Multi-Label Feature Selection for Graph Classification

Xiangnan Kong
Department of Computer Science
University of Illinois at Chicago, IL, USA
xkong4@uic.edu

Philip S. Yu
Department of Computer Science
University of Illinois at Chicago, IL, USA
psyu@cs.uic.edu

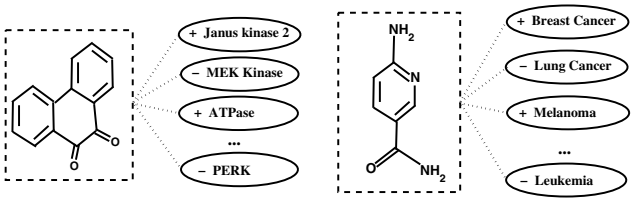
***Abstract*—Nowadays, the classification of graph data has become an important and active research topic in the last decade, which has a wide variety of real world applications, *e.g.* drug activity predictions and kinase inhibitor discovery. Current research on graph classification focuses on single-label 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, named gHSIC, 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 on real-world tasks 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.**

I. 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 raise one fundamental challenge for data mining research: the complex structure and lack of vector representations. 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 (binary classification) [22], [20], 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 [17], [19] and bioinformatics [5]. 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 [3]). 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.



(a) Kinase Inhibitor (b) Anti-Cancer Drug

Figure 1. Two Examples of multi-label graphs.

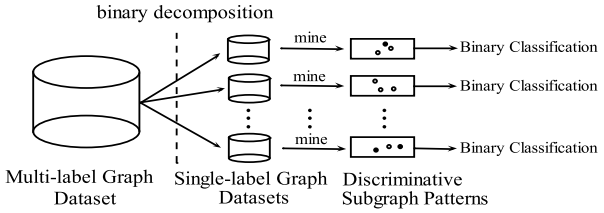
Despite its value and significance, the multi-label feature

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 twofold:

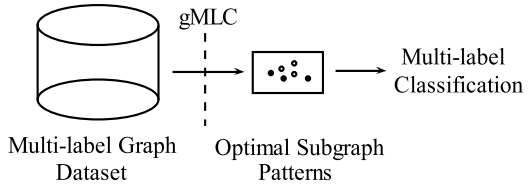
Graph Data: One fundamental problem in multi-label feature selection on graph data lies in the complex structures and lack of feature representations of graphs. Conventional feature selection approaches in vector spaces assume, explicitly or implicitly, that a full set of features is given before the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

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 [15], [22], [20]. 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 [19], [5].

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. In order to evaluate our proposed model, we perform comprehensive experiments on real-world multi-label graph classification tasks. 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.



(a) One-vs-All Single-label Graph Feature Selection



(b) gMLC Multi-label Graph Feature Selection

Figure 2. Two types of Feature Selection Process for Multi-label Graph Classification

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. Then introduce the preliminary concepts, give the problem analysis and present the gHSIC criterion in Section III and Section IV; In Section V, we derive a branch and bound algorithm gMLC based upon gHSIC. Then Section VI reports the experiment results. In Section VII, we conclude the paper.

II. 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 [3]), which transforms the multi-label problem into multiple binary classification problems, one for each label. ML-KNN[24] 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 [5] 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 [17], [21], decision trees [4], maximal margin methods [7], [13] and ensemble methods[6], *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 [23]. 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 [11], FSG [16], MoFa [1], FFSM [10], and Gaston [18]. Supervised subgraph feature extraction approaches have also been proposed in literature, such as LEAP [22], CORK [20], which look for discriminative subgraph patterns for graph classifications, and gSSC[14] for semi-supervised classification.

Our approach is also relevant to graph feature selection approaches based on Hilbert-Schmidt independence criterion [2], 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.

III. 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 *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 *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 (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) $\forall v \in \mathcal{V}'$, $l'(v) = l(\psi(v))$; (2) $\forall (u,v) \in E'$, $(\psi(u), \psi(v)) \in E$ and $l'(u,v) = l(\psi(u), \psi(v))$. 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 follow: 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.

In the following sections, we will first introduce the optimization framework for selecting informative subgraph features from multi-label graphs, then propose an efficient subgraph mining strategy using branch-and-bound to avoid exhaustive enumeration.

IV. Optimization Framework

In this section, we address the problem (1) discussed in Section III 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:

$$\mathcal{T}^* = \underset{\mathcal{T} \subseteq \mathcal{S}}{\operatorname{argmax}} \mathcal{E}(\mathcal{T}) \qquad \text{s.t. } |\mathcal{T}| \leq t, \tag{1}$$

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 [20], [2]. 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 [25].

Many criteria that can be used as dependence evaluation. In this paper, we will derive an evaluation criterion for multi-label graph classification based upon a dependence evaluation criterion named Hilbert-Schmidt Independence Criterion (HSIC) [8]. In detail, by deriving from the definition of HSIC, we can rewrite the optimization problem in Eq. 1 as follow:

$$\max_{\mathcal{T}} \qquad \operatorname{tr}(K_{\mathcal{T}} H L H) \qquad \text{s.t. } \mathcal{T} \subseteq \mathcal{S}, |\mathcal{T}| \leq t \tag{2}$$

where tr$(\cdot)$ is the trace of matrix and $H = [H_{ij}]_{n \times n}$, $H_{ij} = \delta_{ij} - 1/n$, δ_{ij} is the indicator function which takes 1 when $i = j$ and 0 otherwise. $K_{\mathcal{T}}$ denote the matrix of the inner product of graphs' feature vectors corresponding to subgraph feature set $\mathcal{T}$, which is a kernel matrix of graphs 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 $D_{\mathcal{T}} = \operatorname{diag}(\boldsymbol{\delta}_{\mathcal{T}})$ 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 $g_i \in \mathcal{T}$. $L = [L_{ij}]^{n \times n}$ is a kernel matrix for the graph's multiple labels with the kernel function $l(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle \psi(\boldsymbol{y}_i), \psi(\boldsymbol{y}_j) \rangle$. In our current implementation, $l(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle$ is used as the label kernel, and other kernels can also be directly used in this formulation.

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

$$\begin{aligned} &\operatorname{tr}(K_{\mathcal{T}} H L H) \\ =&\operatorname{tr}\left(X^\top {D_{\mathcal{T}}}^\top D_{\mathcal{T}} X H L H\right) \\ =&\operatorname{tr}\left(D_{\mathcal{T}} X H L H X^\top {D_{\mathcal{T}}}^\top\right) \\ =&\sum_{g_i \in \mathcal{T}} \left({\boldsymbol{f}_i}^\top H L H \boldsymbol{f}_i\right) = \sum_{g_i \in \mathcal{T}} \left({\boldsymbol{f}_i}^\top M \boldsymbol{f}_i\right) \end{aligned}$$

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

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

DEFINITION *4 (*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 $M = HLH$. We define a quality criterion q called gHSIC, for a subgraph feature g as

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

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 L and M are positive semi-definite, for any subgraph pattern g, $q(g) \geq 0$.

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:

$$\mathcal{T}^* = \{g_i | i \leq t\}. \tag{5}$$

V. THE PROPOSED SOLUTION

Now we address the second problem discussed in Section III, 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.* [22], [14], which put their evaluation criteria into the subgraph pattern mining steps and develop constrains to prune search spaces, we take a similar approach

by deriving a different constrain for multi-label cases. In order to avoid the exhaustive search, we proposed a branch-and-bound algorithm, named gMLC, which is summarized as follow: 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.

A. Subgraph Enumeration

In order to enumerate all subgraphs from a graph dataset, we adopted an efficient algorithm, gSpan, proposed by Yan et al[23]. 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 [23]. 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.

B. 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 follow:

THEOREM *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 follow:

$$\hat{q}(g) = \boldsymbol{f}_g{}^\top \hat{M} \boldsymbol{f}_g \tag{6}$$

where the matrix $\hat{M}$ 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 $q(g) = \boldsymbol{f}_g{}^\top M \boldsymbol{f}_g$.

Proof:

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

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,

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

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

$\mathcal{T}$ = 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
    $g_{min} = \operatorname{argmin}_{g' \in \mathcal{T}} q(g')$ 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

Figure 3. The gMLC algorithm

C. 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 [14] 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 any supergraph $g' \supseteq g$ ($q(g') \leq \hat{q}(g)$) according to the bound defined in Eq. (6), 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.

VI. Experiments

A. Experimental Setup

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 follow:

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 another benchmark dataset, NCI[1] [22], which consists anti-cancer activity records chemical compounds 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 classification dataset with 812 graphs assigned with 10 candidate labels. Table II 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] [9], 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, and {NE, N} as 'negative', which is the same setting as [12], [15]. 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).
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[3], 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.

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

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

Table I
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 II
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

Evaluation Metrics: Multi-label classification require different evaluation metrics than conventional single-label classification problems. Here we adopt some metrics used in [5], [24] 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). We have the following evaluation criteria:

a) Ranking Loss [5]: 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:

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

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

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

[3]Assay IDs include: 1416(PERK), 1446(JAK2), 1481(ATPase), 1531(MEK)

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.

b) Average Precision [24]: 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:

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

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

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

And $AvgPrec \in [0, 1]$, the larger 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. All experiments are conducted on machines with 4 GB RAM and Intel Xeon^{TM}Quad-Core CPUs of 2.40 GHz.

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

- Multi-label feature selection + multi-label classification (gMLC + BOOSTEXTER): We first use gMLC to find a set of optimal subgraph features. Then BOOSTEXTER [19] 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 + binary classifications (gMLC + SVM): We first use gMLC 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. We use SVM-light software package[4] to train the SVMs, where the parameters are set as default settings.
- Binary decomposition + single-label feature selection + binary classifications (Binary IG+ SVM): We compare with another baseline using a binary decomposition method similar to [3]: 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.
- Top-k Frequent subgraph features + multi-label classification (Freq + 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 is used as the multi-label classifier.

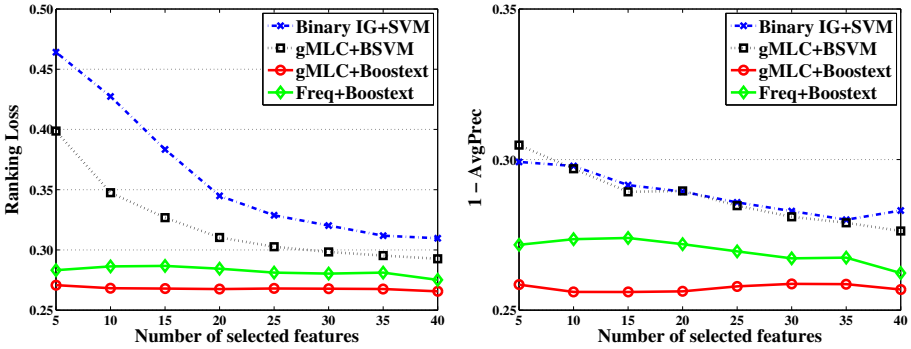
[4]http://svmlight.joachims.org/

B. 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 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.

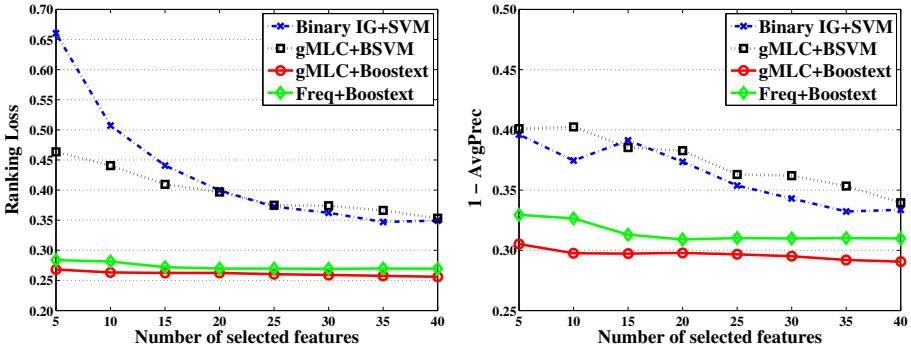
Now, we first study the effectiveness of selecting subgraph features by comparing two approaches: gMLC+SVM, Binary IG+ SVM, where the binary SVMs are used as base learners. It is worth noticing that, our gMLC is specially designed for conventional multi-label classification methods which require one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



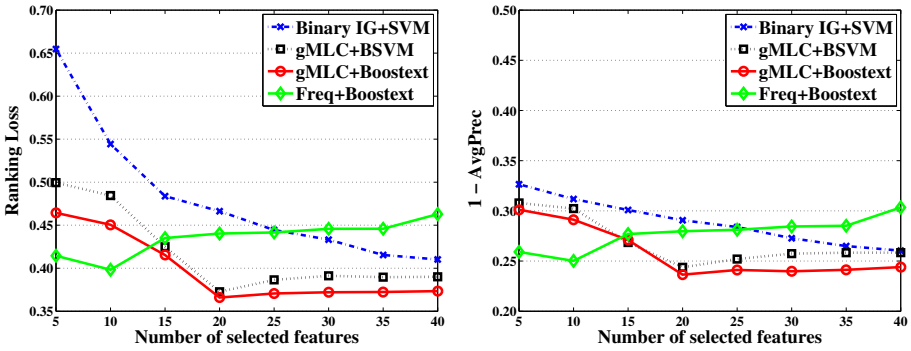
(a) Ranking Loss (b) 1 - AvgPrec

Figure 4. Multi-label graph classification performances on Anti-cancer Activity Prediction



(a) Ranking Loss (b) 1 - AvgPrec

Figure 5. Multi-label graph classification performances on Kinase Inhibition Prediction

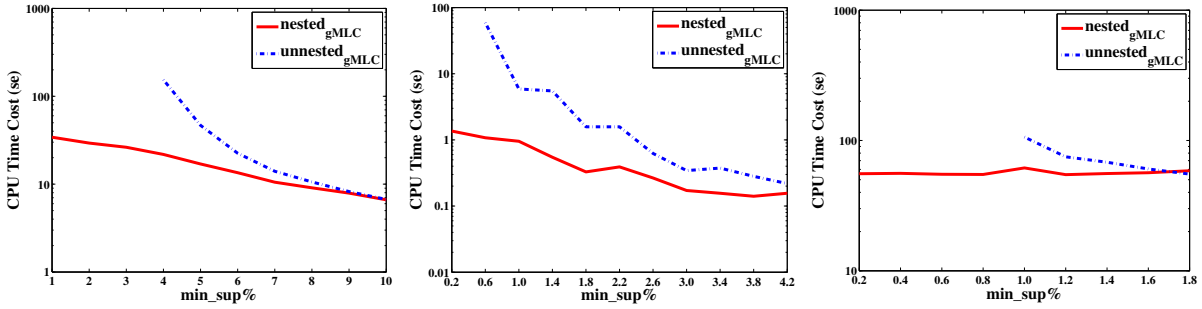


(a) Ranking Loss (b) 1 - AvgPrec

Figure 6. Multi-label graph classification performances on Toxicology Prediction

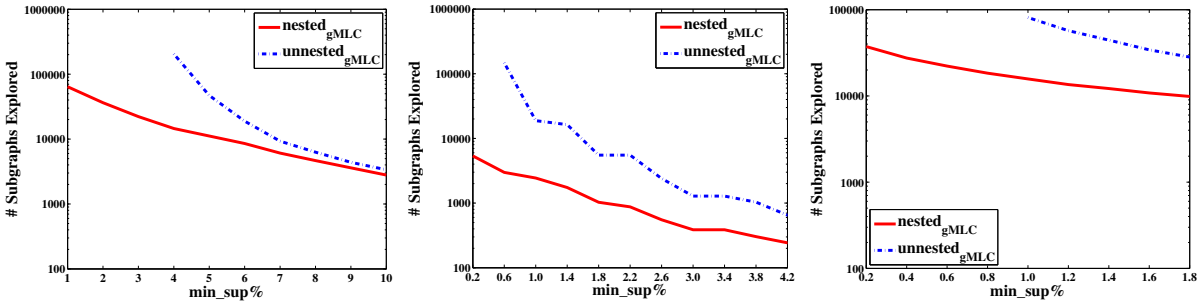
label concepts in order to make a fair comparison. So our only choice is comparing the following two methods: gMLC+BOOSTEXTER and Freq+BOOSTEXTER. We observe that on most tasks gMLC+BOOSTEXTER's performances are better than Freq+BOOSTEXTER, *i.e.* multi-label classification approaches without gHSIC subgraph feature selection. 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.

We further observe that in all tasks and evaluation criteria, our multi-label feature selection algorithm with multi-label classification (gMLC+BOOSTEXTER) outperforms the



a) Anti-cancer Prediction b)Toxicology Analysis c) Kinase inhibition

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



a) Anti-cancer Prediction b) Toxicology Analysis c) Kinase inhibition

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

binary decomposition approach using single-label feature selections (Binary IG+ SVM). gMLC+BOOSTEXTER can achieve good performances with only a small number of features. Although this comparison is not quite fair, and 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.

C. Effectiveness of Subgraph Search Space Pruning

In our second experiment, we evaluated the effectiveness of the upper-bound for gHSIC proposed in Section V-B. 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 V-B 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.

VII. CONCLUSION

In this paper, we study the problem multi-label feature selection for graph classification, 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 shows that multiple labels can help selecting useful features for graph classification. Moreover, label correlations among the multiple labels can be very useful for multi-label feature selection problem in graph data. In our current implementation, we only calculate the inner-products to get the label kernel matrix. However, by adopting more advanced kernels, the label correlations can also be considered under the current gMLC framework.

VIII. Acknowledgements

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

References

[1] C. Borgelt and M. Berthold. Mining molecular fragments: Finding relevant substructures of molecules. In *ICDM*, pages 211–218, Maebashi City, Japan, 2002.

[2] K. M. Borgwardt. *Graph Kernels*. PhD thesis, Ludwig-Maximilians-University Munich, 2007.

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

[4] F. D. Comité, R. Gilleron, and M. Tommasi. Learning multi-label altenating decision tree from texts and data. In *MLDM*, pages 35–49, Leipzig, Germany, 2003.

[5] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *NIPS*, pages 681–687. 2002.

[6] I. Vlahavas G. Tsoumakas. Random k-labelsets: An ensemble method for multilabel classification. In *ECML*, pages 406–417, Warsaw, Poland, 2007.

[7] S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classification. In *PAKDD*, pages 22–30, Sydney, Australia, 2004.

[8] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *ALT*, pages 63–77, Singapore, 2005.

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

[10] J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraph in the presence of isomorphism. In *ICDM*, pages 549–552, Melbourne, FL, 2003.

[11] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *PKDD*, pages 13–23, Lyon, France, 2000.

[12] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *ICML*, pages 321–328, Washington, DC, 2003.

[13] H. Kazawa, T. Izumitani, H. Taira, and E. Maeda. Maximal margin labeling for multi-topic text categorization. In *NIPS*, pages 649–656. 2005.

[14] X. Kong and P. Yu. Semi-supervised feature selection for graph classification. In *KDD*, pages 793–802, Washington, DC, 2010.

[15] T. Kudo, E. Maeda, and Y. Matsumoto. An application of boosting to graph classification. In *NIPS*, pages 729–736. 2005.

[16] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *ICDM*, pages 313–320, San Jose, CA, 2001.

[17] A. McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI'99 Workshop on Text Learning*, Orlando, FL, 1999.

[18] S. Nijssen and J. Kok. A quickstart in frequent structure mining can make a difference. In *KDD*, pages 647–652, Seattle, WA, 2004.

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

[20] M. Thoma, H. Cheng, A. Gretton, J. Han, H. Kriegel, A. Smola, L. Song, P. Yu, X. Yan, and K. Borgwardt. Near-optimal supervised feature selection among frequent subgraphs. In *SDM*, pages 1075–1086, Sparks, Nevada, 2009.

[21] N. Ueda and K. Saito. Parametric mixture models for multi-labeled text. In *NIPS*, pages 721–728. 2003.

[22] X. Yan, H. Cheng, J. Han, and P. Yu. Mining significant graph patterns by leap search. In *SIGMOD*, pages 433–444, Vancouver, BC, 2008.

[23] X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. In *ICDM*, pages 721–724, Maebashi City, Japan, 2002.

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

[25] Y. Zhang and Z.-H. Zhou. Multi-label dimensionality reduction via dependency maximization. In *AAAI*, pages 1053–1055, Chicago, IL, 2008.