Under consideration for publication in Knowledge and Information
Systems

gMLC: a multi-label feature selection framework for graph classification

Xiangnan Kong, Philip S. Yu

Department of Computer Science, University of Illinois at Chicago, Chicago IL, USA

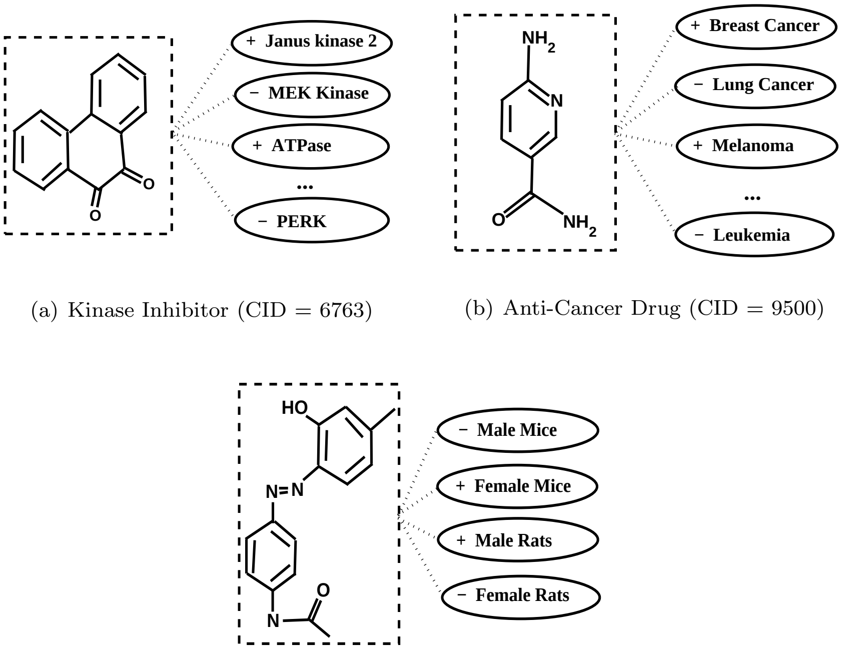
Abstract. Graph classification has been showing critical importance in a wide variety of applications, *e.g.* drug activity predictions and toxicology analysis. Current research on graph classification focuses on single-label settings. However, in many applications, each graph data can be assigned with a set of multiple labels simultaneously. Extracting good features using multiple labels of the graphs becomes an important step before graph classification. In this paper, we study the problem of multi-label feature selection for graph classification and propose a novel solution, called gMLC, to efficiently search for optimal subgraph features for graph objects with multiple labels. Different from existing feature selection methods in vector spaces which assume the feature set is given, we perform multi-label feature selection for graph data in a progressive way together with the subgraph feature mining process. We derive an evaluation criterion to estimate the dependence between subgraph features and multiple labels of graphs. Then a branch-and-bound algorithm is proposed to efficiently search for optimal subgraph features by judiciously pruning the subgraph search space using multiple labels. Empirical studies demonstrate that our feature selection approach can effectively boost multi-label graph classification performances and is more efficient by pruning the subgraph search space using multiple labels.

Keywords: Feature selection; Graph classification; Multi-label learning; Subgraph Pattern; Label correlation

1. Introduction

Due to the recent advances of data collection technology, many application fields are facing various data with complex structures, *e.g.*, chemical compounds, program flows and XML web documents. Different from traditional data in feature spaces, these data are not represented as feature vectors, but as graphs which

Received January 06, 2011
Revised April 04, 2011
Accepted April 16, 2011



(a) Kinase Inhibitor (CID = 6763)

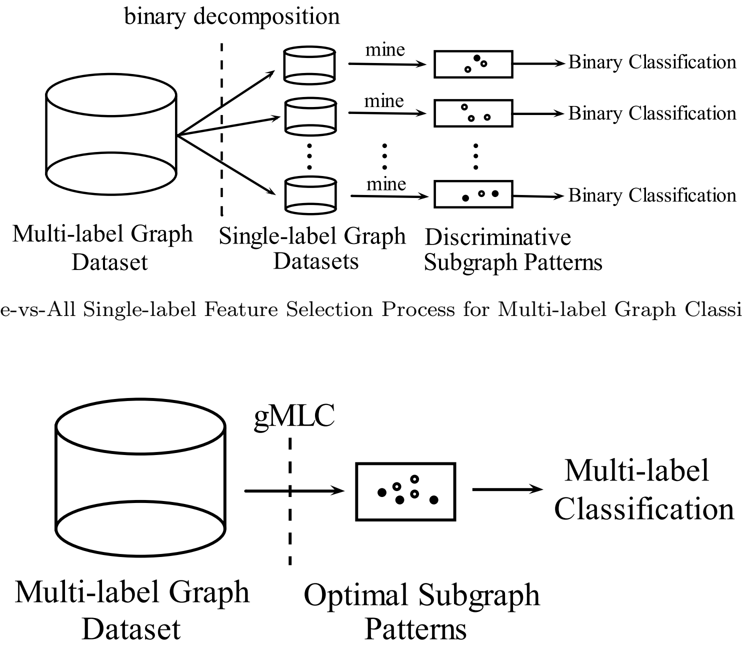
(b) Anti-Cancer Drug (CID = 9500)

(c) Toxicology Analysis (ID = TR222)

Fig. 1. Examples of multi-label graphs. a) In kinase inhibition, each molecule can inhibit the activities of multiple types of kinases; b) In anti-cancer prediction, each molecular medicine can have anti-cancer efficacies on multiple types of cancers; c) In toxicology analysis, each chemical compound has carcinogenicity activities in multiple animal models.

raise one fundamental challenge for data mining research: the complex structure and lack of vector representations (Chen et al., 2009; Tasourakakis and Faloutsos, 2010; Jia et al., 2011; Ying and Wu, 2010). An effective model for graph data should be able to extract or find a proper set of features for these graphs in order to perform analysis or management steps. Motivated by these challenges, graph mining research problems, in particular graph classification, have received considerable attention in the last decade.

In the literature, graph classification problem has been extensively studied. Conventional approaches focus on single-label classification problems (Yan et al., 2008; Thoma et al., 2009; Fei and Huan, 2010; Zou et al., 2010), which assume, explicitly or implicitly, that each graph has only one label. However, in many real-world applications, each graph can be assigned with more than one label. For example, in Figure 1, a chemical compound can inhibit the activities of multiple types of kinases, *e.g.*, *ATPase* and *MEK kinase*; One drug molecular can have anti-cancer efficacies on multiple types of cancers. The selection and discovery of drugs or kinase inhibitors can be significantly improved if these chemical molecules are automatically tagged with a set of multiple labels or potential efficacies. This setting is also known as multi-label classification where each instance can be associated with multiple categories. It has been shown useful in many real-world applications such as text categorization (McCallum,


binary decomposition
mine
mine
mine
Binary Classification
Binary Classification
Binary Classification
Multi-label Graph Dataset
Single-label Graph Datasets
Discriminative Subgraph Patterns
gMLC
Multi-label Classification
Multi-label Graph Dataset
Optimal Subgraph Patterns

(a) One-vs-All Single-label Feature Selection Process for Multi-label Graph Classification

(b) gMLC Feature Selection Process for Multi-label Graph Classification

Fig. 2. Two types of Feature Selection Processes for Multi-label Graph Classification

1999; Schapire and Singer, 2000) and bioinformatics (Elisseeff and Weston, 2002). Multi-label classification is particularly challenging on graph data. The reason is that, in the single-label case, conventional graph mining methods can extract or find one set of discriminative subgraph features for the single label concept within the graph dataset. But in multi-label cases, each graph contains multiple label concepts, and multiple sets of subgraph features should be mined, one for each label concept, in order to decide all the possible categories for each graph using binary classifiers (one-vs-all technique (Boutell et al., 2004)). Thus the time and memory used for classifying multi-label graph data is much larger than for the single-label graphs. A major difficulty in performing multi-label classification on graph data lies in the complex structure of graphs and lack of features which is useful for multiple labels concepts. Selecting a proper set of features for graph data becomes an essential and important procedure for multi-label graph classification.

Despite its value and significance, the multi-label feature selection problem for graph data has not been studied in this context so far. If we consider graph mining and multi-label classification as a whole, the major research challenges on multi-label feature selection for graph classification are as follows:

1. **Graph Data:** One fundamental problem in multi-label feature selection on graph data lies in the complex structures and lack of feature representations of graphs. Conventional feature selection approaches in vector spaces assume, explicitly or implicitly, that a full set of features is given before the feature

selection. In the context of graph data, however, the full set of features for a graph dataset, are usually too large or even infeasible to obtain. For example, in graph mining, the number of subgraph features grows exponentially with the size of the graphs, which makes it impossible to enumerate all the subgraph features before the feature selection.

2. **Multiple Labels:** Another fundamental problem in multi-label feature selection on graph data lies in the multiple label concepts for each graph, *i.e.* how to utilize the multiple label concepts in a graph dataset to find a proper set of subgraph features for classification tasks. Conventional feature selection in graph classification approaches focuses on single-labeled settings (Kudo et al., 2005; Yan et al., 2008; Thoma et al., 2009). The mining strategy of discriminative subgraph patterns strictly follows the assumption that each graph has only one label. However, in many real-world applications, one graph can usually be assigned with multiple labels simultaneously. Directly applying single-label graph feature selection methods by adopting the popular one-versus-all binary decomposition (Figure 2(a)), which performs feature selection on each label concept, will result in different sets of subgraph features on different classes. Thus most state-of-the-art multi-label classification approaches in vector spaces cannot be used, since they assume that the instances should have a same set of features in the input space (Schapire and Singer, 2000; Elisseeff and Weston, 2002).
3. **Label Correlations:** In many real-world applications, the multiple labels of graphs are usually *correlated*, not *independent* from each other. For example, in anti-cancer drug activity prediction tasks, chemical compounds which are active to one type of cancer are more likely to be active to some other related cancers. It is much desirable that the correlations between different labels be exploited in the feature selection process.

Figure 2(a) illustrates the process of directly applying single-label graph feature selection methods by adopting the popular one-versus-all binary decomposition. The problems with this approach are as follows:

- multiple sets of discriminative subgraph features, one for each label or label combination, should to be mined before the classification, which could be too expensive when the number of labels is large;
- the correlations among multiple labels of the graphs are ignored in the feature selection process. In addition, the correlations among labels may result in similar feature sets for different labels. Redundancies in these sets of discriminative subgraph features cause unnecessary time and memory costs, since many of the features are mined multiple times.

In this paper, we introduce a novel framework to the above problems by mining subgraph features using multiple labels of graphs. Our framework is illustrated in Figure 2(b). Different from existing single-label feature selection methods for graph data, our approach, called gMLC, can utilize multiple labels of graphs to find an optimal set of subgraph features for graph classification. We first derive an evaluation criterion for subgraph features, named gHSIC, based upon a given graph dataset with multiple labels. Then in order to avoid exhaustive enumeration of all subgraph features, we propose a branch-and-bound algorithm to efficiently search for optimal subgraph features by pruning the subgraph search space using multiple labels of graphs. Label correlations can also be considered in our proposed framework. In order to evaluate our proposed model, we perform comprehensive experiments on real-world multi-label graph classification tasks, which consist three real-world multi-label graph classification tasks, built on 18 conventional binary graph classification datasets. The experiments demonstrate that our feature selection approach can effectively boost multi-label graph classification performances. Moreover, we show that gMLC is more efficient by pruning the subgraph search space using multiple labels.

The rest of the paper is organized as follows. We start by a brief review on related work of graph feature selection and multi-label classification in Section 2. Then introduce the preliminary concepts, give the problem analysis and present the gHSIC criterion in Section 3 and Section 4; In Section 5, we derive a branch and bound algorithm gMLC based upon gHSIC. In Section 6, we discuss how to incorporate label correlations into the gMLC framework. Then Section 7 reports the experiment results. In Section 8, we conclude the paper.

2. Related Work

To the best of our knowledge, this paper is the first work addressing the multi-label feature selection problem for graph classification. Our work is related to both multi-label classification techniques and subgraph feature based graph mining. We briefly discuss both of them.

Multi-label learning deals with the classification problem where each instance can belong to multiple different classes simultaneously. Conventional multi-label approaches are focused on instances in vector spaces. One well-know type of approaches is binary relevance (one-vs-all technique (Boutell et al., 2004)), which transforms the multi-label problem into multiple binary classification problems, one for each label. Ml-knn(Zhang and Zhou, 2007) is one of the binary relevance methods, which extends the lazy learning algorithm, kNN, to a multi-label version. It employs label prior probabilities gained from each example's k nearest neighbors and use *maximum a posteriori* (MAP) principle to determine label set. Elisseeff and Weston (Elisseeff and Weston, 2002) presented a kernel method Rank-svm for multi-label classification, by minimizing a loss function named *ranking loss.* Extension of other traditional learning techniques have also been studied, such as probabilistic generative models (McCallum, 1999; Ueda and Saito, 2003), decision trees (Comité et al., 2003), maximal margin methods (Godbole and Sarawagi, 2004; Kazawa et al., 2005) and ensemble methods(G. Tsoumakas, 2007), *etc.*

Extracting subgraph features from graph data have also been investigated by many researchers. The goal of such approaches is to extract informative subgraph features from a set of graphs. Typically some filtering criteria are used. Upon whether considering the label information, there are two types of approaches: unsupervised and supervised. A typical evaluation criterion is frequency, which aims at collecting frequently appearing subgraph features. Most of the frequent subgraph feature extraction approaches are unsupervised. For example, Yan and Han develop a depth-first search algorithm: gSpan (Yan and Han, 2002). This algorithm builds a lexicographic order among graphs, and maps each graph to an unique minimum DFS code as its canonical label. Based on this lexicographic order, gSpan adopts the depth-first search strategy to mine frequent connected subgraphs efficiently. Many other frequent subgraph feature extraction approaches have been developed, *e.g.* AGM (Inokuchi et al., 2000), FSG (Kuramochi and Karypis, 2001), MoFa (Borgelt and Berthold, 2002), FFSM (Huan et al., 2003), and Gaston (Nijssen and Kok, 2004). Supervised subgraph feature extraction approaches have also been proposed in literature, such as LEAP (Yan et al., 2008), CORK (Thoma et al., 2009), which look for discriminative subgraph patterns for graph classifications, and gSSC (Kong and Yu, 2010) for semi-supervised classification.

Our approach is also relevant to graph feature selection approaches based on Hilbert-Schmidt independence criterion (Borgwardt, 2007), but there are significant differences between them. Previous graph feature selection approaches assume each graph object only has one label and they focus on evaluating subgraph features effectively using HSIC criterion and perform feature selection using frequent subgraph mining methods (gSpan) as black-boxes. However, our approach assumes that each graph can have multiple labels, and focuses on extracting good subgraph features efficiently by pruning the subgraph search space using branch and bound method inside gSpan. So, our method searches the pruned gSpan tree. In fact, we only generated and searched a much smaller tree than gSpan as the size of the search tree dominates the execution time.

3. Problem Formulation

Before presenting the feature selection model for multi-label graph classification, we first introduce the notations that will be used throughout this paper. Multi-label graph classification is the task of automatically classifying a graph object into a subset of predefined classes. Let $\mathcal{D} = \{G_1, \cdots, G_n\}$ denote the entire graph dataset, which consists of n graph objects, represented as *connected graphs*. The graphs in $\mathcal{D}$ are labeled by $\{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n\}$, where $\boldsymbol{y}_i \in \{0,1\}^Q$ denotes the multiple labels assigned to G_i. Here Q is the number of all possible labels within a label concept set $\mathcal{C}$.

Definition 3.1 (Connected Graph). A graph is represented as $G = (\mathcal{V}, E, \mathcal{L}, l)$, where $\mathcal{V}$ is a set of vertices $\mathcal{V} = \{v_1, \cdots, v_{n_v}\}$, $E \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges, $\mathcal{L}$ is the set of labels for the vertices and the edges. $l : \mathcal{V} \cup E \to \mathcal{L}$, l is a function assigning labels to the vertices and the edges. A connected graph is a graph such that there is a path between any pair of vertices.

Definition 3.2 (Multi-label Graph). A multi-label graph is a graph assigned with multiple class labels $(G, \boldsymbol{y})$, in which $\boldsymbol{y} = [y^1, \cdots, y^Q] \in \{0,1\}^Q$ denotes the multiple labels assigned to the graph G. $y^k = 1$ iff graph G is assigned with the k-th class label, 0 otherwise.

Definition 3.3 (Subgraph). Let $G' = (\mathcal{V}', E', \mathcal{L}', l')$ and $G = (\mathcal{V}, E, \mathcal{L}, l)$ be connected graphs. G' is a subgraph of G ($G' \subseteq G$) iff there exist an injective function $\psi : \mathcal{V}' \to \mathcal{V}$ s.t. (1) $\forall v \in \mathcal{V}'$, $l'(v) = l\left(\psi(v)\right)$; (2) $\forall (u,v) \in E'$, $(\psi(u), \psi(v)) \in E$ and $l'(u,v) = l\left(\psi(u), \psi(v)\right)$. If G' is a subgraph of G, then G is a supergraph of G'.

In our current solution, we focus on the subgraph-based graph classification problem, which assumes that a graph object G_i is represented as a binary vector $\boldsymbol{x}_i = [x_i^1, \cdots, x_i^m]^\top$ associated with a set of subgraph patterns $\{g_1, \cdots, g_m\}$. Here $x_i^k \in \{0,1\}$ is the binary feature of G_i corresponding to the subgraph pattern g_k, and $x_i^k = 1$ iff g_k is a subgraph of G_i ($g_k \subseteq G_i$).

The key issue of feature selection for multi-label graph classification is how to find the most informative subgraph patterns from a given multi-label graph dataset. So, in this paper, the studied research problem can be described as follows: in order to train an effective multi-label graph classifier, how to efficiently find a set of optimal subgraph features using multiple labels of graphs?

Mining the optimal subgraph features for multi-label graphs is a non-trivial task due to the following problems:

1) How to properly evaluate the usefulness of a set of subgraph features based upon multiple labels of graphs?
2) How to determine the optimal subgraph features within a reasonable amount of time by avoiding the exhaustive enumeration using multiple labels of the graphs? The subgraph feature space of graph objects are usually too large, since the number of subgraphs grows exponentially with the size of graphs. It is infeasible to completely enumerate all the subgraph features for a given graph dataset.
3) How to incorporate the correlations among different labels in the feature selection process?

In the following sections, we will first introduce the optimization framework for selecting informative subgraph features from multi-label graphs, and propose an efficient subgraph mining strategy using branch-and-bound to avoid exhaustive enumeration. Then we propose solutions to incorporate label correlations into the feature selection process.

4. Optimization Framework

In this section, we address the problem 1) discussed in Section 3 by defining the subgraph feature selection for multi-label graph classification as an optimization problem. The goal is to find an optimal set of subgraph features based on the multiple labels of graphs. Formally, let us introduce the following notations:

- $\mathcal{S} = \{g_1, g_2, \cdots, g_m\}$: a given set of subgraph features, which we use to predict a set of multiple labels for each graph object. Usually there is only a subset of the subgraph features $\mathcal{T} \subseteq \mathcal{S}$ relevant to the multi-label graph classification task.
- $\mathcal{T}^*$: the optimal set of subgraph features $\mathcal{T}^* \subseteq \mathcal{S}$.
- $\mathcal{E}(\mathcal{T})$: an evaluation criterion to estimate the usefulness of subgraph feature subsets $\mathcal{T}$.
- X: the matrix consisting binary feature vectors using $\mathcal{S}$ to represent the graph dataset $\{G_1, G_2, \cdots, G_n\}$. $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n] = [\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_m]^\top \in \{0,1\}^{m \times n}$, where $X = [X_{ij}]_{m \times n}$, $X_{ij} = 1$ iff $g_i \subseteq G_j$.

We adopt the following optimization framework to select an optimal subgraph feature set:

$$\begin{aligned} &\mathcal{T}^* = \underset{\mathcal{T} \subseteq \mathcal{S}}{\arg\max}\, \mathcal{E}(\mathcal{T}) \qquad (1) \\ &\text{s.t.} \quad |\mathcal{T}| \leq t, \end{aligned}$$

where t denotes the maximum number of feature selected, $|\cdot|$ is the size of the feature set. Similar optimization framework to select an optimal subgraph feature set has also been defined in the context of single-label graph feature selection in (Thoma et al., 2009; Borgwardt, 2007). In Eq. 1 the objective function has two parts: the evaluation criterion $\mathcal{E}$ and the subgraph features of graphs $\mathcal{S}$.

For evaluation criterion, we assume that the optimal subgraph features should have the following property, *i.e. Dependence Maximization*: Optimal subgraph features should maximize the dependence between the subgraph features of graph objects and their multiple labels. This indicates that two graph objects with similar sets of multiple labels are likely to have similar subgraph features. Similar assumptions have also been used for multi-label dimensionality reduction in vector spaces (Zhang and Zhou, 2008).

Many criteria that can be used as dependence evaluation between subgraph features and multiple labels. In this paper, we derive a subgraph evaluation criterion for multi-label graph classification based upon a dependence evaluation criterion named Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al., 2005). We briefly introduce the Hilbert-Schmidt Independence Criterion as a dependence measure between two variables in kernel space. In our case, the target is to derive a dependence measure between the graph objects using a set of subgraph features and their multiple labels. Suppose we have two reproducing kernel Hilbert spaces (RKHS) of functions $\mathcal{G}$ and $\mathcal{F}$, with feature mapping $\phi(G_i) \in \mathcal{G}$ and $\psi(\boldsymbol{y}_i) \in \mathcal{F}$. The corresponding kernel functions are denoted as $\langle \phi(G_i), \phi(G_j) \rangle_{\mathcal{G}} = k(G_i, G_j)$ and $\langle \psi(\boldsymbol{y}_i), \psi(\boldsymbol{y}_j) \rangle_{\mathcal{F}} = k'(\boldsymbol{y}_i, \boldsymbol{y}_j)$. Let C be a covariance operator defined as

$$C = \mathrm{E}\left\{[p(G_i) - \mathrm{E}(p(G_i))][p'(\boldsymbol{y}_i) - \mathrm{E}(p'(\boldsymbol{y}_i))]\right\}$$

for all $p \in \mathcal{G}$ and $p' \in \mathcal{F}$.

Then the HSIC is defined as the Hilbert-Schmidt norm of the operator C, *i.e.* $\|C\|^2_{HS}$. Given a sample of data, an empirical estimate of HSIC is HSIC = $\mathrm{tr}(\mathbf{K\ H\ L\ H})$, where $\mathrm{tr}(\cdot)$ is the trace of matrix and $\mathbf{H} = [H_{ij}]_{n\times n}$, $H_{ij} = \delta_{ij} - 1/n$, δ_{ij} is the indicator function which takes 1 when $i = j$ and 0 otherwise. $\mathbf{K}$ and $\mathbf{L}$ are kernel matrices on the samples with respect to the kernel functions $k(\cdot,\cdot)$ and $l(\cdot,\cdot)$.

There are basically two reasons for using HSIC measure for feature selection:

- The HSIC can evaluate the dependence of two variables in kernel space, which is more general than measuring dependence in the original space. HSIC has been widely used for feature selection on single-label cases. It can also be extended to feature selection in multi-label cases. Moreover, correlations among different labels can naturally be considered in our framework by adopting advanced kernels into the HSIC. Thus it is more effective and flexible to measure the dependence in the kernel space.
- In addition to many good theoretical properties, HSIC has a very simple empirical estimator, $\mathrm{tr}(\mathbf{KHLH})$, which we can use to estimate the dependencies between input and output variables. The feature selection problem corresponds to selecting a subset of features such that the dependence between the input of the graph objects and the outputs (multiple labels) are maximized.

According to our *Dependence Maximization* assumption on the optimal subgraph features for multi-label graph classification, we can adopt the HSIC criterion to evaluate the dependence between the graph objects using a set of subgraph features and their multiple label outputs. Suppose we select a set of subgraph features $\mathcal{T}$, and each graph object G_i can be mapped into a feature space

$\mathcal{G}$ by $\phi(G_i) = D_{\mathcal{T}}\boldsymbol{x}_i$ with the kernel function $k(G_i, G_j) = \langle \phi(G_i), \phi(G_j) \rangle = \langle D_{\mathcal{T}}\boldsymbol{x}_i, D_{\mathcal{T}}\boldsymbol{x}_j \rangle$. Here $D_{\mathcal{T}} = \mathrm{diag}(\boldsymbol{\delta}_{\mathcal{T}})$ is a diagonal matrix indicating which features are selected into feature set $\mathcal{T}$ from $\mathcal{S}$. And $\boldsymbol{\delta}_{\mathcal{T}} = [\delta^1_{\mathcal{T}}, \delta^2_{\mathcal{T}}, \cdots, \delta^m_{\mathcal{T}}]^\top \in \{0,1\}^m$ is an indicator vector, and $\delta^i_{\mathcal{T}} = 1$ iff $g_i \in \mathcal{T}$. Then the kernel matrix on the graph objects with subgraph features $\mathcal{T}$ is denoted as $\mathbf{K}_{\mathcal{T}}$. Suppose $\mathbf{L} = [L_{ij}]_{n\times n}$ is a kernel matrix based upon the multiple labels of each graph, and the kernel function is $l(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle \psi(\boldsymbol{y}_i), \psi(\boldsymbol{y}_j) \rangle$. In our current implementation, $l(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle$ is used as the default label kernel. Other kernels can also be directly used, which will be discussed in Section 6. Then we can evaluate the dependence between graph objects using feature set $\mathcal{T}$ and the multiple labels as follows:

$$\mathrm{HSIC} = \mathrm{tr}(\mathbf{K}_{\mathcal{T}}\mathbf{HLH})$$

The subgraph feature selection task corresponds to the selection of a subset of features in $\mathcal{S}$, such that the dependence between graph objects and their multiple labels are maximized.

In detail, we can rewrite the optimization problem in Eq. 1 as follows:

$$\begin{aligned} \arg\max_{\mathcal{T}\subseteq\mathcal{S}} \quad & \mathrm{tr}\ (\mathbf{K}_{\mathcal{T}}\mathbf{H}\ \mathbf{L}\ \mathbf{H}) \\ \text{s.t.} \quad & |\mathcal{T}| \leq t, \end{aligned} \tag{2}$$

The formula in Eq. 2 can be rewritten as follows:

$$\begin{aligned} & \mathrm{tr}\,(\mathbf{K}_{\mathcal{T}}\mathbf{HLH}) \\ =\ & \mathrm{tr}\left(X^\top {D_{\mathcal{T}}}^\top D_{\mathcal{T}} X\mathbf{HLH}\right) \\ =\ & \mathrm{tr}\left(D_{\mathcal{T}} X\mathbf{HLH}X^\top {D_{\mathcal{T}}}^\top\right) \\ =\ & \sum_{g_i\in\mathcal{T}} \left({\boldsymbol{f}_i}^\top \mathbf{HLH}\boldsymbol{f}_i\right) \\ =\ & \sum_{g_i\in\mathcal{T}} \left({\boldsymbol{f}_i}^\top \mathbf{M}\boldsymbol{f}_i\right) \end{aligned}$$

where $\mathbf{M} = \mathbf{HLH}$. By denoting function $h(g_i, \mathbf{M}) = {\boldsymbol{f}_i}^\top \mathbf{M}\boldsymbol{f}_i$, the optimization (2) can be written as

$$\begin{aligned} \max_{\mathcal{T}} \quad & \sum_{g_i\in\mathcal{T}} h(g_i, \mathbf{M}) \\ \text{s.t.} \quad & \mathcal{T} \subseteq \mathcal{S}, |\mathcal{T}| \leq t \end{aligned} \tag{3}$$

Definition 4.1 (gHSIC**).** Suppose we have a multi-labeled graph dataset $\mathcal{D} = \{(G_1, \boldsymbol{y}_1), \cdots, (G_n, \boldsymbol{y}_n)\}$. Let L be a kernel matrix defined on the multiple label vectors, and $\mathbf{M} = \mathbf{HLH}$. We define a quality criterion q called gHSIC, for a subgraph feature g as

$$q(g) = h(g, \mathbf{M}) = {\boldsymbol{f}_g}^\top \mathbf{M}\boldsymbol{f}_g$$

where $\boldsymbol{f}_g = [f_g^{(1)}, \cdots, f_g^{(n)}]^\top \in \{0,1\}^n$ is the indicator vector for subgraph feature g, $f_g^{(i)} = 1$ iff $g \subseteq G_i$ $(i = 1, 2, \cdots, n)$. Since matrix $\mathbf{L}$ and $\mathbf{M}$ are positive semi-definite, for any subgraph pattern g, we have $q(g) \geq 0$.

The optimal solution to the problem in Eq. 2 can be found by using gHSIC to forward feature selection on a set of subgraphs $\mathcal{S}$. Suppose the gHSIC values for all subgraphs are denoted as $q(g_1) \geq q(g_2) \geq \cdots \geq q(g_m)$ in sorted order. Then the optimal solution to the optimization problem in Eq. 3 is:

$$\mathcal{T}^* = \{g_i | i \leq t\}.$$

5. The Proposed Solution

Now we address the second problem discussed in Section 3, and propose an efficient method to find the optimal set of subgraph features from a given multi-label graph dataset.

Exhaustive enumeration: One of the most simple and straightforward solution for finding an optimal feature set is the exhaustive enumeration, *i.e.*, we first enumerate all subgraph patterns in a multi-label graph dataset, and then calculate the gHSIC values for all subgraph patterns. However, in the context of graph classification, the number of subgraphs grows exponentially with the size of graphs, which makes the exhaustive enumeration approach usually impractical in real-world data.

Inspired by recent advances in graph classification approaches, *e.g.* (Yan et al., 2008; Kong and Yu, 2010), which put their evaluation criteria into the subgraph pattern mining steps and develop constraints to prune search spaces, we take a similar approach by deriving a different constraint for multi-label cases. In order to avoid the exhaustive search, we proposed a branch-and-bound algorithm, named gMLC, which is summarized as follows: a) Adopt a canonical search space where all the subgraph patterns can be enumerated. b) Search through the space, and find the optimal subgraph features by gHSIC. c) Propose an upper bound of gHSIC and prune the search space.

5.1. Subgraph Enumeration

In order to enumerate all subgraphs from a graph dataset, we adopted an efficient algorithm, gSpan, proposed by Yan et al(Yan and Han, 2002). We briefly review the general idea of gSpan approach: Instead of enumerating subgraphs and testing for isomorphism, they first build a lexicographic order over all the edges of a graph, and then map each graph to an unique minimum DFS code as its canonical label. The minimum DFS codes of two graphs are equivalent iff they are isomorphic. Details can be found in (Yan and Han, 2002). Based on this lexicographic order, a depth-first search (DFS) strategy is used to efficiently search through all the subgraphs in a DFS code tree. By a depth-first search through the DFS code tree's nodes, we can enumerate all the subgraphs of a graph in their DFS code's order. And the nodes with non-minimum DFS codes can be directly pruned in the tree, which saves us from performing an explicit isomorphic test among the subgraphs.

5.2. Upper Bound of gHSIC

Now, we can efficiently enumerate all the subgraph patterns of a graph dataset in a canonical search space using gSpan's DFS Code Tree. Then, we derive an upper bound for the gHSIC value which can be used to prune the search space as follows:

Theorem 5.1 (Upper bound of gHSIC). Given any two subgraphs $g, g' \in \mathcal{S}$, g' is a supergraph of g ($g' \supseteq g$). The gHSIC value of g' ($q(g')$) is bounded by $\hat{q}(g)$ (*i.e.*, $q(g') \leq \hat{q}(g)$), where $\hat{q}(g)$ is defined as follows:

$$\hat{q}(g) = \boldsymbol{f}_g{}^{\top}\hat{\mathbf{M}}\boldsymbol{f}_g \tag{4}$$

where the matrix $\hat{\mathbf{M}} = [\hat{M}_{ij}]_{n\times n}$ is defined as $\hat{M}_{ij} = \max\ (\ 0,\ M_{ij}\ )$. $\boldsymbol{f}_g = \{I(g \subseteq G_i)\}_{i=1}^{n} \in \{0,1\}^n$ is a vector indicating which graphs in a graph dataset $\{G_1, \cdots, G_n\}$ contain the subgraph g, $I(\cdot)$ is the indicator function. Suppose the gHSIC value of g is $q(g) = \boldsymbol{f}_g{}^{\top}\mathbf{M}\boldsymbol{f}_g$.

Proof.

$$q\left(g'\right) = \boldsymbol{f}_{g'}{}^{\top}\mathbf{M}\boldsymbol{f}_{g'} = \sum_{i,j:G_i,G_j\in\mathcal{G}(g')} M_{ij}$$

where $\mathcal{G}(g') = \{G_i | g' \subseteq G_i, 1 \leq i \leq n\}$. Since g' is the supergraph of g ($g' \supseteq g$), according to anti-monotonic property, we have $\mathcal{G}(g') \subseteq \mathcal{G}(g)$. Also $\hat{M}_{ij} = \max(0, M_{ij})$, we have $\hat{M}_{ij} \geq M_{ij}$ and $\hat{M}_{ij} \geq 0$. So,

$$\begin{aligned} q\left(g'\right) &= \sum_{i,j:G_i,G_j\in\mathcal{G}(g')} M_{ij} \\ &\leq \sum_{i,j:G_i,G_j\in\mathcal{G}(g')} \hat{M}_{ij} \\ &\leq \sum_{i,j:G_i,G_j\in\mathcal{G}(g)} \hat{M}_{ij} = \hat{q}\left(g\right) \end{aligned}$$

Thus, for any $g' \supseteq g$, $q(g') \leq \hat{q}(g)$. □

5.3. Subgraph Search Space Pruning

In this subsection, we make use of the the upper bound of gHSIC to efficiently prune the DFS Code Tree using a *branch-and-bound* method, which is similar to (Kong and Yu, 2010) but under different problem context: In depth-first search through the DFS Code Tree, we maintain the temporally suboptimal gHSIC value (denoted by θ) among all the gHSIC values calculated before. If $\hat{q}(g) < \theta$, the gHSIC value of any supergraph g' ($g' \supseteq g$) is no greater than θ. Now, we can safely prune the subtree from g in the search space. If $\hat{q}(g) \geq \theta$, we can not prune this space since there might exist a supergraph $g' \supseteq g$ ($q(g') \geq \theta$).

Figure 3 shows the algorithm gMLC. We first initialize the subgraphs $\mathcal{T}$ as an empty set. Then we prune the search space by running gSpan, while always maintaining the top-t best subgraphs according to q. In the course of mining, whenever we search to a subgraph g with $\hat{q}(g) \leq \min_{g_i\in\mathcal{T}} q(g_i)$, such that for

```
$\mathcal{T}$ = gMLC($\mathcal{D}$, $min\_sup$, $t$)

Input:
    $\mathcal{D}$ : Multi-label graphs $\{(G_1, \boldsymbol{y}_1), \cdots, (G_n, \boldsymbol{y}_n)\}$
    $min\_sup$ : Minimum support threshold
    $t$ : Maximum number of subgraph feature selected
Process:
    $\mathcal{T} = \emptyset$, $\theta = 0$;
    Recursively visit the DFS Code Tree in gSpan:
        $g$ = currently visited subgraph in DFS Code Tree
        if $|\mathcal{T}| < t$, then
            $\mathcal{T} = \mathcal{T} \cup \{g\}$;
        else if $q(g) > \min_{g' \in \mathcal{T}} q(g')$, then
            $g_{min} = \mathrm{argmin}_{g' \in \mathcal{T}} q(g')$ and $\mathcal{T} = \mathcal{T} / g_{min}$;
            $\mathcal{T} = \mathcal{T} \cup \{g\}$ and $\theta = \min_{g' \in \mathcal{T}} q(g')$;
        if $\hat{q}(g) > \theta$ and $freq(g) \geq min\_sup$, then
            Depth-first search subtree rooted from node $g$;
    return $\mathcal{T}$;
Output:
    $\mathcal{T}$ :   Set of optimal subgraph features
```

Fig. 3. The gMLC algorithm

any supergraph $g' \supseteq g$ $(q(g') \leq \hat{q}(g))$ according to the bound defined in Eq. (4), we can prune the branches of the search tree originating from g . In the other hand, as long as the resulting subgraph g can still improve the gHSIC value of any subgraph $g_i \in \mathcal{T}$, it is accepted into $\mathcal{T}$ and the last best subgraph is dropped off from $\mathcal{T}$.

Note that in our experiments with the three datasets, the gHSIC criterion based on multiple labels provides such a bound that we can even omit the support threshold min_sup and still find a set of optimal subgraphs within a reasonable time cost.

6. Exploiting Label Correlations

Now we address the third problem discussed in Section 3, and explain how label correlations can be considered in gMLC framework by adopting more informative and advanced kernels.

In the previous sections, we used the simple kernel function, $l(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle$, to generate the label kernel matrix $\mathbf{L}$. The linear kernel treats each label as being independent without considering the correlations among different labels. However in many real world applications, the multiple labels of the graphs are usually correlated. For example, in anti-cancer drug activity prediction tasks, chemical compounds which are active to one type of cancer are more likely to be active to some other related cancers. Subgraph patterns that corresponds to such label co-occurrences can be very useful for multi-label graph classification. In order to put label correlations into consideration during feature mining process, we need to adopt more informative kernels for $\mathbf{L}$ than linear kernel.

One simple solution is that the label correlations can be exploited by adopting more advanced kernels like polynomial or RBF kernels in the label kernel calculation. *i.e.*, the label vector $\boldsymbol{y}$ is mapped to a new feature space using $\psi(\boldsymbol{y})$ with kernel function $l(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle \psi(\boldsymbol{y}_i), \psi(\boldsymbol{y}_j) \rangle$, and the correlations among different labels are explicitly considered in the new feature space.

For example, suppose we use a polynomial kernel with degree 2, $l(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle^2$, as the label kernel function. Given any two label vectors $\boldsymbol{\alpha} = [\alpha_1, \alpha_2] \in \{0,1\}^2$ and $\boldsymbol{\beta} = [\beta_1, \beta_2] \in \{0,1\}^2$, we have

$$\begin{aligned} l(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle^2 \\ &= (\alpha_1\beta_1 + \alpha_2\beta_2)^2 \\ &= \left\langle \left[\alpha_1{}^2, \alpha_2{}^2, \sqrt{2}\alpha_1\alpha_2\right]\ ,\ \left[\beta_1{}^2, \beta_2{}^2, \sqrt{2}\beta_1\beta_2\right] \right\rangle \\ &= \langle \psi(\boldsymbol{\alpha}),\ \psi(\boldsymbol{\beta}) \rangle \end{aligned}$$

Here, $\psi(\boldsymbol{\alpha}) = \left[\alpha_1{}^2, \alpha_2{}^2, \sqrt{2}\alpha_1\alpha_2\right]$, and the component $(\sqrt{2}\alpha_1\alpha_2)$ considers the correlations between label l_1 and l_2 explicitly. Intuitively, by adopting polynomial kernels with degree 2, the *second-order* correlations among different labels can be exploited in our gMLC framework. Higher orders of correlations among labels can also be exploited by adopting polynomial kernels with higher degrees or even RBF kernels to construct the label kernel $\mathbf{L}$.

After using these kernel functions, the new label kernel matrix $\mathbf{L}$ can be directly plugged in the subgraph evaluation criterion, $q(g) = \boldsymbol{f}_g{}^\top \mathbf{HLH}\boldsymbol{f}_g$. Subgraph patterns that best corresponds to the co-occurrence of different labels will get high values, thus being selected into the optimal feature set for multi-label graph classification.

7. Experiments

7.1. Experimental Setup

7.1.1. Data Collections

In order to evaluate the multi-label graph classification performances, we tested our algorithm on three real-world multi-label graph classification tasks as follows: (Summarized in Table 1.)

1) Anti-cancer activity prediction (NCI1): The first task is to classify chemical compounds' anti-cancer activities on multiple types of cancer. We build up a multi-label graph dataset using a benchmark dataset, NCI[1] (Yan et al., 2008), which consists of records of chemical compounds' anti-cancer activities against a set of 10 types of cancer (*e.g.* Leukemia, Prostate, Breast), and each chemical compound is represented as a graph. After removing compounds with incomplete records for 10 types of cancer, we thus have a multi-label graph

[1] http://pubchem.ncbi.nlm.nih.gov

Table 1. Summary of experimental tasks studied. "AvgL" denotes the average number of labels assigned to each graph.

Prediction Task	Dataset	# Graphs	# Labels	AvgL
Anti-cancer	NCI1	812	10	4.36
Toxicology	PTC	253	4	1.60
Kinase Inhibition	NCI2	5,660	4	1.04

Table 2. Details of the anti-cancer activity prediction task (NCI1 dataset). Each label represents the assay result for one type of cancer. "Pos (%)" denotes the average percentage of positive instances for each cancer assay.

Assay ID	Class Name	Pos (%)	Cancer Type
1	NCI-H23	35.6	Non-Small Cell Lung
33	UACC-257	47.7	Melanoma
41	PC-3	38.5	Prostate
47	SF-295	34.1	Central Nerve System
81	SW-620	17.5	Colon
83	MCF-7	59.2	Breast
109	OVCAR-8	42.2	Ovarian
123	MOLT-4	73.5	Leukemia
145	SN12C	54.8	Renal
330	P388	33.4	Leukemia

Table 3. Details of toxicology prediction task (PTC dataset), where each of the multiple labels represents the toxicology test result on one type of animal. "Pos (%)" denotes the average percentage of positive instances for each cancer assay.

Class Name	Pos (%)	Animal Model
MR	41.9	Male Rats
FR	36.0	Female Rats
MM	38.7	Male Mice
FM	43.1	Female Mice

Table 4. Details of kinase inhibition prediction task (NCI2 dataset), where each of the multiple labels represents the inhibition of one type of kinase. "Pos (%)" denotes the average percentage of positive instances for each cancer assay.

Assay ID	Pos (%)	Kinase Type
1416	6.11	PERK
1446	40.5	JAK2
1481	15.9	ATPase
1531	41.4	MEK

classification dataset with 812 graphs assigned with 10 candidate labels. Table 2 provides a brief description of the 10 types of cancer in NCI1 dataset.

2) Toxicology prediction of chemical compounds (PTC): The second task is to classify chemical compounds' carcinogenicity on multiple animal models. We build up our second multi-label graph dataset using a benchmark dataset, PTC[2] (Helma et al., 2001), which consists carcinogenicity records of 417 chemical compounds on 4 animal models: MM (Male Mouse), FM (Female Mouse), MR (Male Rat) and FR (Female Rat). Each chemical compound is assigned with carcinogenicity labels for the 4 animal models. On each animal model the carcinogenicity label is one of {CE, SE, P, E, EE, IS, NE, N}. We assume {CE, SE, P} as 'positive' labels, {NE, N} as 'negative' and { E, EE IS} labels are removed, which is the same setting as (Kashima et al., 2003; Kudo et al., 2005). Each chemical compound is represented as a graph with an average of 25.7 vertices. After removing compounds with incomplete records for the 4 animal models, we thus have a multi-label graph classification dataset with 253 graphs assigned with four candidate labels (MR, FR, MM, FM). Table 3 provides a brief description of the 4 animal models in PTC dataset.
3) Kinase inhibition prediction of chemical compounds (NCI2): The third task is to classify the ability of chemical compounds to inhibit multiple kinases' activity, which is a important problem in finding effective inhibitors for kinase associated diseases (*e.g.* infectious diseases, cancers). We build up our third multi-label graph dataset also from NCI database, which consists kinase inhibition records of 5,660 chemical compounds against a set of 4 types of kinases (*i.e.* ATPase, PERK, MEK, JAK2). After removing compounds with incomplete records for the 4 types of kinases, we thus have a multi-label graph classification dataset with 5,660 graphs assigned with 4 candidate labels. Table 4 provides a brief description of the 4 types of kinases in NCI2 dataset.

7.1.2. Evaluation Metrics

Multi-label classification requires different evaluation metrics than conventional single-label classification problems. Here we adopt some metrics used in (Schapire and Singer, 2000; Elisseeff and Weston, 2002; Zhang and Zhou, 2007) to evaluate the multi-label graph classification performance. Assume we have a multi-label graph dataset $\mathcal{D} = \{(G_1, \boldsymbol{y}_1), \cdots, (G_n, \boldsymbol{y}_n)\}$, where graph G_i is labeled as $\boldsymbol{y}_i \in \{0,1\}^Q$. Let $f(G_i, k)$ denote the classifier's real-value outputs for G_i on the k-th label (l_k), and $h(G_i) \in \{0,1\}^Q$ denotes the classifier's binary output label vector. According to $f(G_i, k)$ we can define a ranking function $rank_f(G_i, k) \in \{1, 2, \cdots, Q\}$, and $rank_f(G_i, k') < rank_f(G_i, k)$ iff $f(G_i, k') < f(G_i, k)$. We have the following evaluation criteria:

- Ranking Loss (Elisseeff and Weston, 2002): evaluates the performance of classifier's real-value outputs $f(G_i, k)$. It is calculated as the average fraction of incorrectly ordered label pairs:

$$RankLoss = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\mathbf{1}^\top \boldsymbol{y}_i \mathbf{1}^\top \overline{\boldsymbol{y}}_i} Loss_f(G_i, \boldsymbol{y}_i)$$

[2] http://www.predictive-toxicology.org/ptc/

Where the $\overline{\boldsymbol{y}}_i$ denotes the complementary of $\boldsymbol{y}_i$ in $\{0,1\}^Q$.

$$Loss_f(G_i, \boldsymbol{y}_i) = \sum_{k:y_i^k=1} \sum_{k':y_i^{k'}=0} [\![f(G_i,k) \leq f(G_i,k')]\!]$$

For any predicate π, $[\![\pi]\!]$ equals 1 if π holds and 0 otherwise. $RankLoss \in [0,1]$. The smaller the value, the better the performance.

- Average Precision (Zhang and Zhou, 2007): evaluates the average fraction of labels ranked above a particular label y s.t. y is in the ground-truth label set. This criterion is originally used in information retrieval (IR) systems to evaluate the document ranking performance for query retrieval:

$$AvgPrec = \frac{1}{n}\sum\nolimits_{i=1}^{n} \frac{1}{\mathbf{1}^\top \boldsymbol{y}_i} \sum_{k:y_i^k=1} \frac{\mathrm{Prec}_f(G_i,k)}{rank_f(G_i,k)}$$

  which measure the number of assigned class labels that are ranked before k-th class. Here

$$\mathrm{Prec}_f(G_i,k) = \sum_{k':y_i^{k'}=1} [\![rank_f(G_i,k') \leq rank_f(G_i,k)]\!]$$

  And $AvgPrec \in [0,1]$, the larger the value, the better the performance.

- *One error*: evaluates how many times the top-ranked label is not in the set of ground-truth labels of the instance.

$$OneError = \frac{1}{n}\sum\nolimits_{i=1}^{n} [\![y_i^{k_i} = 0]\!]$$

  where $k_i = \mathrm{argmax}_{k\in[1,Q]}\, f(G_i,k)$. $OneError \in [0,1]$, the smaller the value, the better the performance.

- *Coverage*: evaluates the performance by considering how far, on average, we need to go down the ranked label list to cover all the ground-truth labels of the instance.

$$Coverage = \frac{1}{n}\sum\nolimits_{i=1}^{n} \max_{k:y_i^k=1} rank_f(G_i,k) - 1$$

  where $Coverage \in [0, Q-1]$. The smaller the coverage, the better the performance.

- *Hamming loss*: evaluates how many times an instance-label pair is misclassified. For single-label problems, it equals the classification error.

$$HammingLoss = \frac{1}{n}\sum\nolimits_{i=1}^{n} \theta(h(G_i), \boldsymbol{y}_i)$$

  where

$$\theta(h(G_i), \boldsymbol{y}_i) = \frac{1}{Q}\sum\nolimits_{k=1}^{Q} [\![y_i^k \neq h(G_i)^k]\!]$$

  and $HammingLoss \in [0,1]$, the smaller the value, the better the performance.

In our experiment, we will show the value of $1 - AvePrec$ instead of *Average Precision*. Thus under all these evaluation criteria, smaller values are all indicating better performances. Note that all the criteria evaluate the performance of multi-label classification systems from different aspects. Usually few algorithms could outperform another algorithm on all those criteria. All experiments are conducted on machines with 4 GB RAM and Intel Xeon$^{\text{TM}}$Quad-Core CPUs of 2.40 GHz.

7.1.3. Comparing Methods

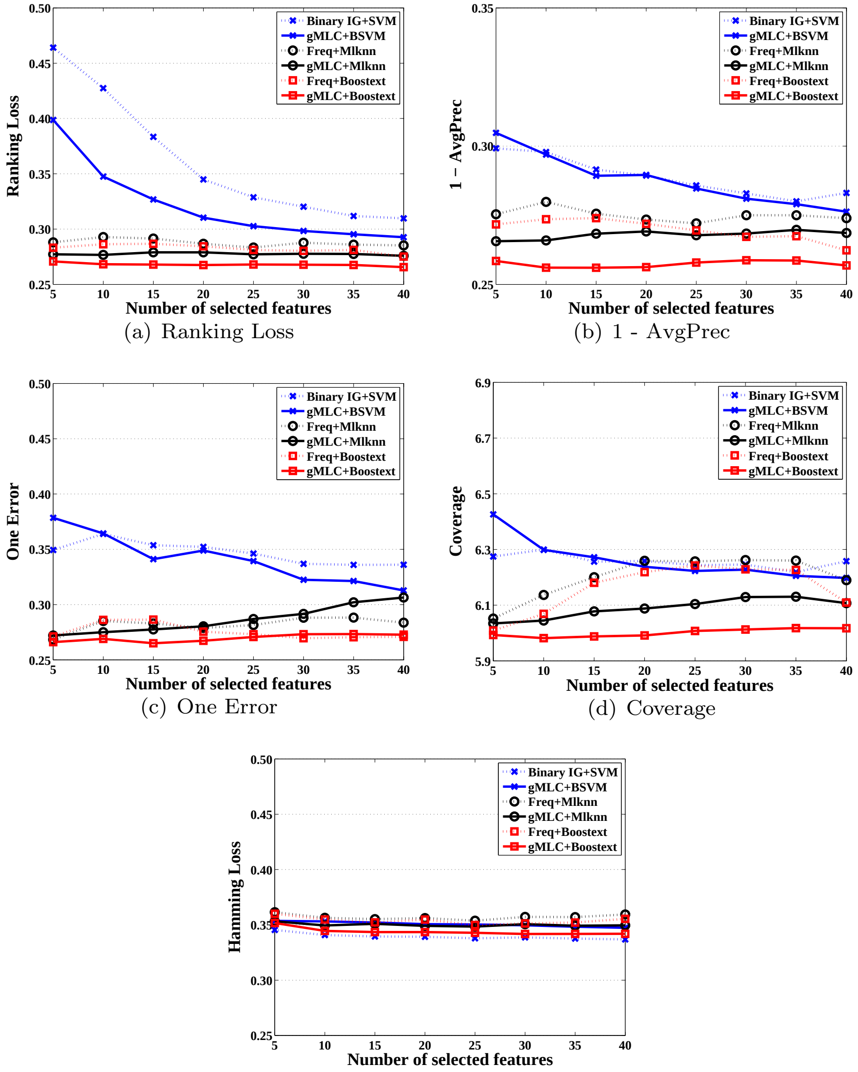
In order to demonstrate the effectiveness of our multi-label graph feature selection approach, we test with following methods:

- Binary decomposition + single-label feature selection + binary classifications (Binary IG+ SVM): We first compare with a baseline using a binary decomposition method similar to (Boutell et al., 2004): The multi-label graph dataset is first divided into multiple single-label graph datasets by one-vs-all binary decomposition. For each binary classification task, we use the Information Gain (IG), an entropy based measure, to select a subset of discriminative features from frequent subgraphs. Then SVMs are used as the binary classification models to classify the graphs into multiple binary classes respectively. We use SVM-light software package[3] to train the SVMs, where the parameters are set as default settings.
- Multi-label feature selection (gMLC) + binary classifications (SVM): gMLC is used to find a set of optimal subgraph features. Then the one-vs-all deduction with one SVM trained for each class is used as the multi-label classifier.
- Top-k frequent subgraph features (Freq) + multi-label classification (BoosTexter): We also compare with another baseline: multi-label classification using the top-k frequent subgraphs as features, *i.e.*, we use the top-k frequent subgraph features in the graph dataset without the gHSIC selections on the subgraph features. Then BoosTexter(Schapire and Singer, 2000) is used as the multi-label classifier. The number of boosting rounds for BoosTexter is set as 500, which does not significantly affect the classification performance.
- Multi-label feature selection (gMLC) + multi-label classification (BoosTexter): gMLC is used to find a set of optimal subgraph features. Then BoosTexter is used as the multi-label classifier.
- Top-k frequent features (Freq) + multi-label classification (Ml-knn): multi-label classification using the top-k frequent subgraphs as features. Ml-knn (Zhang and Zhou, 2007) is used as the multi-label classifier. The number of neighbors is set as the default value 10.
- Multi-label feature selection (gMLC) + multi-label classification (Ml-knn): We first use gMLC to find a set of optimal subgraph features. Then Ml-knn is used as the multi-label classifier.

7.2. Performances on Multi-label Graph Classification

In our experiment, we use 10-round 10-fold cross validation to evaluate the multi-label graph classification performance. Each graph dataset is evenly partitioned into 10 parts. Only one part is used as testing graphs and the other nine are used as training graphs for frequent subgraph mining, feature selection and multi-label classification. We repeat the 10-fold cross validation 10 times and we report the

[3] http://svmlight.joachims.org/



(a) Ranking Loss (b) 1 - AvgPrec

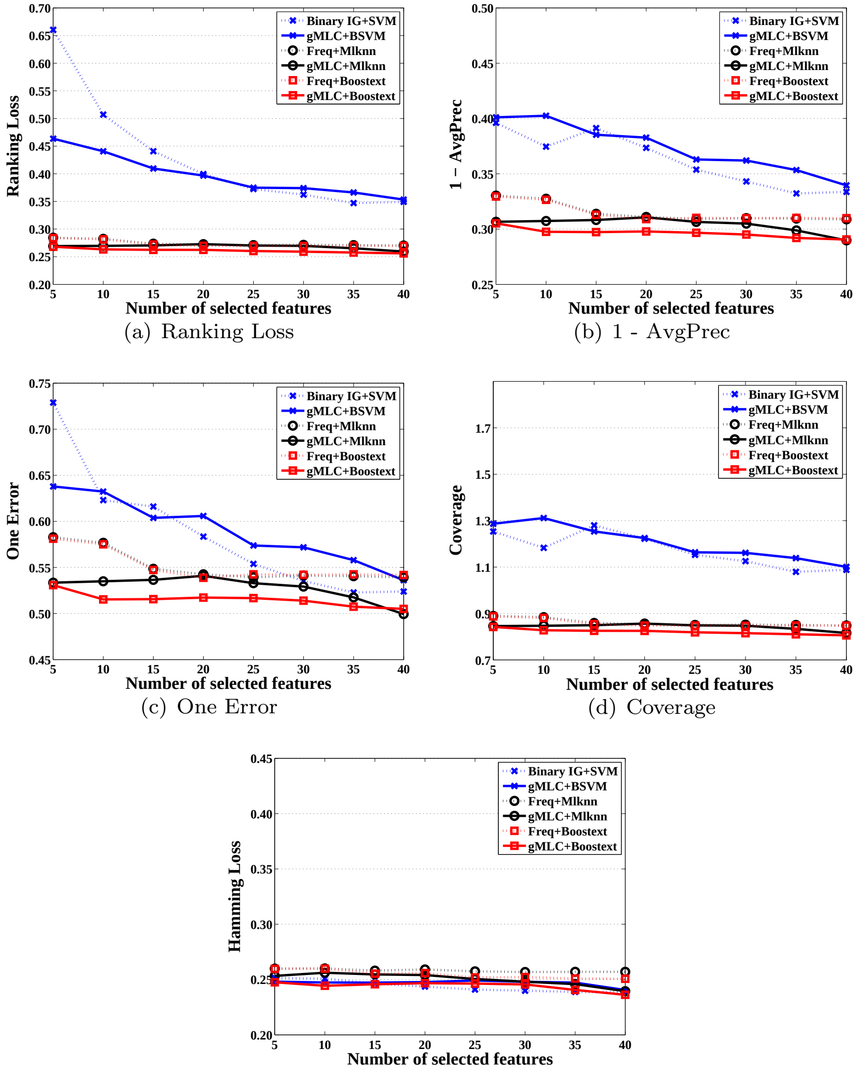
(c) One Error (d) Coverage

(e) Hamming Loss

Fig. 4. Multi-label graph classification performances on Anti-cancer Activity Prediction (NCI1 dataset)

average results for the 10 rounds. The result of the feature selection methods for multi-label graph classification on NCI1, NCI2 and PTC datasets are displayed in Figure 4, Figure 5 and Figure 6. We show the number of selected subgraphs t among frequent subgraphs using $min\_sup = 10\%$, together with evaluation metrics mentioned before.

Now, we first study the effectiveness of selecting subgraph features by comparing two approaches: gMLC+SVM, Binary IG+ SVM, where the binary SVMs are used as base learners. It is worth noticing that, this comparison is only used



(a) Ranking Loss

(b) 1 - AvgPrec

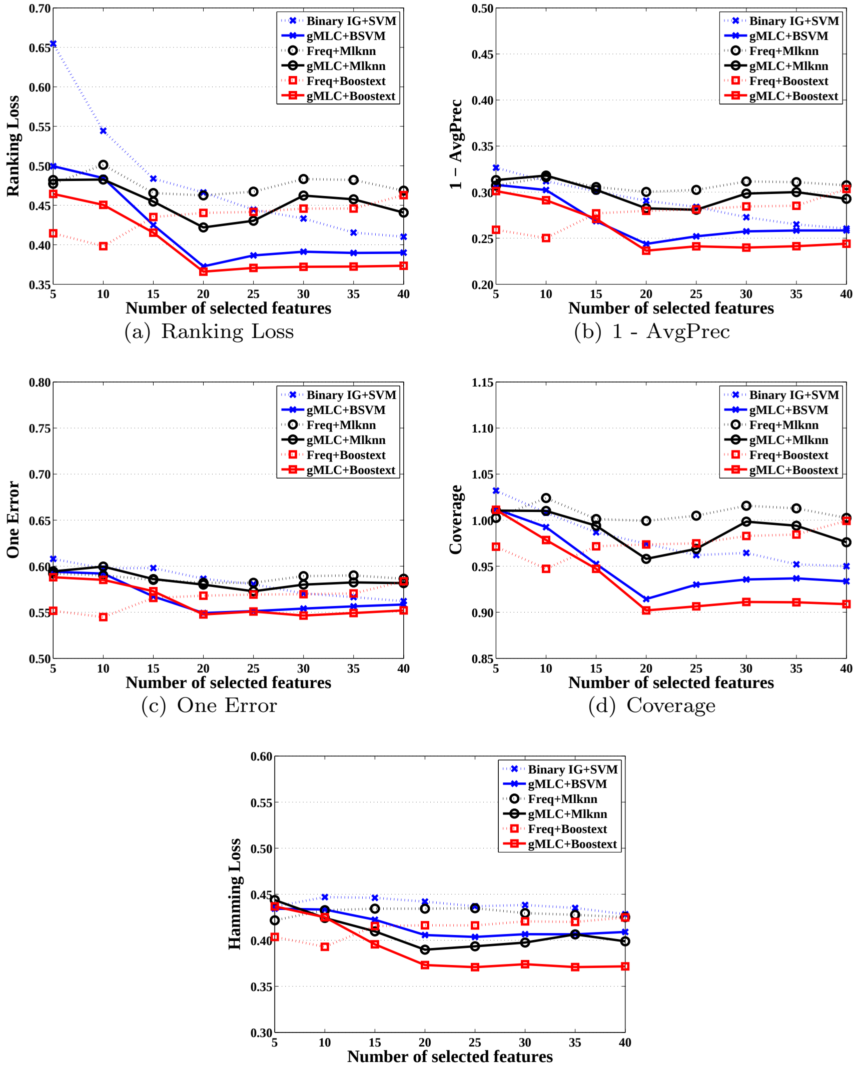
(c) One Error

(d) Coverage

(e) Hamming Loss

Fig. 5. Multi-label graph classification performances on Kinase Inhibition Prediction (NCI2 dataset)

for reference, since different number of features are used in the two methods. Our gMLC is designed for conventional multi-label classification methods, thus in the baseline gMLC+SVM, we select one set of subgraph features which is used on multiple SVMs separately. However, Binary IG+ SVM selects a different set of subgraph features for each label concept and these feature sets are used on multiple SVMs separately. Hence, Binary IG+ SVM method has an advantage over our method by using different feature sets for different SVMs, while gMLC uses the same set of feature for all the SVMs. Figure 4, Figure 5 and Figure 6



(a) Ranking Loss

(b) 1 - AvgPrec

(c) One Error

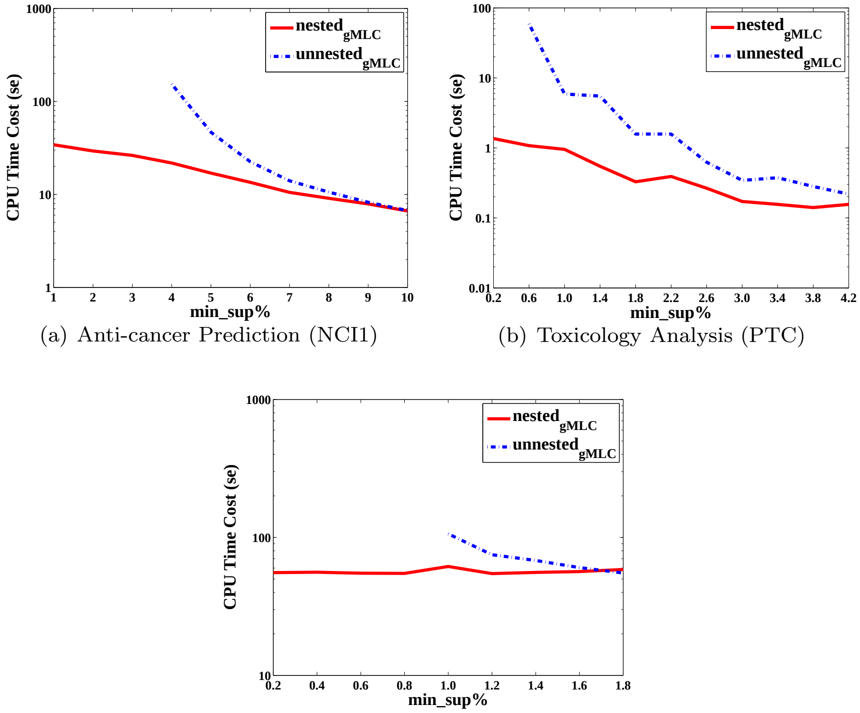
(d) Coverage

(e) Hamming Loss

Fig. 6. Multi-label graph classification performances on Toxicology Prediction Task (PTC dataset)

indicate that gMLC+SVM can achieve comparable or even better performances than Binary IG+ SVM in most cases. This is because the multiple labels of the graphs usually have certain correlations, and the useful subgraph features on one label concept are also likely to be useful on some other label concepts. Thus our gMLC method can achieve better performances over Binary IG+ SVM even though we use a same set of feature for all binary SVMs. Utilizing the potential relations among multiple label concepts to select subgraph features are crucial to the success of our method in this case.

We further study the effectiveness of subgraph features using the general



(a) Anti-cancer Prediction (NCI1)

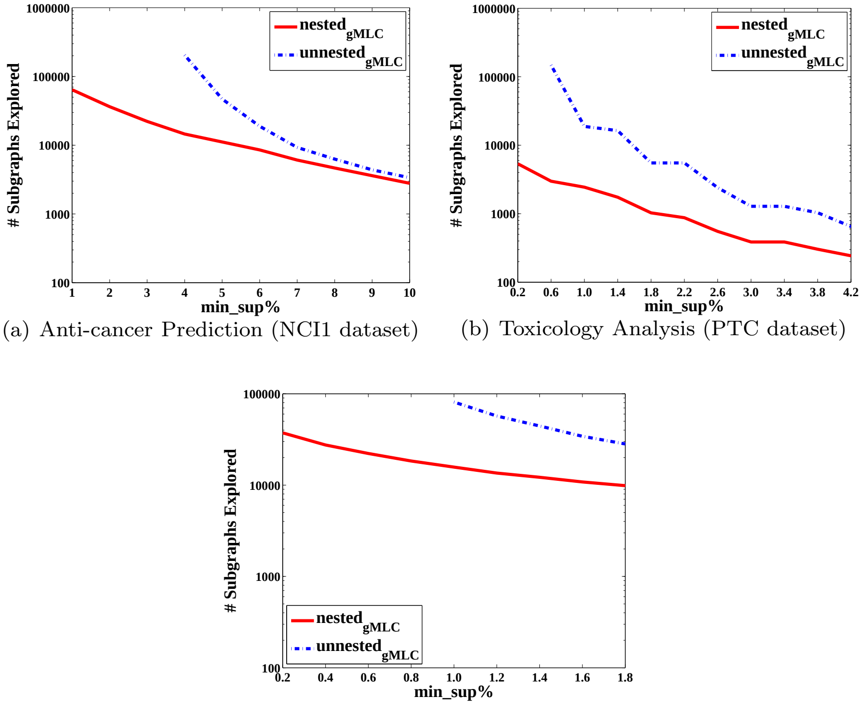
(b) Toxicology Analysis (PTC)

(c) Kinase inhibition (NCI2)

Fig. 7. Average CPU time for nested gMLC versus un-nested gMLC with varying min_sup.

purposed multi-label classification methods, *i.e.* BoosTexter and Ml-knn, as the base classifiers. It is also worth noticing that, to the best of our knowledge, gMLC is the first multi-label feature selection method for graph data. Thus we cannot find any other baseline which select one set of features for multiple label concepts in order to make a fair comparison. So our only choices are comparing the following methods: gMLC+BoosTexter v.s. Freq+BoosTexter and gMLC+Ml-knn v.s. Freq+Ml-knn. We observe that on most tasks the performances of gMLC+BoosTexter are better than Freq+BoosTexter, *i.e.* multi-label classification approaches without gHSIC subgraph feature selection. Similar results can also be found with the cases when Ml-knn is used as the base classifier. These results support our intuition that the gHSIC evaluation criterion in gMLC can find better subgraph patterns for multi-label graph classification than unsupervised top-k frequent subgraph approaches. The exception is only the case on PTC dataset when the number of features selected is small (less than 15). Nonetheless, the Freq+BoosTexter can never reach the best performance achievable by gMLC with a larger number of features. This is because the top 15 frequent features happen to be good classification features. However, the Freq cannot find other good features that are not that frequent.

Now, we first study the effectiveness of selecting subgraph features by comparing two approaches: gMLC+SVM, Binary IG+ SVM, where the binary SVMs are used as base learners. It is worth noticing that, our gMLC is specially designed for conventional multi-label classification methods which require one set



(a) Anti-cancer Prediction (NCI1 dataset)

(b) Toxicology Analysis (PTC dataset)

(c) Kinase inhibition (NCI2 dataset)

Fig. 8. Average number subgraph patterns explored during mining for nested gMLC versus un-nested gMLC with varying min_sup.

of features for all labels concepts. Thus gMLC only selects one set of subgraph features and uses it on multiple SVMs separately. However, Binary IG+ SVM selects a different set of subgraph features for each label concept and these feature sets are used on multiple SVMs separately. Hence, Binary IG+ SVM method has an advantage over our method by using different feature sets for different SVMs, while gMLC uses the same set of feature for all the SVMs. Figure 4, Figure 5 and Figure 6 indicate that gMLC+SVM can achieve compariable or even better performances than Binary IG+ SVM in most cases. This is because the multiple labels of the graphs usually have certain correlations, and the useful subgraph features on one label concept are also likely to be useful on some other label concepts. Thus our gMLC method can achieve better performances over Binary IG+ SVM even though we use a same set of feature for all binary SVMs. Utilizing the potential relations among multiple label concepts to select subgraph features are crucial to the success of our method in this case.

We further observe that in all tasks and evaluation criteria, our multi-label feature selection algorithm with multi-label classification (gMLC+BoosTexter) outperforms the binary decomposition approach using single-label feature selections (Binary IG+ SVM). gMLC+BoosTexter can achieve good performances with only a small number of features. We note that the big improvement can both be counted on the good performance of gMLC feature selection and the state-of-the-art multi-label classification method, BoosTexter. However, this result can just be used for a reference to the relative performances of the two types of multi-label graph classification methods, binary decomposition based and gMLC based. These results support the importance of the proposed multi-label feature selection method in the multi-label graph classification problems.

Additionally, by comparing over different evaluation criteria, we can find that gMLC shows more improvements over other baselines on criteria, *e.g.* Ranking Loss, which are most related to multi-label performances, than Hamming Loss. For Hamming Loss, gMLC gets better performances over other baselines on PTC dataset, but comparible performances on NCI1 and NCI2 dataset. This can be explained that Hamming Loss evaluates the classification performance in a binary way, simply averaging the binary classification error on each label without considering the ranking of all labels which is more important for multi-label classification evaluation.

7.3. Effectiveness of Subgraph Search Space Pruning

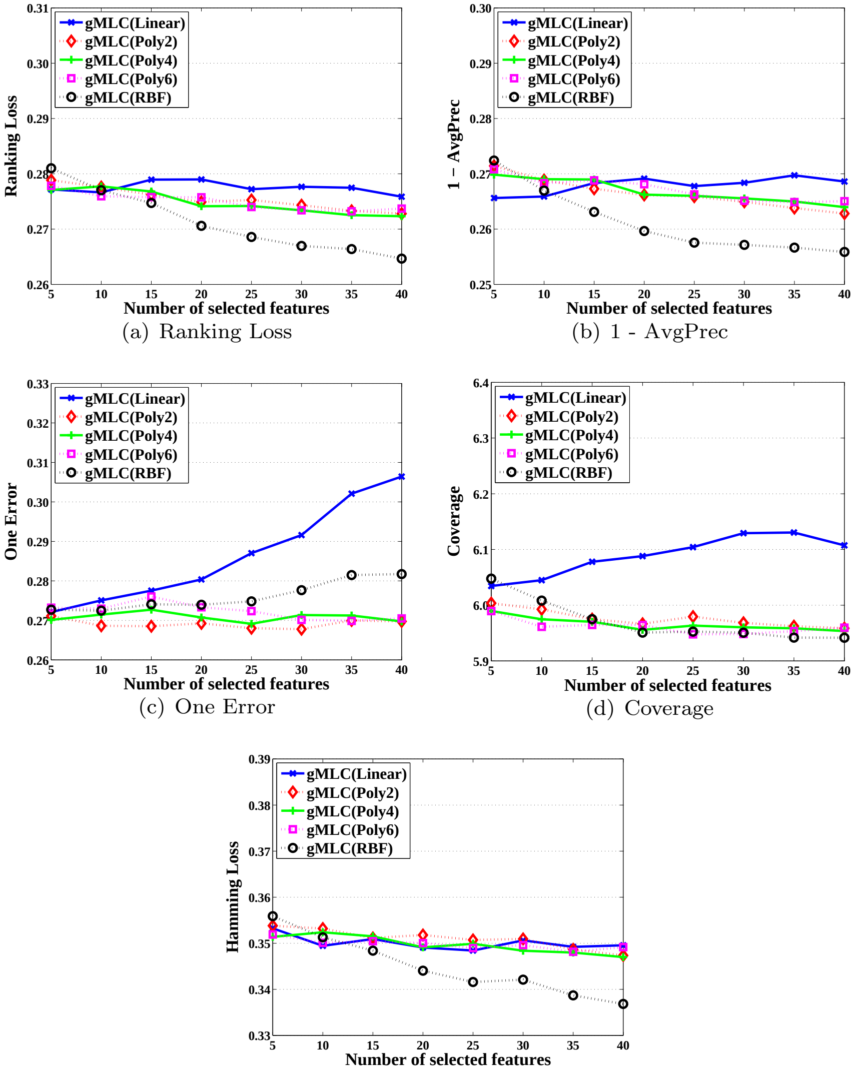
In our second experiment, we evaluated the effectiveness of the upper-bound for gHSIC proposed in Section 5.2. So, in this section we compare the runtime performance of two versions of implementation for gMLC: "nested gMLC" versus "un-nested gMLC". The "nested gMLC" denotes the proposed method using the upper-bound proposed in Section 5.2 to prune the search space of subgraph enumerations; the "un-nested gMLC" denotes the method without the gHSIC's upper-bound pruning, which first uses gSpan to find a set of frequent subgraphs, and then selects the optimal set of subgraphs via gHSIC. We run both approaches on the three tasks and record the average CPU time used on feature mining and selection. The result is shown in Figure 7.

In the NCI1, NCI2 and PTC dataset, we observe that as we decrease the *min_sup* in the frequent subgraph mining, the un-nested gMLC would need to explore larger subgraph search spaces, and this size increases exponentially with the decrease of *min_sup*. In the NCI1 dataset, when the *min_sup* get too low ($min\_sup < 4\%$), the subgraph feature enumeration step in un-nested gMLC can run out of the computer memory. However, the nested gMLC's running time does not increase as much, because the gHSIC can help pruning the subgraph search space using the multi-label information of the graphs. As we can see, the *min_sup* can go to very low value in all datasets for the "nested gMLC".

Figure 8 shows the number of subgraph feature explored in the process of subgraph pattern enumeration in the three tasks. In all tasks, we observe that the number of searched subgraph patterns in nested gMLC is much smaller than that of un-nested gMLC (the gSpan step). In our experiments, we further noticed that on most datasets, nested gMLC provides such a strong bound that we may even allow nested gMLC to omit the minimum support threshold *min_sup* and still receive an optimal set of subgraph features within a reasonable time.

7.4. Effectiveness of Embedding Label Correlations

In our third experiment, we evaluated the effectiveness of the label kernels after incorporating the label correlations in Section 6. In order to consider label correlations of first-order, second-order and higher-orders *etc.*, we use the following kernel functions to produce the label kernel matrix $\mathbf{L}$:



(a) Ranking Loss

(b) 1 - AvgPrec

(c) One Error

(d) Coverage

(e) Hamming Loss

Fig. 9. Performances of gMLC with/without considering label correlations on anti-cancer activity prediction task (NCI1 dataset)

- gMLC(Linear) denotes our gMLC method with linear kernels for **L**, which does not consider the label correlation. The kernel function is $l(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle$.
- gMLC(Poly) denotes the gMLC method with polynomial kernels with different degrees, which can consider label correlations of second-orders or even higher-orders. The kernel function is $l(\boldsymbol{y}_i, \boldsymbol{y}_j) = (\gamma \langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle + \eta)^d$. The γ is set as the default value $\gamma = \frac{1}{\#\text{features}}$, and $\eta = 0$. d denotes the degree of polynomial kernels. For example, gMLC(Poly2) corresponds to the polynomial kernel with degree two ($d = 2$).

- gMLC(RBF) denotes our gMLC method with RBF kernels for $\mathbf{L}$, which can consider label correlations of any orders. The kernel function is $l(\boldsymbol{y}_i, \boldsymbol{y}_j) = \exp(-\gamma|\boldsymbol{y}_i - \boldsymbol{y}_j|^2)$. The γ is set as the default value $\gamma = \frac{1}{\#\text{features}}$.

In all methods, Ml-knn is used as the base classifier, with default parameter settings ($k = 10$). The result of NCI1 dataset is illustrated in Figure 9. From the results, we can see that gMLC with polynomial kernel and RBF kernels can get better performances than gMLC with linear kernels, by considering label correlations in the label kernel matrix $\mathbf{L}$. Here we only use simple strategies to consider label relationship in our gMLC model, and greater improvements are likely to be obtain by defining more advanced kernels for label matrix $\mathbf{L}$.

8. Conclusion

In this paper, we study the problem multi-label feature selection for graph classification. It is significantly more challenging than the conventional single-label feature selection in graph data because of the multiple labels assigned to each graph. To address this challenge, we propose an evaluation criterion gHSIC to evaluate the dependence of subgraph features with the multiple labels of graphs, and derived an upper-bound for gHSIC to prune the subgraph search space. Then we propose a branch-and-bound algorithm to efficiently find a compact set of subgraph feature which is useful for the classification of graphs with multiple labels. Empirical studies on real-world tasks show that our feature selection method for multi-label graph classification, gMLC, can effectively boost multi-label graph classification performances and is more efficient by pruning the subgraph search space using multiple labels. Additionally, the correlations among different labels can be exploited effectively by adopting more informative and advanced kernels for label kernel matrix.

In our current implementation, we only use simple strategies to construct label kernel matrix. Actually various other types of label kernels can also be used to exploit the label correlations among multiple labels more effectively. We will leave related discussions to potential future works.

Acknowledgements. This work is supported in part by NSF through grants IIS 0905215, DBI-0960443, OISE-0968341 and OIA-0963278.

References

Borgelt, C. and Berthold, M. (2002). Mining molecular fragments: Finding relevant substructures of molecules, *Proceedings of the 2nd IEEE International Conference on Data Mining*, Maebashi City, Japan, pp. 211–218.

Borgwardt, K. M. (2007). *Graph Kernels*, PhD thesis, Ludwig-Maximilians-University Munich.

Boutell, M. R., J. Luo, Shen, X. and Brown, C. M. (2004). Learning multi-label scene classification, *Pattern Recognition* **37**(9): 1757–1771.

Chen, C., Yan, X., Zhu, F., Han, J. and Yu, P. (2009). Graph OLAP: a multi-dimensional framework for graph data analysis, *Knowledge and Information Systems* **21**(1): 41–63.

Comité, F. D., Gilleron, R. and Tommasi, M. (2003). Learning multi-label altenating decision tree from texts and data, *Proceedings of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition*, Leipzig, Germany, pp. 35–49.

Elisseeff, A. and Weston, J. (2002). A kernel method for multi-labelled classification, *Advances in Neural Information Processing Systems 14*, pp. 681–687.

Fei, H. and Huan, J. (2010). Boosting with structure information in the functional space: an application to graph classification, *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Washington, DC, pp. 643–652.

G. Tsoumakas, I. V. (2007). Random k-labelsets: An ensemble method for multilabel classification, *Proceedings of the 18th European Conference on Machine Learning*, Warsaw, Poland, pp. 406–417.

Godbole, S. and Sarawagi, S. (2004). Discriminative methods for multi-labeled classification, *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Sydney, Australia, pp. 22–30.

Gretton, A., Bousquet, O., Smola, A. and Schölkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms, *ALT*, Singapore, pp. 63–77.

Helma, C., King, R., Kramer, S. and Srinivasan, A. (2001). The predictive toxicology challenge 2000-2001, *Bioinformatics* **17**(1): 107–108.

Huan, J., Wang, W. and Prins, J. (2003). Efficient mining of frequent subgraph in the presence of isomorphism, *Proceedings of the 3rd IEEE International Conference on Data Mining*, Melbourne, FL, pp. 549–552.

Inokuchi, A., Washio, T. and Motoda, H. (2000). An apriori-based algorithm for mining frequent substructures from graph data, *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, Lyon, France, pp. 13–23.

Jia, Y., Tao, J. and Huan, J. (2011). An efficient graph-mining method for complicated and noisy data with real-world applications, *Knowledge and Information Systems* pp. 1–25.

Kashima, H., Tsuda, K. and Inokuchi, A. (2003). Marginalized kernels between labeled graphs, *Proceedings of the 20th International Conference on Machine Learning*, Washington, DC, pp. 321–328.

Kazawa, H., Izumitani, T., Taira, H. and Maeda, E. (2005). Maximal margin labeling for multi-topic text categorization, *Advances in Neural Information Processing Systems 15*, pp. 649–656.

Kong, X. and Yu, P. (2010). Semi-supervised feature selection for graph classification, *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Washington, DC, pp. 793–802.

Kudo, T., Maeda, E. and Matsumoto, Y. (2005). An application of boosting to graph classification, *Advances in Neural Information Processing Systems 15*, pp. 729–736.

Kuramochi, M. and Karypis, G. (2001). Frequent subgraph discovery, *Proceedings of the 1st IEEE International Conference on Data Mining*, San Jose, CA, pp. 313–320.

McCallum, A. (1999). Multi-label text classification with a mixture model trained by EM, *Working Notes of the AAAI'99 Workshop on Text Learning*, Orlando, FL.

Nijssen, S. and Kok, J. (2004). A quickstart in frequent structure mining can make a difference, *Proceedings of the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Seattle, WA, pp. 647–652.

Schapire, R. E. and Singer, Y. (2000). Boostexter: a boosting-based system for text categorization, *Machine Learning* **39**(2-3): 135–168.

Tasourakakis, U. K. C. and Faloutsos, C. (2010). Pegasus: mining peta-scale graphs, *Knowledge and Information Systems* pp. 1–23.

Thoma, M., Cheng, H., Gretton, A., Han, J., Kriegel, H., Smola, A., Song, L., Yu, P., Yan, X. and Borgwardt, K. (2009). Near-optimal supervised feature selection among frequent subgraphs, *Proceedings of the 9th SIAM International Conference on Data Mining*, Sparks, Nevada, pp. 1075–1086.

Ueda, N. and Saito, K. (2003). Parametric mixture models for multi-labeled text, *Advances in Neural Information Processing Systems 13*, pp. 721–728.

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

Yan, X. and Han, J. (2002). gSpan: Graph-based substructure pattern mining, *Proceedings of the 2nd IEEE International Conference on Data Mining*, Maebashi City, Japan, pp. 721–724.

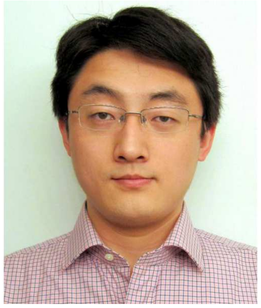
Ying, X. and Wu, X. (2010). On link privacy in randomizing social networks, *Knowledge and Information Systems* pp. 1–19.

Zhang, M.-L. and Zhou, Z.-H. (2007). Ml-knn: A lazy learning approach to multi-label learning, *Pattern Recognition* **40**(7): 2038–2048.

Zhang, Y. and Zhou, Z.-H. (2008). Multi-label dimensionality reduction via dependency maximization, *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, Chicago, IL, pp. 1053–1055.

Zou, Z., Gao, H. and Li, J. (2010). Discovering frequent subgraphs over uncertain graph databases under probabilistic semantics, *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Washington, DC, pp. 633–642.

Author Biographies



Xiangnan Kong is a Ph.D. student in the Department of Computer Science, University of Illinois at Chicago, USA. He received a B.S. degree in Computer Science in 2006 and an M.S. degree of Computer Science from Nanjing University, China. In 2009, he joined the Next Generation Data Mining and Social Computing (NGDS) Lab from University of Illinois at Chicago. He has been working in the area of data mining and machine learning in general, and his current research is focused on graph classification, semi-supervised learning and multi-label learning with applications from chem/bioinformatics and social networks.

Philip S. Yu received his Ph.D. degree in E.E. from Stanford University. He is a Professor in Computer Science at the University of Illinois at Chicago and also holds the Wexler Chair in Information Technology. Dr. Yu spent most of his career at IBM, where he was manager of the Software Tools and Techniques group at the Watson Research Center. His research interests include data mining, database and privacy. He has published more than 620 papers in refereed journals and conferences. He holds or has applied for more than 350 US patents.

Dr. Yu is a Fellow of the ACM and the IEEE. He is an associate editor of ACM Transactions on Knowledge Discovery from Data. He was the Editor-in-Chief of IEEE Transactions on Knowledge and Data Engineering (2001-2004). He received a Research Contributions Award from IEEE Intl. Conference on Data Mining (2003).