subgraph size. The y-axis denotes the precision of positive patterns, calculated by selecting the "most positive graph" for each positive bag (i.e., the graph has the farthest distance from those graphs in negative bags based on the subgraph feature graph representation (Definition 5). At the beginning of the subgraph generation, both MSVBL and dMSVBL have discriminative score criteria, so cannot obtain an accurate positive graph prediction, mainly because a small quantity of the subgraph set has very limited discriminative power. As the size of the subgraph set grows, MSVBL continuously increases and outperforms dMSVBL, which is attributed to the bag constrained discrimination used for subgraph mining in the proposed MSVBL approach.

5) Sensitivity to Noisy Graph Bag Data: To validate that the proposed MSVBL is indeed robust and effective in handling noise in the bag constrained graph data, we investigate the noise sensitivity of MSVBL and baseline methods, including dMSVBL, bMSVBL, and MSVGL (the state-of-the-art graph learning task with multiple structure views) on both DBLP and Image graph bag sets. Following similar settings as those in [61] and [62], we manually inject noise into the graph bag sets by randomly flipping the class labels (i.e., changing a positive graph bag to negative, and vice versa) of $s\%$ graph bags in the training data. As a result, the training set has $2{*}s\%$ graph bags with noisy labels (called noisy graph bags).



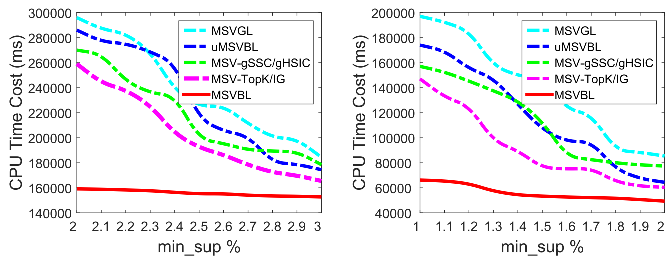
(a) (b)

Fig. 16. Comparisons on noisy graph bag set with respect to different noise levels (s values) on (a) DBLP and (b) image graph bag data set.

The results in Fig. 16 show that the proposed MSVBL is more robust than dMSVBL, bMSVBL, and MSVGL. This validates that combining cross structure-view subgraph feature exploration and learning indeed help MSVBL to effectively handle bag constrained graph data with noise. The increase in noise levels results in a deterioration in accuracy for all algorithms. This is because noise complicates the decision boundaries and makes it difficult for the learner to separate positive and negative classes. In contrast to MSVBL, MSVGL seems to be the most sensitive to labeling noise and suffers the most performance loss; this is because MSVGL only considers the graph level and directly propagates bag labels to graphs inside each bag. A mislabeled noisy graph bag will generate several noisy graphs, which significantly deteriorates the quality of the hyperplanes learned from the data.

6) Time Complexity Analysis: All the methods used in this paper have two major components: 1) subgraph mining and 2) classifier building. The baseline approaches MSV-TopK and MSV-IG under all three structure-view combination strategies (i.e., local, global, and ensemble MSV) take $O(gSpan) = O(l(q))$ for subgraph mining, where q is the number of graphs, with l being the function based on the total number of vertices and edges. In contrast, MSV-gSSC, MSV-gHSIC, and the state-of-the-art MSVGL baseline approaches have the complexity of $O(l(q) + q^2)$, where $O(q^2)$ reflects the informative subgraph evaluation. All the MSV-based baseline approaches use MIBoost as the classifier, where decision dump is used as the weak learner. The computational cost is $O(mq)$, where m is the maximum number of iterations. To sum up, the overall complexity of MSV-TopK and MSV-IG is $O(l(q) + mq)$. MSV-gSSC, MSV-gHSIC. and the state-of-the-art MSVGL will cost $O(l(q) + q^2 + mq)$.

The time complexity of subgraph mining in the proposed MSVBL will take $O(\bar{l}(q)) \ll O(l(q))$, because the proposed pruning strategy in Section IV-D significantly reduces the subgraph search time. MSVBL uses a linear programming for classification with $O(m(p + q^-))$, where p is the number of bags and q^- is the number of graphs in



(a) (b)

Fig. 17. Average CPU runtime comparison between MSVBL versus uMSVBL, dMSVBL, bMSVBL, and MSVGL with respect to different min_sup values on (a) DBLP and (b) image graph bag data set.

negative bags. Therefore, the corresponding overall complexity is $O(\bar{l}(q) + m(p + q^-))$, $O(\bar{l}(q)) \ll O(l(q))$.

7) Efficiency of the Pruning Strategy: For the purpose of evaluating the efficiency of the pruning module of MSVBL as described in Section IV-D, we implement a uMSVBL approach with no pruning module and compare its runtime performance with MSVBL, from which we can demonstrate the efficiency of the pruning module. In our implementation, uMSVBL first exploits gSpan to mine a frequent subgraph set, and then finds the optimal subgraph features by applying the same criteria as MSVBL. We also report the runtime performance for the MSV-based baselines and the state-of-the-art MSVGL. Because the MSV-TopK and MSV-IG have similar runtime performance, we use only one line MSV-TopK/IG to represent them. The same case can be found in MSV-gSSC/gHSIC.

The results in Fig. 17 show that increasing min_sup values results in the decrease in runtime of unbounded uMSVBL, MSV-TopK/IG, MSV-gSSC/gHSIC, and MSVGL, mainly because a larger min_sup value reduces the number of subgraph candidates for validation. By using a pruning strategy (i.e., the constraints including threshold min_sup and upper bound $\hat{\mathfrak{L}}_{g_s} = \max(\zeta_{g_s}^-, \zeta_{g_s}^+) + \boldsymbol{f}_s^\top \hat{L} \boldsymbol{f}_s$ as shown in Algorithm 1), MSVBL's runtime performance is relatively stable with respect to different min_sup values. This observation demonstrates the superiority on runtime performance over the unbounded version, especially when min_sup is small. Of all the MSV-based methods, MSV-gSSC/gHSIC consumes more time than MSV-TopK/IG, because the calculation of the discriminative subgraph criteria (gSSC/gHSIC) is more complicated than IG or TopK. Overall, MSVGL is the most time-consuming, because it requires extra time to ensure minimum redundancy.

VI. Conclusion and Future Work

This paper investigated a novel bag constrained graph classification task under multiple structure views, where the object for classification is a graph bag whose class label is only available at the bag level (but not available for graphs inside each bag). We argued that many real-world objects contain structure information from different structure views, and MSV bag constrained graph representation provides an effective way to preserve structure and complicated features of the object for learning. To build a learning model for MSV bag constrained graph classification, we iteratively select the most discriminative subgraphs, across different structure views, to minimize loss on a learning objective function. By joint regularization across multiple structure views, and enforcing labeling constraints at bag and graph levels MSVBL is able to discover the most effective subgraph features across all structure views to directly optimize learning. The key contribution of this paper, compared with existing works, is threefold: 1) a new MSV bag constrained graph classification problem formulation to advance the fundamental graph classification task; 2) a cross structure-view search space pruning strategy; and 3) a combined cross structure-view subgraph feature exploration and learning method.

We believe that the proposed multiple structure-view-based graph classification opens a new opportunity to expand existing multi-instance learning and multiview learning to increasingly popular graph applications. Although all techniques proposed in this paper are based on using frequent subgraphs to represent different structure views, the principle of combining graph- and bag-level constraints can be extended to many other types of approach, such as graph kernel and graph matching [63] techniques.

References

[1] X. Kong and P. S. Yu, "Semi-supervised feature selection for graph classification," in *Proc. KDD*, Jul. 2010, pp. 793–802.
[2] Z. Harchaoui and F. Bach, "Image classification with segmentation graph kernels," in *Proc. CVPR*, Jun. 2007, pp. 1–8.
[3] X. Kong and P. S. Yu, "Brain network analysis: A data mining perspective," *SIGKDD Explorations Newslett.*, vol. 15, no. 2, pp. 30–38, Dec. 2014.
[4] C. Aggarwal and K. Subbian, "Evolutionary network analysis: A survey," *ACM Comput. Surveys*, vol. 47, no. 1, Jul. 2014, Art. no. 10.
[5] A. Quek, Z. Wang, J. Zhang, and D. Feng, "Structural image classification with graph neural networks," in *Proc. DICTA*, Dec. 2011, pp. 416–421.
[6] R. Angelova and G. Weikum, "Graph-based text classification: Learn from your neighbors," in *Proc. SIGIR*, Aug. 2006, pp. 485–492.
[7] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "Learning sparse kernel classifiers for multi-instance classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 9, pp. 1377–1389, Sep. 2013.
[8] Z. Zhou, "Multi-instance learning: A survey," Nanjing Univ., Nanjing, China, Tech. Rep., 2004.
[9] J. Wu, X. Zhu, C. Zhang, and P. S. Yu, "Bag constrained structure pattern mining for multi-graph classification," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 10, pp. 2382–2396, Oct. 2014.
[10] J. Wu, S. Pan, X. Zhu, and Z. Cai, "Boosting for multi-graph classification," *IEEE Trans. Cybern.*, vol. 45, no. 3, pp. 416–429, Mar. 2015.
[11] J. Wu, Z. Hong, S. Pan, X. Zhu, Z. Cai, and C. Zhang, "Multi-graph-view subgraph mining for graph classification," *Knowl. Inf. Syst.*, vol. 48, no. 1, pp. 29–54, 2016.
[12] M. Mayo and E. Frank, "Experiments with multi-view multi-instance learning for supervised image classification," in *Proc. IVCNZ*, Dec. 2011, pp. 363–369.
[13] X. Wu *et al.*, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.
[14] R.-H. Li, J. X. Yu, X. Huang, and H. Cheng, "Random-walk domination in large graphs," in *Proc. ICDE*, Apr. 2014, pp. 736–747.
[15] Y. Wang, W. Zhang, L. Wu, X. Lin, and X. Zhao, "Unsupervised metric fusion over multiview data by graph random walk-based cross-view diffusion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 1, pp. 57–70, Jan. 2017.
[16] K.-M. Lee, B. Min, and K. Goh, "Towards real-world complexity: An introduction to multiplex networks," *Eur. Phys. J. B*, vol. 88, no. 2, p. 48, Feb. 2015.
[17] X. Yan and J. Han, "gSpan: Graph-based substructure pattern mining," in *Proc. ICDM*, Dec. 2002, pp. 721–724.

[18] J. Huan, W. Wang, and J. Prins, "Efficient mining of frequent subgraphs in the presence of isomorphism," in *Proc. ICDM*, Nov. 2003, pp. 549–552.
[19] H. Fei and J. Huan, "Structure feature selection for graph classification," in *Proc. CIKM*, Oct. 2008, pp. 991–1000.
[20] X. Yan, H. Cheng, J. Han, and P. S. Yu, "Mining significant graph patterns by leap search," in *Proc. SIGMOD*, Jun. 2008, pp. 433–444.
[21] H. Saigo, N. Krämer, and K. Tsuda, "Partial least squares regression for graph mining," in *Proc. KDD*, Aug. 2008, pp. 578–586.
[22] M. Thoma *et al.*, "Near-optimal supervised feature selection among frequent subgraphs," in *Proc. SDM*, Apr. 2009, pp. 1076–1087.
[23] Y. Zhu, J. X. Yu, H. Cheng, and L. Qin, "Graph classification: A diversified discriminative feature selection approach," in *Proc. CIKM*, Nov. 2012, pp. 205–214.
[24] X. Kong, W. Fan, and P. S. Yu, "Dual active feature and sample selection for graph classification," in *Proc. KDD*, Aug. 2011, pp. 654–662.
[25] X. Kong, A. B. Ragin, X. Wang, and P. S. Yu, "Discriminative feature selection for uncertain graph classification," in *Proc. SDM*, May 2013, pp. 82–93.
[26] X. Kong and P. S. Yu, "gMLC: A multi-label feature selection framework for graph classification," *Knowl. Inf. Syst.*, vol. 31, no. 2, pp. 281–305, 2012.
[27] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert–Schmidt norms," in *Proc. ALT*, Oct. 2005, pp. 63–77.
[28] Y. Zhao, X. Kong, and P. S. Yu, "Positive and unlabeled learning for graph classification," in *Proc. ICDM*, Dec. 2011, pp. 962–971.
[29] S. Pan, J. Wu, X. Zhu, and C. Zhang, "Graph ensemble boosting for imbalanced noisy graph stream classification," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 954–968, May 2015.
[30] S. Pan, J. Wu, X. Zhu, G. Long, and C. Zhang, "Task sensitive feature exploration and learning for multitask graph classification," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 744–758, Mar. 2017.
[31] N. Pržulj, "Biological network comparison using graphlet degree distribution," *Bioinformatics*, vol. 23, no. 2, pp. e177–e183, 2007.
[32] R. N. Lichtenwalter and N. V. Chawla, "Vertex collocation profiles: Theory, computation, and results," *SpringerPlus*, vol. 3, no. 1, p. 116, 2014.
[33] T. G. Dietterich, R. T. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, nos. 1–2, pp. 31–71, 1997.
[34] J. Wang, "Solving multiple-instance problem: A lazy learning approach," in *Proc. ICML*, 2000, pp. 1119–1125.
[35] L. Bjerring and E. Frank, "Beyond Trees: Adopting MITI to learn rules and ensemble classifiers for multi-instance data," in *Proc. AI*, Dec. 2011, pp. 41–50.
[36] D. T. Nguyen, C. D. Nguyen, R. Hargraves, L. A. Kurgan, and K. J. Cios, "mi-DS: Multiple-instance learning algorithm," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 143–154, Feb. 2013.
[37] V. Cheplygina, D. M. J. Tax, and M. Loog, "Multiple instance learning with bag dissimilarities," *Pattern Recognit.*, vol. 48, no. 1, pp. 264–275, 2015.
[38] V. Cheplygina, D. M. J. Tax, and M. Loog, "Dissimilarity-based ensembles for multiple instance learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1379–1391, Jun. 2015.
[39] M.-L. Zhang and Z.-H. Zhou, "Improve multi-instance neural networks through feature selection," *Neural Process. Lett.*, vol. 19, no. 1, pp. 1–10, 2004.
[40] M.-L. Zhang and Z.-H. Zhou, "Adapting RBF neural networks to multi-instance learning," *Neural Process. Lett.*, vol. 23, no. 1, pp. 1–26, 2006.
[41] X. Xu and E. Frank, "Logistic regression and boosting for labeled bags of instances," in *Proc. PAKDD*, May 2004, pp. 272–281.
[42] M. Telgarsky, "A primal-dual convergence analysis of boosting," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 561–606, Mar. 2012.
[43] X. Xu, "Statistical learning in multiple instance problems," Ph.D. dissertation, Dept. Comput. Sci., Univ. Waikato, Hamilton, New Zealand, 2003.
[44] Y. Chen, J. Bi, and J. Z. Wang, "MILES: Multiple-instance learning via embedded instance selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1931–1947, Dec. 2006.
[45] Z. Fu, A. Robles-Kelly, and J. Zhou, "MILIS: Multiple instance learning with instance selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 958–977, May 2010.
[46] D. Zhang, F. Wang, L. Si, and T. Li, "Maximum margin multiple instance clustering with applications to image and text clustering," *IEEE Trans. Neural Netw.*, vol. 22, no. 5, pp. 739–751, May 2011.
[47] X.-S. Wei, J. Wu, and Z.-H. Zhou, "Scalable algorithms for multi-instance learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 4, pp. 975–987, Apr. 2016.
[48] B. Xie, Y. Mu, D. Tao, and K. Huang, "m-SNE: Multiview stochastic neighbor embedding," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 41, no. 4, pp. 1088–1096, Aug. 2011.
[49] J. Yu, D. Liu, D. Tao, and H. S. Seah, "On combining multiple features for cartoon character retrieval and clip synthesis," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 42, no. 5, pp. 1413–1427, Oct. 2012.
[50] R. Gan and J. Yin, "Feature selection in multi-instance learning," *Neural Comput. Appl.*, vol. 23, no. 3, pp. 907–912, 2013.
[51] J. Wu, X. Zhu, C. Zhang, and Z. Cai, "Multi-instance multi-graph dual embedding learning," in *Proc. ICDM*, Dec. 2013, pp. 827–836.
[52] J. Tang, X. Hu, H. Gao, and H. Liu, "Unsupervised feature selection for multi-view data in social media," in *Proc. SDM*, May 2013, pp. 270–278.
[53] J. Wu, S. Pan, X. Zhu, Z. Cai, and C. Zhang, "Multi-graph-view learning for complicated object classification," in *Proc. IJCAI*, Jun. 2015, pp. 3953–3959.
[54] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*. New York, NY, USA: McGraw-Hill, 1996.
[55] X. Luo, Z. Xu, J. Yu, and X. Chen, "Building association link network for semantic link on Web resources," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 3, pp. 482–494, Jul. 2011.
[56] J. Li and J. Z. Wang, "Real-time computerized annotation of pictures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 985–1002, Jun. 2008.
[57] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
[58] Z. Hong, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Tracking using multilevel quantizations," in *Proc. ECCV*, Sep. 2014, pp. 155–171.
[59] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Positive and unlabeled multi-graph learning," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 818–829, Apr. 2017.
[60] C. Jiang, F. Coenen, and M. Zito, "A survey of frequent subgraph mining algorithms," *Knowl. Eng. Rev.*, vol. 28, no. 1, pp. 75–105, Mar. 2013.
[61] Y. Xiao, B. Liu, L. Cao, J. Yin, and X. Wu, "SMILE: A similarity-based approach for multiple instance learning," in *Proc. ICDM*, Dec. 2010, pp. 589–598.
[62] W.-J. Li and D.-Y. Yeung, "MILD: Multiple-instance learning via disambiguation," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 1, pp. 76–89, Jan. 2010.
[63] M. Kivelä and M. A. Porter. (Jun. 2015). "Isomorphisms in multilayer networks." [Online]. Available: https://arxiv.org/abs/1506.00508

Jia Wu (M'16) received the Ph.D. degree in computer science from the University of Technology Sydney, Ultimo, NSW, Australia.

He is currently a Lecturer with the Department of Computing, Faculty of Science and Engineering, Macquarie University, Sydney. Prior to that, he was with the Centre for Artificial Intelligence, University of Technology Sydney. His current research interests include data mining and machine learning. Since 2009, he has authored or co-authored over 60 refereed journal and conference papers, such as the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON CYBERNETICS, *Pattern Recognition*, the International Joint Conference on Artificial Intelligence, AAAI Conference on Artificial Intelligence, International Conference on Data Engineering, the International Conference on Data Mining, SIAM International Conference on Data Mining, and the Conference on Information and Knowledge Management, in these areas.

Shirui Pan (M'16) received the Ph.D. degree in computer science from the University of Technology Sydney (UTS), Ultimo, NSW, Australia.

He is currently a Research Associate with the Centre for Artificial Intelligence, UTS. His current research interests include data mining and machine learning. To date, he has authored or co-authored over 20 research papers in top-tier journals and conferences, including the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON CYBERNETICS, *Pattern Recognition*, the International Joint Conference on Artificial Intelligence, ICDE, the International Conference on Data Mining, and SDM.

Xingquan Zhu (SM'12) received the Ph.D. degree in computer science from Fudan University, Shanghai, China.

He is currently an Associate Professor with the Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, USA, and a Distinguished Visiting Professor (Eastern Scholar) with the Shanghai Institutions of Higher Learning, Shanghai, China. His current research interests include data mining, machine learning, and multimedia systems. Since 2000, he has authored or co-authored over 200 refereed journal and conference papers in these areas, including two Best Paper Awards and one Best Student Paper Award.

Dr. Zhu was an Associate Editor of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING from 2008 to 2012. Since 2014, he has been an Associate Editor of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.

Chengqi Zhang (SM'95) received the Ph.D. degree from The University of Queensland, Brisbane, QLD, Australia, in 1991, and the D.Sc. degree (Higher Doctorate) from Deakin University, Geelong, VIC, Australia, in 2002.

Since 2001, he has been a Professor of Information Technology with the University of Technology Sydney (UTS), Ultimo, NSW, Australia, where he has been the Director of the UTS Priority Investment Research Centre for Quantum Computation and Intelligent Systems since 2008. His current research interests include data mining and its applications.

Dr. Zhang is a fellow of the Australian Computer Society. He is a General Co-Chair of KDD 2015, Sydney, and the Local Arrangements Chair of IJCAI-2017, Melbourne. He has served as an Associate Editor for three international journals, including the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING from 2005 to 2008.

Philip S. Yu (F'93) has spent most of his career with IBM, Yorktown Heights, NY, USA, where he was a Manager of the Software Tools and Techniques Group, Watson Research Center. He is currently a Distinguished Professor in computer science with the University of Illinois at Chicago, Chicago, IL, USA, where he also holds the Wexler Chair in information technology. He has authored or co-authored over 780 papers in refereed journals and conferences. He holds or has applied for more than 250 U.S. patents. His current research interests include big data, data mining, data stream, database, and privacy.

Dr. Yu was a member of the IEEE Data Engineering Steering Committee. He is a fellow of the ACM. He has received several IBM honors, including two IBM Outstanding Innovation Awards, an Outstanding Technical Achievement Award, two Research Division Awards, and the 94th plateau of Invention Achievement Awards. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING from 2001 to 2004. He is the Editor-in-Chief of the *ACM Transactions on Knowledge Discovery from Data*. He is on the Steering Committee of the IEEE Conference on Data Mining and the ACM Conference on Information and Knowledge Management.