Transfer Significant Subgraphs across Graph Databases

Xiaoxiao Shi* Xiangnan Kong* Philip S. Yu*†

Abstract

A key step of graph classification is to identify informative subgraphs that encode label information. For instance, in drug efficacy prediction, the drugs (chemical compounds) effective against the same disease usually contain similar chemical-subgraphs effective to control the disease. Then, one can use such chemical subgraphs to identify effective drugs. We call these subgraphs *significant subgraphs*. In this paper, the aim is to utilize the significant subgraphs from related graph datasets to help label graphs of the target dataset. For example, we utilize the breast cancer drug data, and transfer the anti-cancer subgraphs to help label another set of drug data against lung cancer. To do so, we propose a Bayesian-based transfer learning model. The key idea is to first evaluate the similarity between the target and source datasets by estimating the degree they share on their significant subgraphs. This dataset similarity is then used to judiciously select significant subgraphs from similar (related) datasets to the target dataset. An optimization problem is devised to maximize the likelihood that the selected subgraphs are significant in the target dataset. The objective function is further proven to have the antimonotone property which can help prune the search space significantly. Sixteen sets of experiments show that the proposed algorithm can effectively reduce the error rates by as much as 40%. More importantly, it is 10 times faster than the comparison models, which include unsupervised and supervised significant subgraph mining algorithms.

1 Introduction

Transfer learning (*e.g.*, [1]) attracts intensive attentions in recent years. It aims at borrowing supervision knowledge from one dataset to help the learning on another dataset. In text classification and opinion mining, transfer learning is applied to reuse related words (*e.g.*, [2, 3]) or sentiment words (*e.g.*, [4]) to improve the accuracy. However, for graph database (*e.g.*, chemical topology data, XML structure data), transfer learning is a difficult task.

- First, there is nothing obvious to transfer, or anything known that can make a successful transfer in graph database.
- Second, there is no obvious indicator to identify task relatedness between two graph datasets. Hence, it is difficult to tell when transfer learning works.

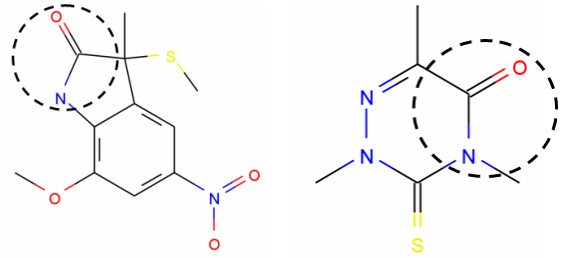
In this paper, we investigate these problems, and apply transfer learning to improve the effectiveness and efficiency of graph pattern mining and graph classification.

Motivation Note that one key issue in graph classification is to find a set of subgraphs that encode label information, and use the subgraphs as features. We call these subgraphs *significant subgraphs*. Traditionally, significant subgraphs can be obtained by two steps sequentially or interactively: (a) perform frequent pattern mining (*e.g.*, [5, 6]); (b) view the frequent subgraphs as features, and perform supervised feature selection to obtain significant graphs (*e.g.*, [7, 8]). Conventional feature selection assumes, explicitly or implicitly, there are a large amount of labeled examples.

However, class labels are sometimes extremely expensive and difficult to obtain. This is particularly true in graph dataset. For example, in molecular medicine, it requires time, efforts and excessive resources to test drugs' efficacies by preclinical studies and clinical trials. Hence, in practice, there may be only a limited number of labeled examples for a learning task. Without sufficient labeled data, it is difficult to find informative significant subgraphs. Hence, to improve significant subgraph mining, it is desirable to obtain more supervision knowledge from other sources (datasets). Intuitively, if the source datasets are correlated with the target dataset, the source significant subgraphs may contain information also useful in the target dataset. For example, Fig. 1 presents two drug data from different graph datasets in the National Cancer Institute (NCI) database. The left graph is a drug fragment that has anti-cancer property effective against leukemia, and the right chemical is effective against lung cancer. We highlight their significant subgraphs in Fig. 1 (mined by

*Computer Science Department, University of Illinois at Chicago, USA. {xshi9, xkong4, psyu}@uic.edu.

†Computer Science Department, King Abdulaziz University, Jeddah, Saudi Arabia



(a) Effective drug fragment against leukemia (b) Effective drug fragment against lung cancer

Figure 1: The same significant subgraph may have the same semantics in different graph datasets

the method in [9]). It is clear that the two chemicals share the same significant subgraph that probably has anti-cancer property. Thus, in this example, when the number of labeled graphs is limited in one dataset, it is possible and desirable to utilize the significant subgraphs from the other dataset to improve learning.

The proposed model In this paper, a statistical framework is first proposed to formally model significant subgraph mining. It is derived by introducing a latent variable to infer how likely a candidate subgraph is significant. We further generalize the statistical model with Bayesian theory to consider a latent variable space. The variable space is then used as a bridge to enable the knowledge transfer from related source datasets to the target dataset. Moreover, two key challenges are addressed:

1. First, how can one identify related and similar sources? Although it is well established in the text domain, commonality of words is a good indicator on relatedness between source and target domains, there is no previous study showing the effective indicators for graph domains. Our intuition is to estimate the degree that the source and target datasets share on their significant subgraphs. If they share many significant subgraphs, the graph datasets are related. Specifically, Kullback-Leibler divergence is used to estimate the amount of shared subgraphs, and it is incorporated into the Bayesian framework to automatically assign higher weights to related sources.

2. Second, how can one make good use of the related source datasets to improve the accuracy? The key idea is to view the significant subgraphs from related source datasets as strong candidate-subgraphs for the target dataset. The Bayesian framework thus summarizes the weighted "votes" from all source datasets, and infers the likelihood that the candidate subgraph is significant.

Table 1: Notation Descriptions

Notations	Descriptions
$\mathcal{T}$	Target graph dataset
$\mathcal{S}_i$	The set of significant subgraphs of the i-th source dataset
$\mathcal{P}$	A pool of source significant subgraphs $\mathcal{P} = \{\mathcal{S}_1, \cdots, \mathcal{S}_t\}$
G	A graph instance
g	A subgraph
$\mathcal{G}_\mathcal{T}$	Subgraph space of dataset $\mathcal{T}$
$\mathbf{sig}(\mathcal{G}_\mathcal{T})$	The set of significant subgraphs of $\mathcal{T}$
$\theta_\mathcal{T}$	A latent variable to indicate how likely a subgraph g is significant in $\mathcal{T}$ via $p(g\|\theta_\mathcal{T})$
$\theta_\mathcal{S}$	The latent variable of source dataset $\mathcal{S}$
Θ	Generalized latent variable space
	Input and output of the problem setting
Input	(a) Target dataset $\mathcal{T}$ (b) A pool of source significant subgraphs $\mathcal{P}$
Output	$\mathbf{sig}(\mathcal{G}_\mathcal{T})$ defined in Eq. 3.8.

An optimization problem is devised to select significant subgraphs based on the above intuition. It aims at maximizing the likelihood that the selected subgraphs are significant in the target dataset. The objective function is further proven to have the antimonotone property which can prune the search space significantly. It is important to emphasize that to increase the applicability of the proposed model, we do not require that the user needs to know a priori whether the source datasets are related to the target dataset. For example, Yeast dataset is given as an auxiliary source when the target dataset is about lung cancer drugs. It is a capability of the proposed model to judiciously decrease the weights of (or filter out) unrelated datasets, in order to avoid them hurting the learning accuracy. Sixteen sets of experiments were performed to evaluate the effectiveness and efficiency of the proposed model. The utility of the resulting significant subgraphs was evaluated by using the subgraphs as features in classification tasks. It is observed that the proposed model runs 10 times faster than the comparison models. In addition, its resulting significant subgraphs can reduce the classification error rates by as much as 40%, when compared with both unsupervised and supervised subgraph mining algorithms.

2 Problem Formulation

In this section, we formulate the problem of significant subgraph mining for graph classification, and extend it to the scenario of transfer learning. First, let $\mathcal{T} = \{G_1, \cdots, G_n\}$ denote the entire target graph dataset,

which consists of n *connected graphs.*

DEFINITION 2.1. (CONNECTED GRAPH) *A graph is represented as $G = (\mathcal{V}, E, \mathcal{L}, l)$, where $\mathcal{V}$ is a set of vertices $\mathcal{V} = \{v_1, \cdots, v_{n_v}\}$, and $E \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges, and $\mathcal{L}$ is the set of symbols for the vertices and the edges, and $l : \mathcal{V} \cup E \to \mathcal{L}$ 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 2.2. (SUBGRAPH) *Let $g = (\mathcal{V}_g, E_g, \mathcal{L}_g, l_g)$, 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 $f : \mathcal{V}_g \to \mathcal{V}$ s.t. (1) $\forall v \in \mathcal{V}_g$, $l_g(v) = l(f(v))$; (2) $\forall (u, v) \in E_g$, $(f(u), f(v)) \in E$, and $l_g(u, v) = l(f(u), f(v))$. If g is a subgraph of G, then G is a supergraph of g.*

DEFINITION 2.3. (VECTOR BASED REPRESENTATION) *We adopt the idea of subgraph-based graph representation, which describes each graph object G_i as a feature vector $\boldsymbol{x}_i = [x_i^1, \cdots, x_i^m]^\top$ corresponding to a set of subgraph patterns $\{g_1, \cdots, g_m\}$. Denote x_i^k as the feature corresponding to the subgraph pattern g_k. Define $x_i^k = 1$ iff g_k is a subgraph of G_i ($g_k \subseteq G_i$), otherwise $x_i^k = 0$.*

In addition to the target graph dataset, we are also given a pool of auxiliary source datasets. It is assumed that the top k significant subgraphs in each source dataset are already derived. One aim of the proposed model is to determine whether the source significant subgraphs are useful or not. More formally, the pool of source significant subgraph datasets is defined as follows.

DEFINITION 2.4. (SOURCE SIGNIFICANT SUBGRAPHS) *Let $\mathcal{P} = \{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_t\}$ be the pool of source significant subgraph datasets where $\mathcal{S}_i = \{g_1^{(i)}, g_2^{(i)}, \cdots, g_k^{(i)}\}$ ($1 \leq i \leq t$) is a set of significant subgraphs for the i-th source dataset. It is assumed that these significant subgraphs are already derived. For example, they can be identified by domain experts, or by chemical test, or by supervised method (e.g., [9, 7, 10]) with sufficient training data, .*

The useful notations are summarized in Table 1. Thus, the studied research problem can be described as follows: how can one utilize the pool of source significant subgraphs $\mathcal{P}$ to improve significant subgraph mining in the target dataset $\mathcal{T}$ in either of the two settings:

- Supervised pattern mining: Assume that the first l ($l \ll n$) graphs within $\mathcal{T}$ are labeled by $\{y_1, \cdots, y_l\}$, where y_i denotes the class label assigned to G_i.

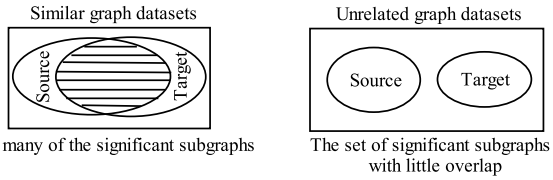


Figure 2: Significant subgraphs encode the label information; hence similar/related graph datasets share many of their significant subgraphs

- Unsupervised pattern mining: There is no target labeled graph used in finding the significant subgraphs.

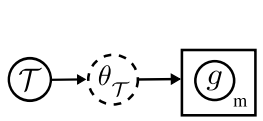
The aim is to use the significant subgraphs as features, and assign labels to the unlabeled graphs. We investigate a general approach applicable in the above two settings.

3 Transfer Significant Subgraphs

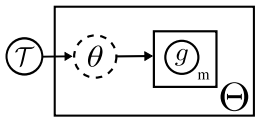
We note that in the text domain, it is relatively easy to judge the relatedness of the source and target domains by looking at the common vocabularies or topics. It is much harder to judge the relatedness of two graph datasets as there is no simple intuitive way to do that. While drugs for two different types of cancers are not necessarily always related, drugs for seemingly unrelated diseases such as HIV and cancers may be related, *e.g.*, HIV drug, nelfinavir, is under clinical testing for cancer treatment. It is thus desirable to have an automated mechanism to make this evaluation and avoid transferring from unrelated sources. We next explain the intuition behind the algorithm.

3.1 Intuition The significant subgraphs from related source datasets are utilized to improve pattern mining in the target dataset. Two challenges are addressed:

- First, how can one evaluate whether the graph datasets are similar and related? A high level intuition is to estimate how much they share on the significant subgraphs. As in Fig. 2, if two datasets are similar and related, they share most of their significant subgraphs that encode similar label information; otherwise, they share few significant subgraphs. To evaluate the amount of common subgraphs, we first estimate the significant subgraphs of the target dataset $\mathcal{T}$ (1) by its frequent subgraphs in an unsupervsied setting, or (2) by the significant subgraphs mined with a limited number of labels in a supervised setting. Note that it is just a rough estimation of the target significant subgraphs since we use none (in the unsupervised setting) or



(a) Significant subgraph mining.



(b) Transfer learning with source significant subgraphs.

Figure 3: Graphical model representation of significant subgraph mining (unobserved latent variable is highlighted with dash line)

only a limited number of labeled graphs (in the supervised setting). We then compare the set of estimated target significant subgraphs with the source significant subgraphs. If they share many common subgraphs, they are labeled as related. This step is formulated and solved by the Kullback-Leibler divergence [11].

- Second, how can one make good use of the related source datasets to improve the accuracy and the efficiency? The general idea is as follows. If the source and target dataset is similar, we view the source significant subgraphs as strong candidates of significant subgraphs in the target dataset. This is modeled by first assigning different weights to the source datasets according to their relatedness. A Bayesian model is then proposed to estimate the likelihood that a candidate subgraph is significant by summarizing the weighted "votes" from the source datasets.

The two steps estimate whether a candidate subgraph is significant. The final aim is to maximize the likelihood that the selected subgraphs are significant.

3.2 A Statistical Explanation of Significant Subgraph Mining In this section, we first introduce a statistical explanation of significant subgraph mining, and then extend it to the scenario of transfer learning. Given the target dataset $\mathcal{T}$ and all its subgraphs $\mathcal{G}_\mathcal{T} = \{g_1, g_2, \cdots\}$, the aim is to extract the set of significant subgraphs $\mathbf{sig}(\mathcal{G}_\mathcal{T}) \subseteq \mathcal{G}_\mathcal{T}$ that encodes label information. It can be explained as a sampling process from the subgraph space $\mathcal{G}_\mathcal{T}$ as sketched in Fig. 3(a). The sampling process contains two steps. First, a latent variable $\theta_\mathcal{T}$ is generated from the dataset $\mathcal{T}$. Given any subgraph g, the latent variable $\theta_\mathcal{T}$ determines the likelihood that the subgraph g is significant. Second, the significant subgraphs are sampled from the subgraph space $\mathcal{G}_\mathcal{T}$ with the distribution proportional to $p(g|\theta_\mathcal{T})$. If $p(g|\theta_\mathcal{T})$ is large, it means that g is significant with high probability. Hence, the sampling process described above specifies a generative model for significant subgraphs. With the aim to select a subset of subgraphs that are significant, the objective function can be written as maximizing the joint probability $p\big(\mathcal{T}, \theta_\mathcal{T}, \mathbf{sig}(\mathcal{G}_\mathcal{T})\big)$

$$\max_{\mathbf{sig}(\mathcal{G}_\mathcal{T})} p\big(\mathcal{T}, \theta_\mathcal{T}, \mathbf{sig}(\mathcal{G}_\mathcal{T})\big) \tag{3.1}$$

Usually, for the sake of computational efficiency, $\mathbf{sig}(\mathcal{G}_\mathcal{T})$ is directly obtained by

$$\mathbf{sig}(\mathcal{G}_\mathcal{T}) = \Big\{ g | g \in \mathcal{G}_\mathcal{T} \text{ and } p\big(\mathcal{T}, \theta_\mathcal{T}, g\big) \geq \lambda \Big\} \tag{3.2}$$

where λ is a threshold to filter out the subgraphs that are less likely to be significant. Different interpretations of the joint probability derive different graph pattern mining algorithms:

1. Unsupervised pattern mining, or frequency based pattern mining (*e.g.*, [5, 12, 13]): the following equation can be obtained from Fig. 3(a):

$$\begin{aligned} p(g|\mathcal{T}) &= \frac{p(g,\mathcal{T})}{p(\mathcal{T})} \\ &= \frac{\int p(g,\mathcal{T},\theta) d\theta}{p(\mathcal{T})} \qquad \text{(only one } \theta \text{ in Fig. 3(a): } \theta_\mathcal{T}) \\ &= \frac{p(\mathcal{T}) p(\theta_\mathcal{T}|\mathcal{T}) p(g|\theta_\mathcal{T})}{p(\mathcal{T})} \quad \text{(from Fig. 3(a))} \\ &= p(\theta_\mathcal{T}|\mathcal{T}) p(g|\theta_\mathcal{T}) \end{aligned} \tag{3.3}$$

By applying the above equation, the joint probability can then be written as:

$$\begin{aligned} p\big(\mathcal{T}, \theta_\mathcal{T}, g\big) &= p(\mathcal{T}) p(\theta_\mathcal{T}|\mathcal{T}) p(g|\theta_\mathcal{T}) = p(\mathcal{T}) p(g|\mathcal{T}) \\ &\propto p(g|\mathcal{T}) = \text{Frequency}(g) \end{aligned} \tag{3.4}$$

Note that $p(g|\mathcal{T})$ is the relative frequency [1] of the subgraph g in the dataset $\mathcal{T}$. This category of algorithms thus only selects frequent subgraphs as significant patterns.

One important advantage of the frequency based model is that the objective function satisfies the anti-monotone property. That is, if a subgraph g_j is a supergraph of g_i ($g_i \subseteq g_j$), we have $p(g_j|\mathcal{T}) \leq p(g_i|\mathcal{T})$. This property is the key to prune the search space and collect the subgraph patterns efficiently [5].

2. Supervised pattern mining, or feature selection based algorithms (*e.g.*, [9, 7, 8, 14]): the joint probability can also be written as:

$$\begin{aligned} p\big(\mathcal{T}, \theta_\mathcal{T}, g\big) &= p(\theta_\mathcal{T}|g,\mathcal{T}) p(g|\mathcal{T}) p(\mathcal{T}) \\ &\propto p(\theta_\mathcal{T}|g,\mathcal{T}) p(g|\mathcal{T}) \end{aligned} \tag{3.5}$$

[1] relative frequency: the number of graphs that contain the subgraph, divided by the total number of graphs.

Note that $\theta_{\mathcal{T}}$ is a latent variable indicating how significant the subgraph is to classification. One way to approximate the term $p(\theta_{\mathcal{T}}|g,\mathcal{T})$ is to consider how informative of the subgraph feature g over graph labels:

$$p(\theta_{\mathcal{T}}|g,\mathcal{T}) \propto \int_{G_i\in\mathcal{T}} p(y_i|g,G_i)dG \tag{3.6}$$

where G_i is a graph data in $\mathcal{T}$ with label y_i. The joint probability can then be written as:

$$\begin{aligned} p\big(\mathcal{T},\theta_{\mathcal{T}},g\big) &\propto p(\theta_{\mathcal{T}}|g,\mathcal{T})p(g|\mathcal{T}) \\ &\propto p(g|\mathcal{T})\int_{G_i\in\mathcal{T}} p(y_i|g,G_i)dG \\ &= \text{Frequency}(g)\times\text{Informative}(g) \end{aligned} \tag{3.7}$$

This category of algorithms thus considers two factors: (a) the subgraph has to be frequent (a large $p(g|\mathcal{T})$); (b) the subgraph is informative and useful to classification. This is usually done by supervised feature selection on a set of frequent subgraphs. However, this category of algorithm usually requires a large number of labeled graphs to approximate the term $\int_{\mathbf{x}_i\in\mathcal{T}} p(y_i|g,\mathbf{x}_i)d\mathbf{x}$.

Compared with the frequency based methods, the supervised significant subgraph mining incorporates labeled information to improve the result. It is thus more powerful to find the significant subgraphs that encode label information. However, in many applications, it is usually very difficult and expensive to obtain many labeled graphs, which limits the effectiveness of supervised subgraph selection. To solve this problem, we generalize the statistical model to utilize the rich set of significant subgraphs from auxiliary sources.

3.3 Incorporate Auxiliary Sources In this section, we generalize the above statistical model to utilize the significant subgraphs from auxiliary sources. To do so, we first generalize the latent variable θ in Fig. 3(a) by introducing a variable space Θ ($\theta\in\Theta$) as in Fig. 3(b). The generative model thus becomes: the significant subgraph g is sampled with distribution $p(g|\theta)$ where the latent variable θ is sampled from Θ with probability $p(\theta|\mathcal{T})$. Intuitively, θ can be the latent variable from source datasets, and we can use $p(\theta|\mathcal{T})$ to assign high weights to the source datasets related to the target dataset, and use $p(g|\theta)$ to assign high weights to the source significant subgraphs. We can then write the objective function as

$$\mathbf{sig}(\mathcal{G}_{\mathcal{T}}) = \Big\{g|g\in\mathcal{G}_{\mathcal{T}} \text{ and } p\big(\mathcal{T},\Theta,g\big)\geq\ \lambda\Big\} \tag{3.8}$$

where the joint probability can be written as (with Fig. 3(b)):

$$\begin{aligned} p\big(\mathcal{T},\Theta,g\big) &= \sum_{\theta\in\Theta} p\big(\mathcal{T},\theta,g\big) = \sum_{\theta\in\Theta} p(\mathcal{T})p(\theta|\mathcal{T})p\big(g|\theta\big) \\ &\propto \sum_{\theta\in\Theta} p(\theta|\mathcal{T})p\big(g|\theta\big) \end{aligned} \tag{3.9}$$

There are at least two advantages to apply the above generalized model to mine significant subgraphs. First, the proposed model can avoid overfitting: it does not exclusively use the latent variable θ learned from the target dataset. This is particularly important when the latent variable θ is learned from only a limited number of labeled data. Second, from the view point of transfer learning, the supervision knowledge from auxiliary sources can be conveniently incorporated by using θ learned from the source datasets. We next introduce how to obtain

- $p\big(g|\theta\big)$ that assigns high weights to the subgraphs that are identified as significant in a source dataset.
- $p(\theta|\mathcal{T})$ that assigns weights to the source datasets according to their relatedness to the target dataset.

Using Source Significant Subgraphs via $p(g|\theta)$ Denote the latent variable of the i-th source dataset as $\theta_{\mathcal{S}_i}$, and the set of significant subgraphs as $\mathbf{sig}(\mathcal{G}_{\mathcal{S}_i})$. Since the significant subgraphs are generated from the latent variable, we then define

$$\begin{aligned} p(g|\theta_{\mathcal{S}_i}) &= \begin{cases} 1 & \text{if } g\subseteq g'\subseteq \mathbf{sig}(\mathcal{G}_{\mathcal{S}_i}) \text{ of } \mathcal{S}_i \\ 0 & \text{Otherwise} \end{cases} \\ &= \begin{cases} 1 & \text{if there exists a supergraph of } g \text{ significant in } \mathcal{S}_i \\ 0 & \text{Otherwise} \end{cases} \end{aligned} \tag{3.10}$$

where g' is a significant subgraph of the source dataset $\mathcal{S}$, and it is also a supergraph of g ($g\subseteq g'$). The term $p\big(g|\theta_{\mathcal{S}_i}\big)$ assigns high weights to the patterns that are identified as significant in the source dataset $\mathcal{S}_i$. We next discuss how to model the dataset similarity by $p(\theta|\mathcal{T})$.

Model Dataset Similarity via $p(\theta|\mathcal{T})$ We first define the variable space Θ as the joint of latent variables of the target and source datasets as $\Theta = \{\theta_{\mathcal{T}},\theta_{\mathcal{S}_1},\theta_{\mathcal{S}_2},\cdots,\theta_{\mathcal{S}_t}\}$. We then define $p(\theta_{\mathcal{S}_i}|\mathcal{T})$ as:

$$p(\theta_{\mathcal{S}_i}|\mathcal{T}) = \mathcal{N}\Big(\exp\big\{\mathbf{KL}\big(\mathbf{sig}(\mathcal{G}_{\mathcal{S}_i})\|\mathbf{sig}(\mathcal{G}_{\mathcal{T}})\big)\big\}-1;0,\sigma\Big) \tag{3.11}$$

where $\mathcal{N}(x;0,\sigma)$ is the normal distribution density function with zero mean, and σ serves as a normalization term to ensure $\int p(\theta|\mathcal{T})d\theta = 1$, and the term

$\mathbf{KL}\big(\mathbf{sig}(\mathcal{G}_{\mathcal{S}_i})\|\mathbf{sig}(\mathcal{G}_{\mathcal{T}})\big)$ is the Kullback-Leibler Divergence used to evaluate the difference of the two sets of subgraphs $\mathbf{sig}(\mathcal{G}_{\mathcal{S}_i})$ and $\mathbf{sig}(\mathcal{G}_{\mathcal{T}})$ in distribution. More specifically, we first represent each set of the subgraphs as a distribution of smaller subgraphs (described later), and then evaluate the differences of the two sets in distribution. This is similar to text classification when we think of each set of documents as a distribution of words. The KL divergence returns zero when the two datasets have the same distribution, and becomes large when the two datasets are unrelated [11]. There are two important properties of Eq. 3.11.

- First, $p(\theta_{\mathcal{S}_i}|\mathcal{T})$ gets the highest value when the source and target data have exactly the same distribution such that $\mathbf{KL}(\mathbf{sig}(\mathcal{G}_{\mathcal{S}_i})\|\mathbf{sig}(\mathcal{G}_{\mathcal{T}})) = 0$.
- Second, it can decrease exponentially when the source dataset is unrelated ($\mathbf{KL}(\mathbf{sig}(\mathcal{G}_{\mathcal{S}_i})\|\mathbf{sig}(\mathcal{G}_{\mathcal{T}}))$ is large). It is a key step to assign low weights to filter out unrelated sources.

Note that we need to determine $\mathbf{KL}\big(\mathbf{sig}(\mathcal{G}_{\mathcal{S}_i})\|\mathbf{sig}(\mathcal{G}_{\mathcal{T}})\big)$. Three steps are performed to calculate the term:

1. First, we estimate the target significant subgraphs $\mathbf{sig}(\mathcal{G}_{\mathcal{T}})$ by either (1) frequent subgraphs in an unsupervised setting; or (2) initial significant subgraphs mined from the labeled graphs in a supervised setting as in Eq. 3.7.
2. Second, we mine frequent subgraphs $\{f_1, f_2, \cdots\}$ on the combined dataset $\mathbf{sig}(\mathcal{G}_{\mathcal{T}}) \cup \mathbf{sig}(\mathcal{G}_{\mathcal{S}_i})$. This step is to discover small subgraph patterns that generate $\mathbf{sig}(\mathcal{G}_{\mathcal{T}})$ and $\mathbf{sig}(\mathcal{G}_{\mathcal{S}_i})$. Hence, both $\mathbf{sig}(\mathcal{G}_{\mathcal{T}})$ and $\mathbf{sig}(\mathcal{G}_{\mathcal{S}_i})$ can be represented as "bag of small subgraph patterns". Similar to the "bag of words" presentation in text mining, we can now define $P_{\mathcal{T}}(f_1)$ as the probability that the subgraph f_1 appears in the dataset $\mathbf{sig}(\mathcal{G}_{\mathcal{T}})$. This is calculated by the number of occurrences of the subgraph f_1 divided by the total number of occurring significant subgraphs. Then, with the data (subgraph) distributions, we can now compute the KL divergence.
3. Third, we calculate $\mathbf{KL}\big(\mathbf{sig}(\mathcal{G}_{\mathcal{S}_i})\|\mathbf{sig}(\mathcal{G}_{\mathcal{T}})\big)$ by the distributions described above:

$$\begin{aligned}&\mathbf{KL}\big(\mathbf{sig}(\mathcal{G}_{\mathcal{S}_i})\|\mathbf{sig}(\mathcal{G}_{\mathcal{T}})\big)\\ &= \sum_g \Big(P_{\mathcal{S}_i}(g)\mathbf{log}P_{\mathcal{S}_i}(g) - P_{\mathcal{S}_i}(g)\mathbf{log}P_{\mathcal{T}}(g)\Big)\end{aligned} \tag{3.12}$$

It is important to note that the first two steps can be very expensive when we enumerate all the frequent subgraphs. We thus only retain the subgraphs with at most l nodes where l is a parameter discussed in the experiment section. The KL divergence calculated on the subgraphs with at most l nodes are denoted as $\mathbf{KL}_l\big(\mathbf{sig}(\mathcal{G}_{\mathcal{S}_i})\|\mathbf{sig}(\mathcal{G}_{\mathcal{T}})\big)$. With the definition of $p(g|\theta)$ in Eq. 3.10 and $p(\theta|\mathcal{T})$ in Eq. 3.11, we can now apply Eq. 3.9 to evaluate how likely a candidate subgraph g is significant in the target dataset. We next introduce an efficient enumeration algorithm to select the set of subgraphs g with $p\big(\mathcal{T}, \Theta, g\big) > \lambda$.

3.4 Subgraph Enumeration and Pruning Eq. 3.9 is used to evaluate the likelihood of a given subgraph g to be significant. A naive approach is to enumerate all possible subgraphs, and select those subgraphs with high scores. Note that one difficulty to enumerate the subgraphs is to identify the isomorphism subgraphs [5]. To handle the isomorphism issue in graph pattern mining, we adopt the subgraph enumeration strategy in gSpan proposed by Yan et al [5]. The key idea of gSpan is to build a lexicographic order of all the edges of a graph, and then map each graph to a unique minimum DFS code as its canonical label. The minimum DFS codes of two graphs are equivalent iff they are isomorphic. Based on this lexicographic order, a depth-first search (DFS) strategy is used to 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 codes' order. More details can be found in [5]. In addition to isomorphism test, another issue is that the number of subgraphs grows exponentially with the number of nodes and edges. It is thus impractical to enumerate all of them. We next show the antimonotone property of Eq. 3.9, and introduce a pruning strategy based on the theorem.

THEOREM 3.1. *Given any two subgraphs $g, g' \in \mathcal{S}$, if g' is a supergraph of g ($g' \supseteq g$), then*

$$p\big(\mathcal{T}, \Theta, g\big) \geq p\big(\mathcal{T}, \Theta, g'\big) \tag{3.13}$$

Proof. First of all, according to Eq. 3.10 and the antimonotone property of [5], we can obtain:

$$p\big(g|\theta\big) \geq p\big(g'|\theta\big)$$

With the above property, we can prove the antimonotone of the general framework as:

$$\begin{aligned}p\big(\mathcal{T}, \Theta, g\big) &= \sum_{\theta\in\Theta} p(\mathcal{T})p(\theta|\mathcal{T})p\big(g|\theta\big)\\ &\geq \sum_{\theta\in\Theta} p(\mathcal{T})p(\theta|\mathcal{T})p\big(g'|\theta\big) = p\big(\mathcal{T}, \Theta, g'\big)\end{aligned} \tag{3.14}$$

Input: $\mathcal{T}$ is the target graph dataset; P is the pool of source datasets $P = \{S_1, S_2, \cdots, S_p\}$; λ is the threshold; l is the maximum size of subgraphs for KL divergence with default value 3;

Output: $\mathbf{sig}(\mathcal{G})$: Set of significant subgraphs

$\mathbf{sig}(\mathcal{G}) = \emptyset$;
`/* Recursively visit the DFS Code Tree in gSpan */`
g = currently visited subgraph in the DFS Code Tree in gSpan;
Calculate $p(\mathcal{T}, \Theta, g)$ as in Eq. 3.9 with the dataset similarity in Eq. 3.11 and weights of the source significant subgraphs in Eq. 3.10.
if $p(\mathcal{T}, \Theta, g) \geq \lambda$ **then**
    $\mathbf{sig}(\mathcal{G}) = \mathbf{sig}(\mathcal{G}) \cup \{g\}$;
    Depth-first search the subtree rooted from node g (goto Step 2);
end
return $\mathbf{sig}(\mathcal{G})$;

Algorithm 1: Significant Subgraph Mining with Auxiliary Sources

The antimonotone property allows us to prune the search space significantly as in Algorithm 1. In other words, for a subgraph g, if $p(\mathcal{T}, \Theta, g) < \lambda$, we can prune all the supergraphs $g' \supseteq g$. Note that Algorithm 1 is a general framework which can be used in both supervised and unsupervised settings. We will evaluate the framework in both settings in the next section.

4 Experiments

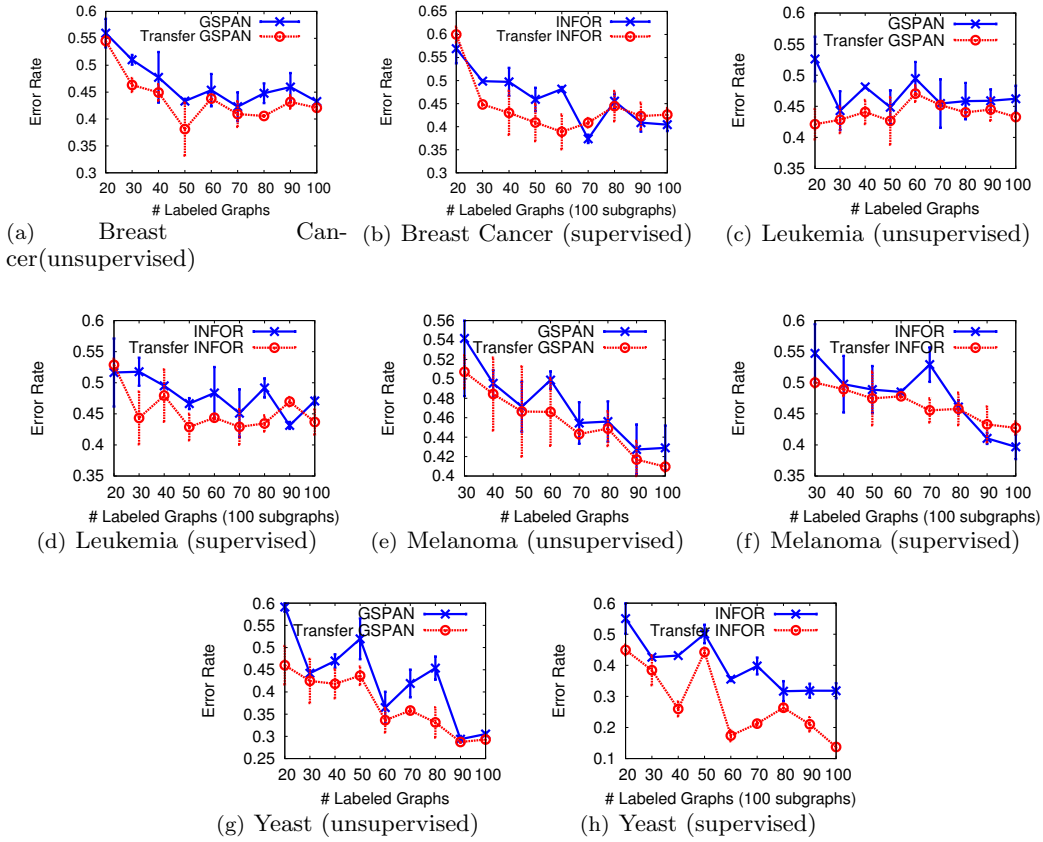
A transfer learning model is proposed to utilize auxiliary sources to improve significant subgraph mining. We evaluated it with 16 sets of experiments. The aim of the experiments is to answer the following questions:

1. Can related sources be identified to improve the accuracy?
2. Can the proposed model avoid using unrelated sources?
3. Can the proposed model improve the efficiency of significant subgraph mining?

4.1 Experimental Setup *Data Collections:* Eight graph datasets were collected from the NCI chemical graph database, which were divided into two groups as in Table 2. The two groups took turns to be the pool of source datasets, and the other one was set as the pool of target datasets. As such, for each experiment, we selected one dataset from the target group as the target dataset, and used all datasets from the source group as the pool of sources. Thus, we had 16 experiments in total: the 8 datasets took turns to be the target dataset in both supervised and unsupervised graph pattern mining. According to the problem setting, the top k most significant subgraphs in each source dataset were already derived. In the experiment, k was set to be 700 so that the source datasets contained rich information, but they might be totally irrelevant to the target dataset. These significant subgraphs were mined by the method in [9] along with all the labeled examples in the source datasets. The proposed model is supposed to find the significant subgraphs in the target dataset by making use of these source significant subgraphs. From Table 2, it can be observed that the source datasets may be totally unrelated to the target dataset. For example, the anti-cancer datasets should be similar to each other, but they should have weak similarity to the AIDS data. It is a capability of the proposed model to judiciously use only the related datasets, and avoid using the unrelated ones.

Comparison Methods: The proposed model can handle both unsupervised and supervised graph pattern mining. They are only different in how to estimate the target significant subgraphs when calculating the dataset similarity in Eq. 3.12. We evaluated the model in both settings. In the unsupervised setting, gSpan [5] was chosen as an example for comparison. Then, in the proposed algorithm, we correspondingly used gSpan as a based method to estimate the dataset similarity in Eq. 3.12, and we denote our method "Transfer-GSPAN". In the supervised setting, the method in [9] (using information gain to select features) was set to be the comparison method which is denoted as "INFOR". Then we correspondingly used [9] in Eq. 3.12, and we denote the supervised version of our model as "Transfer-INFOR". Note that the outputs of the models were sets of subgraph patterns. We then evaluated the utility of the subgraphs by using them as features in classification tasks. The proposed model was also studied in terms of efficiency.

4.2 Result Analysis In the experiment, all results were summarized on 10 runs, and in each run we randomly sampled certain number of target data as the target training data, and the rest were used as test data. We then used SVM to give the classification result. It is important to mention again that in the unsupervised setting, the significant subgraphs were extracted without using any label information. The purpose of classification and the labeled graphs reported in Fig. 4 and Fig. 5 was to evaluate the utility of the



(a) Breast Cancer(unsupervised) (b) Breast Cancer (supervised) (c) Leukemia (unsupervised)

(d) Leukemia (supervised) (e) Melanoma (unsupervised) (f) Melanoma (supervised)

(g) Yeast (unsupervised) (h) Yeast (supervised)

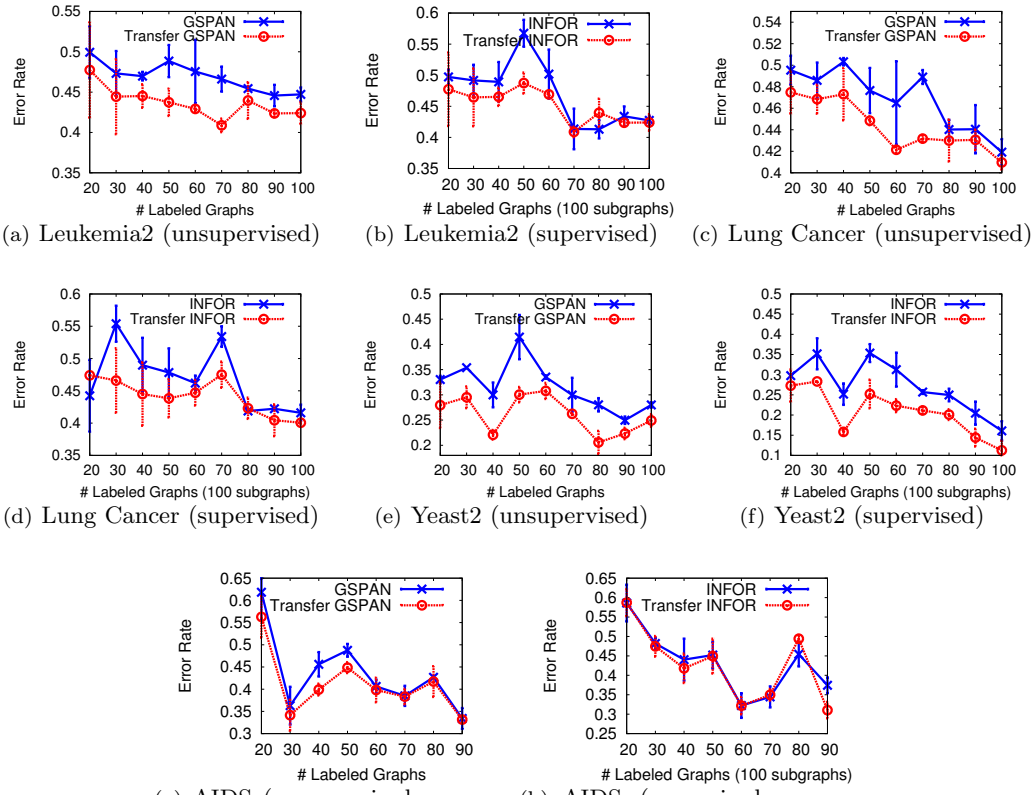
Figure 4: Pool I as target datasets and Pool II as auxiliary sources. Unsupervised means no label graph is used in mining the significant subgraph patterns.

Table 2: Summary of experimental graph datasets (each instance is a graph in SMILES format).

	Graph datasets	#Graph	Function
Pool I:	Breast Cancer	40560	Anti-cancer
	Leukemia	40189	Anti-cancer
	Melanoma	40244	Anti-cancer
	Yeast	10000	Others
Pool II:	Leukemia2	43599	Anti-cancer
	Lung Cancer	40560	Anti-cancer
	Yeast2	8807	Others
	AIDS	10000	AIDS screen data

selected significant subgraphs. No label information was used in unsupervised subgraph pattern mining. For parameter setting, we set the threshold $\lambda = 0.25$ and $l = 3$. Parameter sensitivity is studied in the next section.

Can related sources be identified and help improve the accuracy? Fig. 4 and Fig. 5 provide the results summarized on 10 runs. Note that the error rates on some datasets are larger than 50%. This is because (1) some datasets are imbalanced (the error rate for random guessing is around 0.6); (2) it is known that graph classification is a tough task with low accuracy [15]. However, it can be observed that the proposed transfer learning model can improve the accuracy in most cases. In the comparison with gSpan, the proposed model can reduce the error rates by as much as 33% (in Fig. 4(g) when the number of labeled graphs is only 20). For the comparison with the feature selection algorithm, the proposed method reduces the error rates significantly when there are a limited number of training data. For instance, on yeast dataset, the error rate reduces over 40% when there are only 60 labeled graphs and 9940 unlabeled graphs. This demonstrates the necessity and effectiveness of transfer learning to improve the accuracy when there are only a limited number of labeled data. Note that the error rates on some of the datasets are larger than 50%.



(a) Leukemia2 (unsupervised) (b) Leukemia2 (supervised) (c) Lung Cancer (unsupervised)

(d) Lung Cancer (supervised) (e) Yeast2 (unsupervised) (f) Yeast2 (supervised)

(g) AIDS (unsupervised, no related source, and hence no help to learning) (h) AIDS (supervised, no related source, and hence no help to learning)

Figure 5: Pool II as target datasets and Pool I as auxiliary sources. Unsupervised means no label graph is used in mining the significant subgraph patterns.

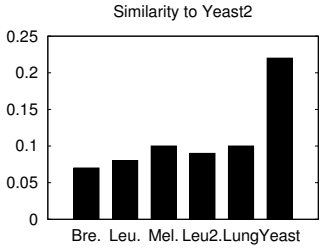
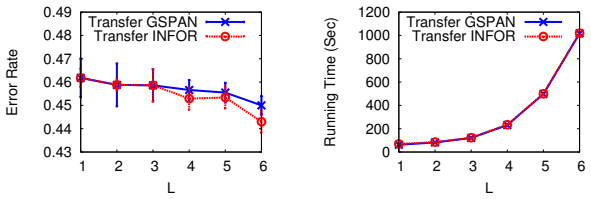


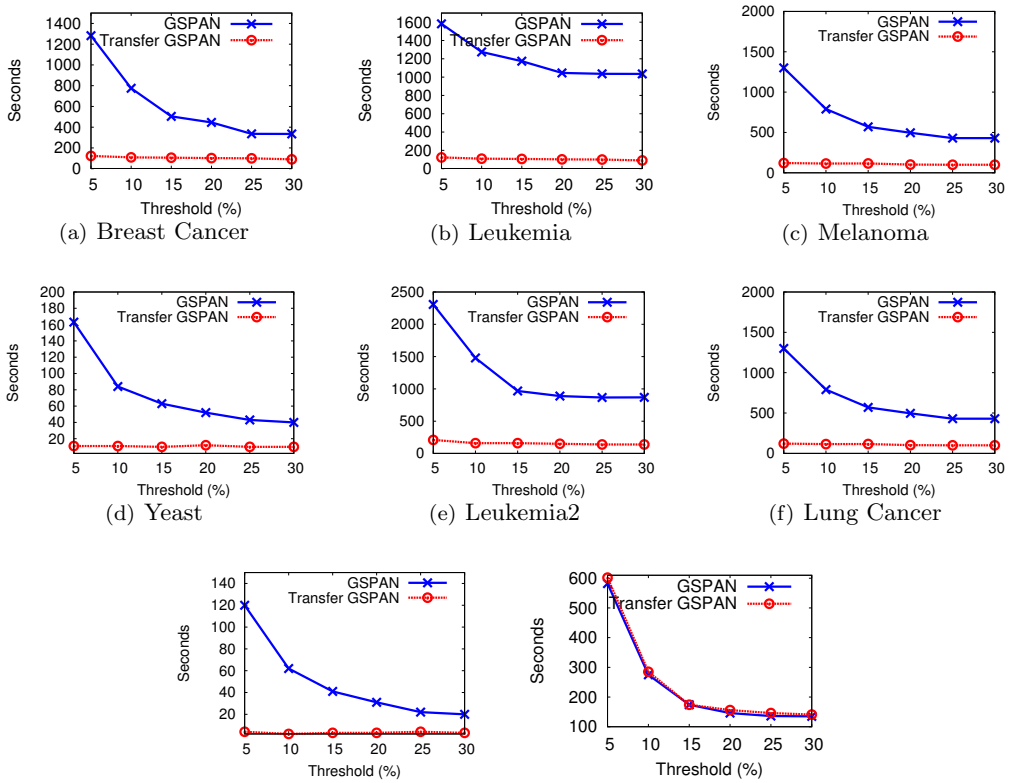
Figure 7: Task similarities reflected by $p(\theta_{\mathcal{S}}|\mathcal{T})$



(a) Error rate with l in calculating KL divergence (b) Running Time with l in calculating KL divergence

Figure 8: Parameter sensitivity

Can the proposed model avoid using unrelated sources? It is important to note that we have a graph dataset on AIDS drug data (Table 2). Because of the uniqueness of the disease, there are no similar drug data that can help identify its significant subgraphs. From Fig. 4 and Fig. 5, we can also observe that the improvement of transfer learning on this dataset is marginal. However, although there is no related source, the learning accuracy is still comparable to the comparison methods. This is because our approach is able to avoid using wrong models by automatically filtering out wrong examples. As such, negative transfer can be avoided. Note that the key component to identify the unrelated sources is the KL divergence discussed in Eq. 3.12. It helps model the similarity between the source and target data $p(\theta_{\mathcal{S}}|\mathcal{T})$. We next present in Fig. 7 the value of $p(\theta_{\mathcal{S}}|\mathcal{T})$ given the "Yeast2" dataset



(a) Breast Cancer (b) Leukemia (c) Melanoma

(d) Yeast (e) Leukemia2 (f) Lung Cancer

(g) Yeast2 (h) Running Time on AIDS (no related source, and hence no help to learning)

Figure 6: Running time comparison

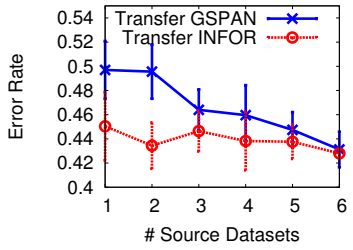


Figure 9: Effect of number of source datasets

as the target task. Recall that a high value of $p(\theta_{\mathcal{S}}|\mathcal{T})$ indicates a strong similarity of the source dataset and the target dataset. We can observe from Fig. 7 that the most similar dataset is the "Yeast" from Pool I. This is because the two datasets are all about graph structure of yeast although they may study on different samples. It shows the effectiveness of the KL divergence to evaluate the task similarities and filter out unrelated datasets.

Can the proposed model improve efficiency? The comparison on the running time is presented in Fig. 6. All algorithms were run on a PC with CPU Duo 2.4G and 3G memory. In this experiment, we set the number of training data as 50, and plot the running time as a function of the threshold λ. Owing to the limited space, we just report the results on "Transfer-GSPAN" but a similar phenomenon can be observed in "Transfer-INFOR". From Fig. 6, it can be observed that in general, the running time decreases with the increasing value of the threshold. Importantly, the transfer learning model is 10 times faster than the comparison methods. For example, in Fig. 6(b), the comparison method needs about 1600 seconds to finish the pattern mining with a threshold of 5%, while the transfer learning model needs only about 160 seconds. This is because related sources help greatly prune the search space by filtering out or deemphasizing the candidate subgraphs that are not significant in the source dataset. However, we can also observe that when there is no related source, the running time of the transfer learning model is almost the same as the comparison method (Fig. 6(h)). The

reason is that when there is no related source, the model in Eq. 3.9 casts into a traditional pattern mining algorithm. But we can conclude that there is almost no loss for the transfer learning model in terms of running time even in the worst case.

4.3 Discussion and Parameter Sensitivity In the above experiments, we provided a pool of source datasets to study whether the proposed algorithms can use the right amount of useful knowledge and avoid those unrelated datasets. Note that this setting is very important in real world practice, and it is desirable to let the learning method to automatically find which datasets are related. Under this experiment setting, an interesting discussion is to investigate whether the result improves with more available sources. We plot Fig. 9 to study the phenomenon. In this experiment, the dataset "Melanoma" was set to be the target dataset, and "Leukemia", "Leukemia 2", "Lung Cancer", "Yeast", "Yeast 2", and "Breast Cancer" were set to be the source datasets. We then changed the number of source datasets to study how it affected the result on the target dataset. In each setting, we first set the number of selected source datasets to be n (n varied from 1 to 6), and we then randomly sampled n datasets from the pool to give the result. This process was performed 10 times for a given n, and we summarized the result in Fig. 9. The other parameter settings were: 50 labeled graphs, threshold $\lambda = 0.25$. It is interesting to see that the error rates of both methods decrease with the increasing number of source datasets, especially the method "transfer-GSPAN". This is because with more source datasets, the proposed Bayesian model obtains more information to infer a more accurate result. This improvement is especially significant for "transfer-GSPAN" because it uses no labeled graph in the target dataset but mainly depends on the supervision knowledge from the sources. Hence, with more sources, it may absorb more useful information to improve the learning accuracy.

When calculating the KL divergence, we use the parameter l to control the maximum number of nodes of each selected subgraph. We plot Fig. 8(a) and Fig. 8(b) to study the effect of different values of l. The "Lung Cancer" dataset was set to be the target dataset, and Pool I was used to be the source datasets. Furthermore, 50 graphs were labeled, and the threshold $\lambda = 0.25$. It can be observed that the error rate decreases a little with increasing l. This is because with a larger l, more frequent subgraphs can be utilized to calculate the KL, and thus the result is more accurate. However, From Fig. 8(b), we can observe that the running time grows exponentially with l. Intuitively, when $l \to \infty$, all the subgraphs are used to calculate the KL, but it takes a long time to finish. In the experiment, we set $l = 3$, which is also the default setting to balance both the accuracy and the efficiency.

5 Related Work

Extracting subgraph features from graph data have been investigated by many researchers. The aim 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 [5]. This algorithm builds a lexicographic order among graphs, and maps each graph to a 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 also been developed, *e.g.* AGM [12], MoFa [13], FFSM [6], and Gaston [16]. Many supervised subgraph feature extraction approaches have also been proposed in literature (*e.g.*, [17], [7], [8]), which look for discriminative subgraph patterns for graph classifications. However, so far as we know, transfer learning technique has not been introduced into the field; but as analyzed in the introduction section, transfer learning can greatly help solve the label deficiency problem in this area.

Another related area of works is transfer learning, which is proposed to help build the target model by extracting knowledge from related source datasets (*e.g.*, [18, 19, 20, 1, 21, 22]). There are various interpretations of transfer learning research. For example, one category of algorithms mainly tackles the problem of different data distributions. A general approach is based on re-sampling (*e.g.*, [19]), where the motivation is to emphasize the examples that are discriminating and similar to the target data. There are also some other solutions such as transfer across feature subspaces (*e.g.*, [23]), transfer across similar learning parameters (*e.g.*, [24]), and the like. It is also used in different kinds of applications such as text mining, web mining (*e.g.*, [3, 2, 25]). Furthermore, transfer learning is applied to unsupervised learning (*e.g.*, [26, 27, 28]). Recently, some works are proposed to transfer knowledge on relational data (*e.g.*, [18, 29, 30]). However, as far as we know, they do not look at graph database whose instances are complex structural graphs.

6 Conclusion

We study transfer learning on graph data, by utilizing the significant subgraphs from auxiliary sources to improve subgraph pattern mining on the target dataset. The key idea is to introduce a latent variable to infer the likelihood of a candidate subgraph to be significant, and assign high weights to the significant subgraphs from related source datasets. Furthermore, KL divergence is modified to evaluate the dataset similarity by estimating the degree they share on the significant subgraphs. The optimization objective is to maximize the likelihood that the selected subgraphs are significant. Experiments show that the proposed transfer learning model can extensively reduce the error rates by as much as 40%, but more importantly, 10 times faster than the comparison models.

Acknowledgements This work is supported in part by NSF through grants IIS 0905215, DBI-0960443, CNS-1115234, IIS-0914934, OISE-1129076, and OIA-0963278, and Google Mobile 2014 Program.

References

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[2] G. Xue, W. Dai, Q. Yang, and Y. Yu, "Topic-bridged plsa for cross-domain text classification," in *SIGIR*, pp. 627–634, 2008.

[3] B. Chen, W. Lam, I. W. Tsang, and T. Wong, "Extracting discriminative concepts for domain adaptation in text mining," in *KDD*, pp. 179–188, 2009.

[4] S. J. Pan, X. Ni, J. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *WWW*, pp. 751–760, 2010.

[5] X. Yan and J. Han, "gSpan: Graph-based substructure pattern mining," in *ICDM*, pp. 721–724, 2002.

[6] J. Huan, W. Wang, and J. Prins, "Efficient mining of frequent subgraphs in the presence of isomorphism," in *ICDM*, pp. 549–552, 2003.

[7] X. Yan, H. Cheng, J. Han, and P. S. Yu, "Mining significant graph patterns by leap search," in *SIGMOD*, pp. 433–444, 2008.

[8] M. Thoma, H. Cheng, A. Gretton, J. Han, H. Kriegel, A. J. Smola, L. Song, P. S. Yu, X. Yan, and K. M. Borgwardt, "Near-optimal supervised feature selection among frequent subgraphs," in *SDM*, pp. 1075–1086, 2009.

[9] H. Cheng, X. Yan, J. Han, and C. Hsu, "Discriminative frequent pattern analysis for effective classification," in *ICDE*, pp. 716–725, 2007.

[10] N. Jin and W. W. 0010, "Lts: Discriminative subgraph mining by learning from search history," in *ICDE*, pp. 207–218, 2011.

[11] X. Shi, W. Fan, Q. Yang, and J. Ren, "Relaxed transfer of different classes via spectral partition," in *ECML/PKDD*, pp. 366–381, 2009.

[12] A. Inokuchi, T. Washio, and H. Motoda, "An apriori-based algorithm for mining frequent substructures from graph data," in *PKDD*, pp. 13–23, 2000.

[13] C. Borgelt and M. R. Berthold, "Mining molecular fragments: Finding relevant substructures of molecules," in *ICDM*, pp. 51–58, 2002.

[14] W. Fan, K. Zhang, H. Cheng, J. Gao, X. Yan, J. Han, P. S. Yu, and O. Verscheure, "Direct mining of discriminative and essential frequent patterns via model-based search tree," in *KDD*, pp. 230–238, 2008.

[15] X. Kong and P. Yu, "Semi-supervised feature selection for graph classification," in *KDD*, 2010.

[16] S. Nijssen and J. N. Kok, "A quickstart in frequent structure mining can make a difference," in *KDD*, pp. 647–652, 2004.

[17] X. Kong, W. Fan, and P. S. Yu, "Dual active feature and sample selection for graph classification," in *KDD*, pp. 654–662, 2011.

[18] J. Davis and P. Domingos, "Deep transfer via second-order markov logic," in *ICML*, p. 28, 2009.

[19] S. Bickel, J. Bogojeska, T. Lengauer, and T. Scheffer, "Multi-task learning for HIV therapy screening," in *ICML*, pp. 56–63, 2008.

[20] X. Shi, Q. Liu, W. Fan, Q. Yang, and P. S. Yu, "Predictive modeling with heterogeneous sources," in *SDM*, pp. 814–825, 2010.

[21] Q. Gu and J. Zhou, "Learning the shared subspace for multi-task clustering and transductive transfer classification," in *ICDM*, pp. 159–168, 2009.

[22] X. Shi, Q. Liu, W. Fan, P. S. Yu, and R. Zhu, "Transfer learning on heterogenous feature spaces via spectral transformation," in *ICDM*, pp. 1049–1054, 2010.

[23] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *NIPS*, pp. 137–144, 2006.

[24] N. D. Lawrence and J. C. Platt, "Learning to learn with the informative vector machine," in *ICML*, 2004.

[25] T. Chen, J. Yan, G. Xue, and Z. Chen, "Transfer learning for behavioral targeting," in *WWW*, pp. 1077–1078, 2010.

[26] S. Ando and E. Suzuki, "Unsupervised cross-domain learning by interaction information co-clustering," in *ICDM*, pp. 13–22, 2008.

[27] J. Zhang and C. Zhang, "Multitask bregman clustering," in *AAAI*, 2010.

[28] S. J. Pan, D. Shen, Q. Yang, and J. T. Kwok, "Transferring localization models across space," in *AAAI*, pp. 1383–1388, 2008.

[29] L. Mihalkova and R. J. Mooney, "Transfer learning from minimal target data by mapping across relational domains," in *IJCAI*, pp. 1163–1168, 2009.

[30] L. Mihalkova, T. N. Huynh, and R. J. Mooney, "Mapping and revising markov logic networks for transfer learning," in *AAAI*, pp. 608–614, 2007.