gRegress: Extracting Features from Graph Transactions for Regression

Abstract

In this work we propose, gRegress, a new algorithm which given set of graph transactions and a real value associated with each graph transaction extracts the complete set of subgraphs such that a) each subgraph in this set has correlation with the real value above a user-specified threshold and b) each subgraph in this set has correlation with any other subgraph in the set below a user-specified threshold. gRegress incorporates novel pruning mechanisms based on correlation of a subgraph feature with the output and correlation with other subgraph features. These pruning mechanisms lead to significant speedup.

Experimental results indicate that in terms of runtime, gRegress substantially outperforms gSpan, often by an order of magnitude while the regression models produced by both approaches have comparable accuracy.

1 Motivation

Regression models are the trusted workhorse for predictive modeling in a variety of application domains. The problem of mining subgraph features from a database of graph transactions for building regression models is critical when an attribute-valued representation is insufficient to capture the domain of study. An example of such a scenario would be the case where we are trying to build a regression model for the toxicity of chemical compounds which is a real value collected from in-vivo experiments. The chemical compounds are represented as graph transactions and the real value of interest associated with each transaction is the toxicity. Is such a scenario, how do we extract relevant features for building a regression model? Currently the state of the art in this regard is the large body of work on the problem of frequent subgraph mining (relevant literature on this topic is reviewed later in the paper). A typical frequent subgraph mining algorithm will mine the complete set of subgraphs with a user defined frequency threshold and these subgraphs can be used as features to build a regression model. Such an approach involving feature extraction using a frequent subgraph mining algorithm has been studied in the context of the graph classification problem and has been applied to the task of classifying chemical compounds [Deshpande *et al.*, 2005] and proteins [Huan *et al.*, 2004] with promising results. However, this approach is plagued with a number of problems which we now illustrate by describing a small case study. The objective of this case study is to motivate our approach and set the stage for the rest of the paper. The case study involves building regression models for predicting the melting point of a set of chemical compounds (details on the data set can be found later in the paper) based solely on subgraph features extracted by the frequent subgraph mining system gSpan using support vector regression. We ran gSpan [Yan and Han, 2002] on the dataset at thresholds ranging from 20% to 5% in 1% decrements with a maximum size of 10. Regression models were built using the feature vectors based on the presence/absence of subgraph features using SVR (Support Vector Regression) [Smola and Schölkopf, 2004], [Cristianini and Shawe-Taylor, 2000] (the particular implementation used was SVMlite [Joachims, 1999]) and were evaluated us-

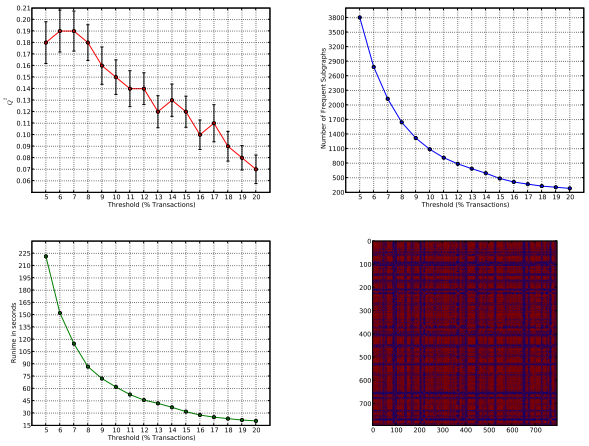

Figure 1: Results of the Case Study: The Q^2 score for the model, the number of subgraphs discovered, the runtimes of gSpan and the absolute pairwise correlations between subgraph features (threshold 10%) for those subgraphs whose absolute correlation with the output is at least 0.20 (out of the total 1100 subgraph features). Pairwise correlation lower than 0.20 is denoted in red while pairwise correlation greater than 0.20 is denoted in blue.

ing the Q^2 (details on the Q^2 metric are found later in the paper) score on a 5-fold cross validation. The Q^2 score for the model, the number of subgraphs discovered and the runtimes of gSpan for each threshold setting are illustrated in Figure 1. We can observe the following.

1. The predictive accuracy of the regression model improves as the threshold frequency reduces. This is an expected result [Deshpande *et al.*, 2005] and has been observed earlier. It can be explained by the fact that additional relevant subgraph features are available on which the model can be based.
2. The number of frequent subgraphs and the runtime also increases as the threshold decreases (as expected and observed earlier [Deshpande *et al.*, 2005]) which in the worst case is expected to grow exponentially.

These observations raise the question on how many of the newly considered subgraph features actually contribute to increased predictive accuracy of the regression model. To answer this question we analyzed the frequent subgraphs generated at the threshold of 10%. Figure 1 shows the absolute pairwise correlations between subgraph features for those subgraphs whose absolute correlation with the output is at least 0.20. Pairwise correlation lower than 0.20 is denoted in blue while pairwise correlation greater than 0.20 is denoted in red. The subgraphs in blue are the ones that contribute most to the predictive accuracy of the regression model based on these thresholds on correlation with the output and the pairwise correlations. While these thresholds are somewhat arbitrary, they do give a certain measure of the redundancy of the subgraphs generated. Typically, feature selection for building regression models considers the trade off between how much a feature correlates with the output and how much the feature correlates with the features already selected. Our claim is that mining features based on their frequency produces useful features but also produces additional redundant features at an added cost. Of course, redundant features could be eliminated by a simple post processing step but this is computationally expensive as the redundant subgraphs are still generated in the first place. We should prefer to mine for a set of subgraphs such that each member of this set has high correlation with the output value and that the members of this set have low correlation with each other. Mining a complete set of subgraphs based on two thresholds, correlation with the output and correlation with other features, is an intuitive approach for building regression models and, as this work will show, is also computationally efficient. This brings us to key contributions of this work:

1. For a given subgraph feature, we prove an upper bound on the correlation with the output that can be achieved by any supergraph for this subgraph feature.
2. For a given subgraph feature, we prove an lower bound on the correlation that can be achieved by any supergraph for this subgraph feature with any other subgraph feature.
3. Using these two bounds we design a new algorithm called gRegress, which extracts the complete set of subgraphs such that a) each subgraph in this set has correlation with the real value above a user-specified threshold and b) each subgraph has correlation with any other subgraph in the set below a user-specified threshold.
4. We conduct an experimental validation on a number of real-world datasets showing that in terms of runtime, gRegress substantially outperforms gSpan, often by an order of magnitude while the regression models produced by both approaches have comparable accuracy.

2 Problem Formulation

Our graphs are defined as $G = (V, E, L, \mathcal{L})$, where V is the set of vertices, $E \subseteq V \times V$ is a set of edges, L is the set of labels and $\mathcal{L}$ is the labelling function $\mathcal{L} : V \cup E \rightarrow L$. The notions of subgraph (denoted by $G \subseteq G'$), supergraph, graph isomorphism (denoted $G = G'$) and subgraph isomorphism in the case of labelled graphs are intuitively similar to the notions of simple graphs with the additional condition that the labels on the vertexes and edges should match. Our examples consist of pairs,

$$E = \{< x_1, y_1 >, < x_2, y_2 >, ..., < x_n, y_n >\}$$

where x_i is a labelled graph and $y_i \in \mathbb{R}$ and is assumed to be centered, that is, $\sum_i y_i = 0$. We define the set $\mathcal{S}$ to contain contains every distinct subgraph of every graph in E. For any subgraph feature $g \subseteq x$ we define,

$$h_g(x) = \begin{cases} 1, & \text{if } g \subseteq x \\ -1, & \text{otherwise} \end{cases}$$

We define the the indicator function $I(x) = y$.The absolute correlation of a subgraph feature g_i with the output is given by,

$$\rho_{g_i,I} = \left| \frac{h_{g_i} \cdot I}{\|h_{g_i}\| \, \|I\|} \right|$$

The absolute correlation of a a subgraph feature g_i with another subgraph feature g_j is given by,

$$\rho_{g_i,g_j} = \left| \frac{h_{g_i} \cdot h_{g_j}}{\|h_{g_i}\| \, \|h_{g_j}\|} \right|$$

We can now define the problem as follows.

Given:

1. A set of examples E,
2. A threshold on the correlation with the output $\alpha \in \mathbb{R}$, $0 \leq \alpha \leq 1$
3. A threshold on the correlation between subgraph features $\beta \in \mathbb{R}$, $0 \leq \alpha \leq 1$

Find: A maximal set $H = \{g_1, g_2, ..., g_k\}$ such that,

1. For each $g_i \in H$,

$$\rho_{g_i,I} = \left| \frac{h_{g_i} \cdot I}{\|h_{g_i}\| \, \|I\|} \right| \geq \alpha$$

2. For any $g_i, g_j \in H$,

$$\rho_{g_i,g_j} = \left| \frac{h_{g_i} \cdot h_{g_j}}{\|h_{g_i}\| \left\|h_{g_j}\right\|} \right| \leq \beta$$

We now discuss why it makes intuitive sense to mine for the set H. First, note that the formulation is in terms of absolute correlations. This is simply because we are interested in mining subgraph features with high correlations either positive or negative. Negative correlation, implying that the absence of a subgraph correlates with the output is equivalent to positive correlation as the regression model will simply learn negative weights for such a feature. Next, note that the set H is the maximal or the largest possible set of subgraphs such that a) each subgraph in this set has correlation with the real value above a user-specified threshold α and b) each subgraph has correlation with any other subgraph in the set below a user-specified threshold β. Feature selection for building regression models considers the trade off between how much a feature correlates with the output and how much the feature correlates with the features already selected. The problem definition intuitively captures this trade off.

3 Proposed Algorithm: gRegress

Given the formulation of the problem in the previous section a naive solution would be an algorithm that searches the complete space of subgraph features (of the graph transitions) checking for each each subgraph feature conditions (1) and (2) retaining only those subgraph features that satisfy all of them. Of course, one of the many canonical labeling schemes introduced in frequent subgraph mining systems could be incorporated to prevent the generation of duplicate subgraph features (relevant literature on this topic is reviewed later in the paper).

The critical problem here is determining pruning conditions corresponding to the frequency antimonotone pruning condition used by all frequent subgraph mining systems. The frequency antimonotone pruning condition is a simple observation that if a subgraph feature has frequency that the user specified threshold, no supergraph of this subgraph can be frequent given this threshold. This simple observation allows for massive pruning of the search space in the case of frequent subgraph mining.

Thus the key problem is to answer the following two questions.

1. Given a subgraph feature, what is the highest possible correlation any supergraph of this subgraph feature can achieve with the output?
2. Given a subgraph feature what is the lowest possible correlation any supergraph of this subgraph feature can achieve with some other subgraph feature?

It must be noted that once we have a quantitative measure for questions (1) and (2) it becomes very easy to adapt any frequent subgraph mining system to solve the problem at hand. Quantitative measures for (1) and (2) in a sense correspond the frequency antimonotone condition in the case of frequent subgraph mining.

For a graph g we define

$$m_g(x) = \begin{cases} -1, & \text{if } I(x) \leq 0 \\ h_g(x), & \text{if } I(x) > 0 \end{cases}$$

We have the following upper bound on the correlation any supergraph of a subgraph feature can achieve with the output.

For some subgraph features g_i and g_j if $g_j \subseteq g_i$, then,

$$\rho_{g_j,I} = \left| \frac{h_{g_j} \cdot I}{\left\|h_{g_j}\right\| \|I\|} \right| \leq \left| \frac{m_{g_i} \cdot I}{\|m_{g_i}\| \|I\|} \right|$$

Proof. It is easy to see that for any subgraph feature say g_i if $h_{g_i}(x) = -1$ then for no subgraph feature $g_i \subseteq g_j$, $h_{g_j}(x) = 1$. That is, all those x such that $h_{g_i}(x) = -1$, for any $g_j \subseteq g_j$ $h_{g_j}(x) = -1$. Furthermore, only for those x where $h_{g_i}(x) = 1$ can $h_{g_j}(x) = -1$ for some $g_i \subseteq g_j$. The highest possible $\rho_{g_j,I}$ can occur in the case where for all x such that $I(x) \leq 0$ $h_{g_j}(x) = -1$. The result follows. □

For a graph g we define

$$n_g(x) = \begin{cases} -1, & \text{if } h_g(x) > 0 \\ h_g, & \text{if } h_g(x) \leq 0 \end{cases}$$

We have the following lower bound on the correlation any supergraph of a subgraph feature can achieve with some other subgraph feature.

For some subgraph features g_i, g_j and g_k, if $g_j \subseteq g_i$, then,

$$\rho_{g_j,g_k} = \left| \frac{h_{g_j} \cdot h_{g_k}}{\left\|h_{g_j}\right\| \|h_{g_k}\|} \right| \geq \left| \frac{n_{g_i} \cdot h_{g_k}}{\|n_{g_i}\| \|h_{g_k}\|} \right|$$

Proof. As before is easy to see that for any subgraph feature say g_i if $h_{g_i}(x) = -1$ then for no subgraph feature $g_i \subseteq g_k$, $h_{g_k}(x) = 1$. That is, all those x such that $h_{g_i}(x) = -1$, for any $g_i \subseteq g_k$ $h_{g_k}(x) = -1$. Furthermore, only for those x where $h_{g_i}(x) = 1$ can $h_{g_k}(x) = -1$ for some $g_i \subseteq g_k$. The lowest possible ρ_{g_j,g_k} can occur in the case where for all x such that $h_{g_i}(x) > 0$ $h_{g_k}(x) = -1$. The result follows. □

Using these bounds it is now possible to adapt any subgraph enumeration scheme to the task defined earlier. In particular, we adopt the DFS search and DFS canonical labelling used by gSpan. The key steps of our algorithm, which we refer to as gRegress, are summarized in Algorithm 3.1 and Procedure 3.2.

Algorithm 3.1 gRegress(E,α,β,S)

1: $H \leftarrow \emptyset$
2: $P \leftarrow$ DFS codes of 1-vertex subgraphs in E
3: **for all** g_i such that $g_i \in P$ **do:**
4: Extend(E, α, β, S, H, g_i)
5: **return** H

Procedure 3.2 Extend(E, α, β, S, H, g_i)

1: **if** g not minimum DFS code **:**
2: **return**
3: **if** $\left| \frac{h_{g_i} \cdot I}{\|h_{g_i}\| \|I\|} \right| \leq \alpha$ **:**
4: **return**
5: **for all** g_j such that $g_j \in H$**:**
6: **if** $\left| \frac{h_{g_i} \cdot h_{g_j}}{\|h_{g_i}\| \|h_{g_j}\|} \right| \geq \beta$ **:**
7: **return**
8: $H \leftarrow H \cup g_i$
9: $P \leftarrow$ DFS codes of rightmost extensions of g_i
10: **for all** g_k such that $g_k \in P$ **:**
11: **if** $\left| \frac{m_{g_i} \cdot I}{\|m_{g_i}\| \|I\|} \right| \geq \alpha$**:**
12: **for every** $g_j \in H$ **if** $\left| \frac{n_{g_j} \cdot h_{g_k}}{\|n_{g_j}\| \|h_{g_k}\|} \right| \leq \beta$ **:**
13: Extend(E, α, β, S, H, g_k)

4 Experimental Evaluation

Our experimental evaluation of the proposed gRegress algorithm seeks to answer the following questions.

1. How do the subgraph features extracted by gRegress compare with frequent subgraph mining algorithms with respect to predictive accuracy of the regression model developed based on these features?
2. How does the gRegress algorithm compare with frequent subgraph mining algorithms in terms of runtime when applied to the task of feature extraction for building regression models?
3. How does the runtime gRegress algorithm vary for various choices of α and β parameters?

4.1 Selecting Data Sets and Choice of Representation

In order to answer these questions we collected a number of data sets, the general properties of which are presented in Table **??**. All the data sets are publicly available and are from the domain of computational chemistry. They consist of chemical compounds with a specific property of interest associated with each compound. In every case we use a simple graph representation for the chemical compounds with element symbols as vertex labels and bond types as edge labels. The value for which the regression model was to be built was centered to have a mean of zero. No information other than the subgraph features are used to build the regression models for any experiments reported in the paper.

The Karthikeyan [Karthikeyan *et al.*, 2005] data set, was originally collected to build a regression model for the prediction of melting points of chemical compounds. The Bergstrom [Bergstrom *et al.*, 2003] data set was originally collected to develop a model for the melting point of solid drugs. The Huuskonen [Huuskonen, 2000] data set was originally collected to develop a model for the aqueous solubility of chemical compounds. The Delaney [Delaney, 2004] data set was originally collected to develop a model for the aqueous solubility of compounds with low molecular weights. The ERBD (Estrogen Receptor Binding Dataset) [Tong *et al.*, 2002] was originally collected for developing predictive models of the estrogen receptor binding activity. The ARBD (Androgen Receptor Binding Data) [Blair *et al.*, 2000], [Branham *et al.*, 2002] was originally collected for developing predictive models of the androgen receptor binding activity.

4.2 Selecting a Frequent Subgraph Mining System

Among the various frequent subgraph mining systems to compare gRegress with, we chose gSpan. While it is unclear whether gSpan is the best frequent subgraph mining [Worlein *et al.*, 2005], [Nijssen and Kok,] (relevant literature on this topic is reviewed later in the paper) it can definitely be considered to be among the state of the art as far as the frequent subgraph mining problem is concerned. In order to ensure that our results generalize to frequent subgraph mining algorithms in general, we compare the number of subgraphs considered by both gRegress and gSpan. This is simply a count of all minimal DFS codes considered by each of the systems. The difference between the number of minimal DFS codes considered by gSpan and gRegress gives us a measure of how gRegress compares with any other frequent subgraph mining system. This is because different frequent subgraph mining systems may use other forms of canonical labelling and search mechanisms will prevent the generation of duplicate subgraph features better than gSpan and gRegress but every subgraph feature (the minimal code in the case of gSpan and gRegress) will have to be considered at least once. If gRegress considers significantly fewer subgraphs, the speedup in terms of runtime would most likely apply to other frequent subgraph mining systems also.

4.3 Selecting a Regression Algorithm

Among the various approaches to regression we chose SVR (Support Vector Regression) [Smola and Schölkopf, 2004], [Cristianini and Shawe-Taylor, 2000] which can be can be considered among the state of the art as far as the regression problem is concerned. In particular, we use the SVM-Lite [Joachims, 1999] package. While it is possible that in certain situations other regression algorithms might outperform SVR, we find it unlikely to get opposite results while comparing the quality of the regression models based on the subgraph features produced by gSpan and gRegress with any regression algorithm. But in future work we will consider other regression methods.

4.4 The Q^2 score

We use the Q^2 score to evaluate the predictive accuracy of the regression models. The Q^2 score for a regression function f is defined as follows.

$$Q^2 = \frac{\sum_{i=1}^{n}(y_i - f(x_i))^2}{\sum_{i=1}^{n}(y_i - \frac{1}{n}\sum_{i=1}^{n} y_i)^2}$$

Note that the Q^2 score is a real number between 0 and 1 and its interpretation is similar to the Pearson correlation coefficient. The closer it is to 1, the better the regression function fits the testing data.

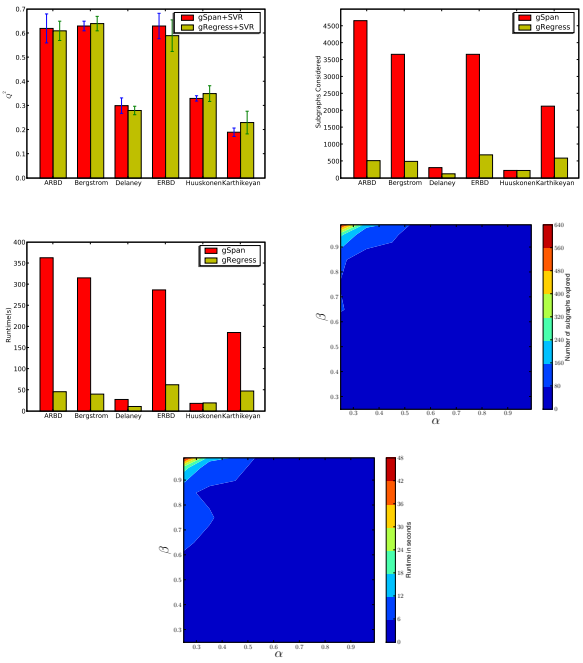

Figure 2: Results of the Experiments: Q^2 scores, number of subgraphs considered, running times and contour plots for number of subgraphs considered and running times for gRegress on the Karthikeyan data set for varying α and β parameters

4.5 Experiments

In order to answer question (1) and (2) we conducted experiments on gSpan and gRegress on the six data sets described above. The subgraph features produced by each algorithm were used to build a regression model using SVR. The predictive accuracy of the models was evaluated based on the Q^2 score using a 5-fold cross validation. Additionally the runtimes and the number of subgraphs considered by each algorithm was also recorded. In the case of gRegress the selection function chose the subgraph feature with the higher correlation. The maximum subgraph size for each system was set to ten. The parameters of each system (threshold frequency in the case of gSpan and the α and β parameters in the case of gRegress) were systematically varied. While comparing results on the various runs of the algorithms, we select the significantly highest Q^2 scores achieved by each system and then compare the lowest possible runtimes and the subgraphs considered for this Q^2 score. The intuition behind this to compare the lowest computational cost for the best possible predictive accuracy. The results of these experiments are reported in Figure 2.

In order to answer question (3) we ran gRegress on the Karthikeyan data set (we chose this data set as this was the largest data set in terms of transactions) with α and β parameters systematically varied in small increments of 0.05. Figure 2 illustrates these results with contour plots.

4.6 Observations

We can observe the following from the experimental results.

1. The predictive accuracy of the regression models based on the features generated by gSpan and gRegress is comparable.
2. gRegress substantially outperforms gSpan in terms of runtime and the number of subgraphs explored.
3. The runtime and the number of subgraphs explored by gRegress increases for small values of α and large values of β.

5 Related Work

The field of graph mining or the discovery of interesting patterns from structured data represented as graphs has been extensively researched and good surveys can be found in [Washio and Motoda, 2003] and [Cook and Holder, 2006]. Here, due to space limitations, we only present a concise overview.

The problem of developing regression models from graph transactions is relatively new as compared to the related problem of graph classification. Graph classification was first studied in [Gonzalez *et al.*, 2000]. A variety of approaches to the task have been proposed since. This includes, using a frequent subgraph mining system to extract subgraph features and using these features to train a classifier based on support vector machines [Deshpande *et al.*, 2005] [Huan *et al.*, 2004], graph kernels [Gartner *et al.*, 2003] [Kashima *et al.*, 2003] and application of boosting [Kudo *et al.*, 2005] to graph classification. The boosting based approach has also been recently extended to perform regression [Saigo *et al.*, 2008]. In this work the authors also propose an approach based on partial least square regression. Work related to the task of developing regression models from graph transactions also includes [Ke *et al.*, 2007] in which the authors investigate the search from subgraphs with high correlation from a database of graph transactions.

6 Conclusions and Future Work

The findings of this work are as follows.

1. Mining features from graph transactions for building regression models based on their frequency produces useful features but also produces additional redundant features at an added cost.
2. Features can be mined based on two thresholds, correlation with the output and correlation with other features, at a significantly lower cost without sacrificing the predictive accuracy of the regression model.

The work raises the following questions which we plan to investigate as a part of our future work.

1. How can the relation between the α and β parameters and the predictive accuracy of the regression model be characterized?
2. How to systematically select the α and β parameters to get the best regression model?
3. How does the choice of the selection function affect the regression model?

References

[Bergstrom *et al.*, 2003] C.A.S. Bergstrom, U. Norinder, K. Luthman, and P. Artursson. Molecular descriptors influencing melting point and their role in classification of solid drugs. *Journal of Chemical Information and Computer Sciences*, 43(4):1177–1185, 2003.

[Blair *et al.*, 2000] R.M. Blair, H. Fang, W.S. Branham, B.S. Hass, S.L. Dial, C.L. Moland, W. Tong, L. Shi, R. Perkins, and D.M. Sheehan. The Estrogen Receptor Relative Binding Affinities of 188 Natural and Xenochemicals: Structural Diversity of Ligands. *Toxicological Sciences*, 54(1):138–153, 2000.

[Branham *et al.*, 2002] W.S. Branham, S.L. Dial, C.L. Moland, B.S. Hass, R.M. Blair, H. Fang, L. Shi, W. Tong, R.G. Perkins, and D.M. Sheehan. Phytoestrogens and Mycoestrogens Bind to the Rat Uterine Estrogen Receptor 1. *Journal of Nutrition*, 132(4):658–664, 2002.

[Cook and Holder, 2006] D.J. Cook and L.B. Holder. *Mining Graph Data*. John Wiley & Sons, 2006.

[Cristianini and Shawe-Taylor, 2000] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[Delaney, 2004] J.S. Delaney. Esol: Estimating aqueous solubility directly from molecular structure. *Journal of Chemical Information and Computer Sciences*, 44(3):1000–1005, 2004.

[Deshpande *et al.*, 2005] M. Deshpande, M. Kuramochi, N. Wale, and G. Karypis. Frequent Substructure-Based Approaches for Classifying Chemical Compounds. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, pages 1036–1050, 2005.

[Gartner *et al.*, 2003] T. Gartner, P. Flach, and S. Wrobel. On Graph Kernels: Hardness Results and Efficient Alternatives. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 129–143, 2003.

[Gonzalez *et al.*, 2000] J.A. Gonzalez, L.B. Holder, and D.J. Cook. Graph Based Concept Learning. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pages 1072–1072. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2000.

[Huan *et al.*, 2004] J. Huan, W. Wang, D. Bandyopadhyay, J. Snoeyink, J. Prins, and A. Tropsha. Mining protein family specific residue packing patterns from protein structure graphs. In *Proceedings of the eighth annual international conference on Resaerch in computational molecular biology*, pages 308–315. ACM New York, NY, USA, 2004.

[Huuskonen, 2000] J. Huuskonen. Estimation of aqueous solubility for a diverse set of organic compounds based on molecular topology. *Journal of Chemical Information and Computer Sciences*, 40(3):773–777, 2000.

[Joachims, 1999] T. Joachims. Making large-scale support vector machine learning practical. 1999.

[Karthikeyan *et al.*, 2005] M. Karthikeyan, R.C. Glen, and A. Bender. General melting point prediction based on a diverse compound data set and artificial neural networks. *Journal of Chemical Information and Modeling*, 45(3):581–590, 2005.

[Kashima *et al.*, 2003] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized Kernels Between Labeled Graphs. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, volume 20, page 321, 2003.

[Ke *et al.*, 2007] Y. Ke, J. Cheng, and W. Ng. Correlation search in graph databases. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 390–399. ACM Press New York, NY, USA, 2007.

[Kudo *et al.*, 2005] T. Kudo, E. Maeda, and Y. Matsumoto. An application of boosting to graph classification. *Advances in neural information processing systems*, 17:729–736, 2005.

[Nijssen and Kok,] S. Nijssen and J.N. Kok. Frequent Subgraph Miners: Runtimes Dont Say Everything. *MLG 2006*.

[Saigo *et al.*, 2008] Hiroto Saigo, Nicole Krämer, and Koji Tsuda. Partial least squares regression for graph mining. In *KDD*, pages 578–586, 2008.

[Smola and Schölkopf, 2004] A.J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[Tong *et al.*, 2002] W. Tong, R. Perkins, H. Fang, H. Hong, Q. Xie, SW Branham, D. Sheehan, and J. Anson. Development of quantitative structure-activity relationships (QSARs) and their use for priority setting in the testing strategy of endocrine disruptors. *Regul Res Perspect*, 1(3):1–16, 2002.

[Washio and Motoda, 2003] T. Washio and H. Motoda. State of the art of graph-based data mining. *ACM SIGKDD Explorations Newsletter*, 5(1):59–68, 2003.

[Worlein *et al.*, 2005] M. Worlein, T. Meinl, I. Fischer, and M. Philippsen. A Quantitative Comparison of the Subgraph Miners MoFa, gSpan, FFSM, and Gaston. *LECTURE NOTES IN COMPUTER SCIENCE*, 3721:392, 2005.

[Yan and Han, 2002] X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, page 721, 2002.