Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 02-027

Automated Approaches for Classifying Structures

Mukund Deshpande, Michihiro Kuramochi, and George Karypis

June 26, 2002

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
26 JUN 2002		**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Automated Approaches for Classifying Structures	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Army Research Office,PO Box 12211,Research Triangle Park,NC,27709-2211**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**16**	

Automated approaches for classifying structures*

Mukund Deshpande, Michihiro Kuramochi and George Karypis

University of Minnesota, Department of Computer Science/Army HPC Research Center
Minneapolis, MN 55455
Technical Report #02-027
{deshpand,kuram,karypis}@cs.umn.edu

Abstract

In this paper we study the problem of classifying chemical compound datasets. We present an algorithm that first mines the chemical compound dataset to discover discriminating sub-structures; these discriminating sub-structures are used as features to build a powerful classifier. The advantage of our classification technique is that it requires very little domain knowledge and can easily handle large chemical datasets. We evaluated the performance of our classifier on two widely available chemical compound datasets and have found it to give good results.

Keywords: Classification, Graphs, Chemical Structures, SVM

1 Introduction

There is a great need to develop reliable computational techniques (*in silico*), based on classification, that can quickly screen thousands of compounds and identify the compounds that display the highest levels of the desired property. For example, identifying potentially toxic compounds or compounds that can inhibit the active sites of viruses in the early phase of drug discovery is becoming an appealing strategy with pharmaceutical companies [9]. However individually determining the suitability of a chemical compound to a particular biological state is not a very viable solution—mainly for two reasons. First, the number of chemical compounds in the repository and those that can be generated by combinatorial chemistry is extremely large. Second, experimentally determining the suitability of a compound using bio-assays (*in vivo*) techniques is an expensive and time consuming process.

One of the key challenges in developing classification techniques for chemical compounds stems from the fact that their properties are strongly related to their chemical structure. However, traditional machine learning techniques are suited to handle datasets represented by multidimensional vectors or sequences, and cannot handle the relational nature of the chemical structures.

In recent years a number of techniques have been proposed for classifying chemical compounds. These techniques can be broadly categorized into two groups. The first group consists of techniques that rely mainly on various global properties of the chemical compounds, such as molecular weight, ionization potential, inter-atomic distance *etc*., for capturing the structural properties of the compounds. Since this information is not relational, existing classification

*This work was supported by NSF CCR-9972519, EIA-9986042, ACI-9982274, by Army Research Office contract DA/DAAG55-98-1-0441, by the DOE ASCI program, and by Army High Performance Computing Research Center contract number DAAH04-95-C-0008. Access to computing facilities was provided by AHPCRC, Minnesota Supercomputer Institute. Related papers are available via WWW at URL: **http://www.cs.umn.edu/~karypis**

techniques can be easily used on these datasets. However, the absence of actual structural information limits the accuracy of such classifiers [17]. The second group of techniques directly analyze the structure of the chemical compounds to identify patterns that can be used for classification [6, 19, 11, 10, 13, 2]. One of the earliest studies in discovering sub-structures was carried out by Dehaspe *et al* [6], in which Inductive Logic Programming(ILP) techniques were used, though this approach is quite powerful it is not designed to scale to large graph databases hence may not be able to handle large chemical compound databases. A number of recent approaches focuses on analyzing the (geometric) graph representation of the chemical compounds, to identify frequently occurring patterns, and use these patterns to aid in the classification. Wang *et al*[19] developed an algorithm to find frequently occurring *blocks* in the geometric representation of protein molecules and showed that these blocks can be used for classification. Inikuchi *et al* [11] developed an algorithm to find all frequently occurring *induced* subgraphs and presented some evidence that such subgraphs can be used to features for future classification. Similarly, Gonzalez *et al*[10] used the SUBDUE [4] system to find frequently occurring approximate graphs; however, they reported no result regarding the success of using these subgraphs for classification. Finally, Kramer *et al*[13] used the SMILES [5] representation of the chemical compounds to find a restricted set of molecular fragments and showed that even that restricted set of sub-structures was able to provide some differentiation between the various classes.

Our work builds upon the earlier research on using substructure discovery methods to aid in the classification of chemical compounds and extends it in two significant directions. First, it uses the most generic description of a substructure—that of a connected subgraph, and employs a highly efficient frequent subgraph discovery algorithm to find the complete set of subgraphs that occur in a non-trivial fraction of the compounds. Second, it uses these structures to develop general purpose classifiers using state-of-the-art machine learning techniques such as rules and support vector machines. Key advantages of our approach are that it is generic, requires very little domain knowledge, and can easily scale to extremely large datasets.

We evaluated the performance of our classification approach using two publicly available datasets. The first dataset was published by the National Toxicity Program of the U.S. National Institute for Environmental Health Sciences, it contains the bio-assays of different chemical compounds on rodents to study the carcinogencity (cancer inducing) of the compounds. The ultimate goal being to estimate the carcinogenicity of different compounds on humans. The second dataset is published by the National Cancer Institute and contains the results of AIDS anti-viral screen, that was developed to discover new compounds capable of inhibiting the HIV virus. The AIDS anti-viral screen uses a soluble formazan assay to measure the protection of human CEM cells from HIV-1 infection [8]. The goal of the first dataset is to discover carcinogenic compounds, whereas for the second dataset the goal is to discover compounds that inhibit the HIV virus.

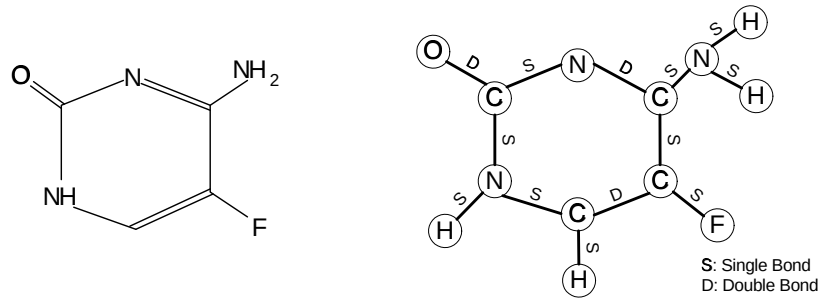
Our experimental results showed that our classifier outperforms the contestants of Predictive Toxicology Challenge [17] on two out of four models, and on the AIDS anti-viral screen dataset the classifier has a high true positive rate.

The rest of the paper is organized as follows, in Section 2 we discuss some background on chemical compounds and we formulate the chemical compound classification problem and present the terminology. In Section 3 we discuss in the detail the classification methodology, followed by Section 4 that experimentally evaluates the classification schemes. Finally, in Section 5 we present the conclusions and future work.

2 Problem Formulation & Terminology

The focus of this paper is to develop classification techniques that will correctly classify datasets consisting of chemical compounds. The input to this problem is a set of chemical compounds with known class labels and the goal is to build a model that can correctly predict the classes of previously unclassified compounds. The meaning of the various classes is application dependent. In some applications, the classes will capture the extent to which a particular compound is toxic, whereas in other applications they may capture the extent to which a compound can inhibit (or enhance) a particular factor and/or active site.

In most applications each of the compounds is assigned to only one of two classes, that are commonly referred to as the *positive* and *negative* class. The positive class corresponds to compounds that exhibit the property in question,



(a) NSC 103025 Flucytosine (b) Graph Representation

Figure 1: Chemical and Graphical representation of Flucytosine

whereas the compounds of the negative class do not. It is quite expensive and time consuming to experimentally determine the class of the various compounds; thus, the development of accurate classification techniques can greatly improve the speed and efficiency of the overall process.

Nature of Chemical Datasets A chemical compound consists of different atoms being held together via bonds adopting a well-defined geometric configuration. Figure 1(a) represents the chemical compound Flucytosine from the DTP AIDS repository [8] it consists of a central benzene ring and other elements like N, O and F.

There are many different ways to represent such chemical compounds. The simplest representation is the molecular formula that lists the various molecules making up the compound; the molecular formula for Flucytosine is `C4H4FN3O`. A more sophisticated representation can be achieved using the SMILES [5] representation, it not only represents the atoms but also represents the bonds between different atoms. The SMILES representation for Flucytosine is `Nc1nc(O)ncc1F`.

In addition to these topological representations, the actual 3D coordinates of the various atoms can also be supplied. However, these coordinates are hard to obtain experimentally and in some cases not feasible as the actual compounds correspond to not yet synthesized chemical molecules. For this reason, coordinate information is usually obtained via molecular dynamic simulations that try to find the lowest energy conformation, and may be inaccurate.

Graph Representation of Chemical Compounds The activity of a compound largely depends on its chemical structure. As a result, effective classification algorithms must be able to directly take into account the structural nature of these datasets.

In this paper, in order to facilitate this requirement, we represent each compound as undirected (topological) graphs[1]. The vertices of these graphs correspond to the various atoms, and the edges correspond to the bonds between the atoms. Each one of the vertices and edges of has a label associated with it. The labels on the vertices correspond to the type of atoms, and the labels on the edges correspond to the type of bond. The graph representation of Flucytosine is shown in Figure 1(b).

Formally, a topological graph $G = (V, E)$ is made of two sets, the set of vertices V and the set of edges E. Each edge itself is a pair of vertices. Throughout this paper we assume that the graph is undirected, *i.e.*, each edge is an unordered pair of vertices. Furthermore, we will assume that the graph is *labeled*. That is, each vertex and edge has a label associated with it that is drawn from a predefined set of vertex-labels (L_V) and edge-labels (L_E). Each vertex (or edge) of the graph is not required to have a unique label and the same label can be assigned to many vertices (or edges) in the same graph. For example, in Figure 1 the vertex-label **C** appears four times in the graph. Through-out the paper we assume that each chemical compound has been transformed into its topological graph representation.

Also, we will be using the notions of connected (induced) sub-graphs which are briefly defined as follows. Given a graph $G = (V, E)$, a graph $G_s = (V_s, E_s)$ will be a *subgraph* of G if and only if $V_s \subseteq V$ and $E_s \subseteq E$. Also, G_s will

[1]Note that for the purpose of this study we do not consider any available geometric information, but we plan to explore its utility in later studies.

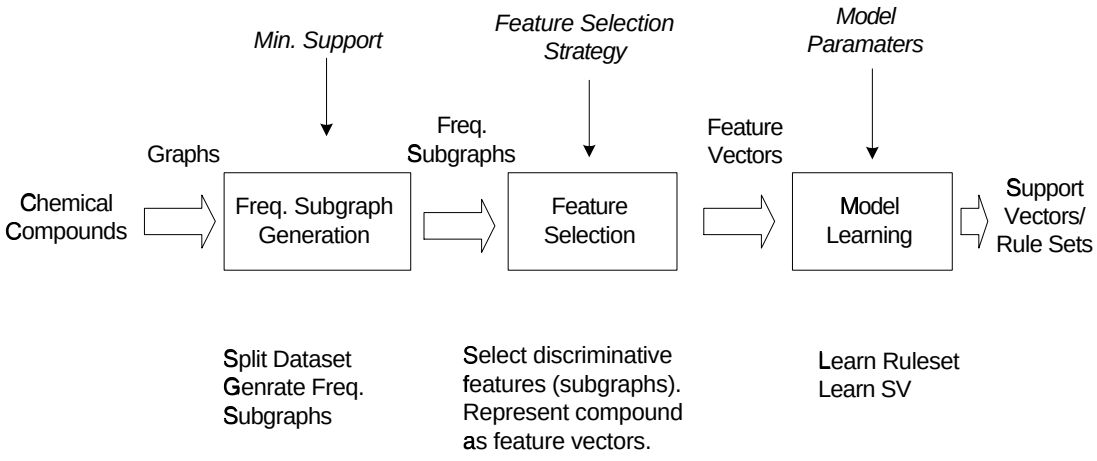


Figure 2: Graph Classification Flowchart

be an *induced subgraph* of G if $V_s \subseteq V$ and E_s contains all the edges of E that connect vertices in V_s. Another way to say is that the subgraph G_s is contained in the graph G.

3 Classification Methodology

The overall outline of our approach is shown in Figure 2. Initially, each of the chemical compounds is represented using a graph (following the model described in Section 2). Then, a frequent subgraph discovery algorithm is used to mine these graphs and find all subgraphs that occur in a non-trivial fraction of the compounds. Each of these subgraphs then becomes a candidate feature, and various feature selection algorithms are used to obtain a smaller set of discriminatory and non-redundant features. The remaining subgraphs are then used to create a feature space such that each of the selected feature corresponds to a dimension, and each compound is represented as a boolean vector based on the particular set of features (*i.e.*, subgraphs) that it contains. Finally, these feature vector representation of the chemical compounds are supplied into the classifier to learn the appropriate classification model.

Our approach shares some of the characteristics with earlier approaches for chemical compound classification based on substructure discovery [19, 2], but it has a number of inherent advantages. First, unlike previous approaches that used restricted definitions of what constitutes a substructure (*e.g.*, induced subgraphs [11], or blocks [19]), we use the most generic model possible that of connected subgraphs. Moreover, unlike the approaches based on approximate matching [4], the structure discovery algorithm that we use is guaranteed to find all subgraphs that satisfy the user-supplied minimum support constraint. Consequently, our approach can be used to mine chemical compounds with very little domain knowledge. Second, our approach decouples the processes of feature discovery from the actual classification process, and can be used with any existing (and future developed) classification algorithm. Third, since the classifier is built on the transformed feature-space, the above approach can easily incorporate any additional non-graphical features that may be available for the compounds such as molecular weight, surface area, *etc*., by simply adding them as additional dimensions in the feature-space.

In the rest of this section we describe in detail the approaches and algorithms that we used for finding frequent subgraphs and building the classification models. Due to space constraints we did not include any discussion about feature selection. This task is to a large extent domain independent and the reader should refer to [7] for a discussion of the various options.

3.1 Discovery of Frequent Subgraphs

To mine the database of chemical compounds and discover the sub-structures that occur frequently in them, we used the frequent subgraph discovery algorithm, called FSG, that was recently developed by members of our group [14].

FSG takes as input a database D of graph transactions that are specified using the graph model described in Section 2, and a minimum support σ, and finds all connected subgraphs that occur in at least σ% of the transactions.

There are three important aspects of the subgraphs discovered by FSG. The first has to do with the fact that we are only interested in subgraphs that are connected. This is motivated by the fact that the resulting frequent subgraphs will be encapsulating a chemical-substructure and connectivity is a natural property of such patterns. The second has to do with the fact that as discussed in Section 2 we use labeled graphs, and each graph (and discovered pattern) can contain vertices and/or edges with the same label. This greatly increases our modeling ability, as it allow us to find patterns containing multiple occurrences of the same atom(s) and bond(s). Finally, the third has to do with the fact that we are only interested in subgraphs that occur in at least σ% of the transactions. This ensures that the discovered subgraphs are statistically significant and not spurious. Furthermore, this minimum support constraint also helps in making the problem of frequent subgraph discovery computationally tractable.

FSG's frequent subgraph discovery strategy uses the same level-by-level approach adopted in the well-known Apriori [1] algorithm for finding frequent itemsets. The high level structure of the algorithm is shown in Algorithm 3.1. Edges in the algorithm correspond to items in traditional frequent itemset discovery. As these apriori based algorithms increase the size of frequent itemsets by adding a single item at a time, our algorithm increases the size of frequent subgraphs by adding an edge one-by-one. FSG initially enumerates all the frequent single and double edge graphs. Then, based on those two sets, it starts the main computational loop. During each iteration it first generates candidate subgraphs whose size is greater than the previous frequent ones by one edge (Line 5 of Algorithm 3.1). Next, it counts the frequency for each of these candidates, and prunes subgraphs that do no satisfy the support constrain (Lines 7–11). Discovered frequent subgraphs satisfy the downward closure property of the support condition, which allows us to effectively prune the lattice of frequent subgraphs.

Algorithm 3.1 fsg(D, σ) (Frequent Subgraph)

1: $F^1 \leftarrow$ detect all frequent 1-subgraphs in D
2: $F^2 \leftarrow$ detect all frequent 2-subgraphs in D
3: $k \leftarrow 3$
4: **while** $F^{k-1} \neq \emptyset$ **do**
5: $\quad C^k \leftarrow$ fsg-gen(F^{k-1})
6: $\quad$ **for each** candidate $g^k \in C^k$ **do**
7: $\quad\quad g^k$.count $\leftarrow 0$
8: $\quad\quad$ **for each** transaction $t \in D$ **do**
9: $\quad\quad\quad$ **if** candidate g^k is included in transaction t **then**
10: $\quad\quad\quad\quad g^k$.count $\leftarrow g^k$.count $+ 1$
11: $\quad F^k \leftarrow \{g^k \in C^k \mid g^k.\text{count} \geq \sigma D\}$
12: $\quad k \leftarrow k + 1$
13: **return** $F^1, F^2, \ldots, F^{k-2}$

Notation	Description
D	A dataset of graph transactions
t	A transaction of a graph in D
k-(sub)graph	A (sub)graph with k edges
g^k	A k-subgraph
C^k	A set of candidates with k edges
F^k	A set of frequent k-subgraphs

Detailed experimental evaluation of FSG presented in [14] showed that FSG was able to scale to large graph datasets and low support values. The key to FSG's computational scalability lies on a highly efficient graph canonical labeling algorithm that it employs, which allows it to uniquely identify the various generated subgraphs without having to resort to computationally expensive graph and subgraph isomorphism. Furthermore, FSG uses a TID-list based approach for frequency counting that besides speeding up the counting of the support of the various subgraphs it can also be used to compute the feature-based representation of each compound in our classification framework.

Even though FSG provides the general functionality required to find all frequently occurring sub-structures in chemical datasets, there are a number of issues that need to be addressed before it can be applied as a black-box tools for feature discovery in the context of classification. Two of these issues are explored in the rest of this section.

Handling Classes of Different Sizes The only user-supplied parameter in the FSG algorithm is the value of the minimum support σ that is used to prune infrequent patterns. The overall success of the proposed approach is highly dependent on selecting the right value for this parameter. If the minimum support is set too low, the computational complexity of the frequent subgraph discovery process will increase substantially (in some cases the problem will become intractable) and will lead to a large number of frequent subgraphs—potentially confusing certain classes of classification algorithms. On the other hand, if the minimum support is set too high, FSG may unnecessarily prune

some subgraphs that have good discriminating ability and are critical for the correct classification of the dataset.

The support sensitivity becomes even worse in the cases in which the dataset contain classes of significantly different sizes. In such cases, in order to ensure that FSG is able to find features that are meaningful for all the classes, it must use a support that depends on the size of the smaller class. Failure to do so, may lead to a situation in which the selected support value is greater than the size of the smallest class. However, such a low value of support will most likely generate a lot of poor subgraphs for the largest classes.

For this reason we first partition the complete dataset, using the class label of the examples, into specific class specific datasets. We then run FSG on each of these *class datasets*. This partitioning of the dataset ensures that sufficient subgraphs are discovered for those class labels which occur rarely in the dataset. Next, we combine subgraphs discovered from each of the *class dataset*. After this step each subgraph has a vector that contains the frequency with which it occurs in each class.

Focusing on the Positive Class Traditional classification problems assume that there are well-defined positive and negative classes. However, in many problem instances involving chemical compounds and their relation to certain chemical properties and interactions (*e.g.*, toxicity, binding strength, *etc.*), the negative class is not well-defined or we do not have complete information about it. Furthermore, quite often instead of a binary classification system we have a continuous system indicating the strength of the desired property.

In such cases, a better classification can be obtained by using the FSG algorithm to find patterns only in chemical compounds of the positive class, and ignore any patterns that were found only in the negative class. Once these subgraphs have been discovered, they can then be used as features to describe both positive and negative instances as before. The motivation behind this *feature selection* approach is that a structure occurring only in the negative class may not necessarily be a good indicator of the absence of the desired property, and may have occurred due to improper sampling of the positive class. Also, an additional advantage of this *positive-only* subgraph discovery, is that it greatly improves the computational complexity of our approach since we do not need to mine the much larger set of negative instances.

3.2 Learning Classification Models

Given the frequent subgraphs discovered in the previous step, our algorithm treats each of these subgraphs as a feature and represents the chemical compound as a boolean vector. The ith entry of this vector will be one if the compound's graph contains the ith subgraph, otherwise it will be zero. This mapping into the feature space of frequent subgraphs is performed both for the training and the test dataset[2]. The mapping of the training set can potentially be done efficiently by using the TID-lists for each discovered subgraph that is outputed by FSG. However, the mapping of the test set requires that we check each frequent subgraph against the graph of the test compound using subgraph isomorphism. Fortunately, the overall process can be substantially speeded up by taking into account the frequent subgraph lattice that is also generated by FSG. In this case, we traverse the lattice from top to bottom and only visit the child nodes of a subgraph if that subgraph is isomorphic to the chemical compound.

Once the feature vectors for each chemical compound have been build, any one of the existing classification algorithms can potentially be used for classification. However, the characteristics of the transformed dataset and the nature of the classification problem itself tends to limit the applicability of certain classes of classification algorithms. In particular, the transformed dataset will most likely be high dimensional, and second, it will be sparse, in the sense that each compound will have only a few of these features, and each feature will be present in only a few of the compounds. Moreover, in most cases the positive class will be much smaller than the negative class, making it unsuitable for classifiers that primarily focus on optimizing the overall classification accuracy.

In our study we used two classes of classification algorithms, one uses classification rules and the other uses support vector machines, that we believe are well-suited for operating in such sparse and high-dimensional datasets and at the same time can be tuned to focus on the small positive class. The details of these algorithms and how they were used

[2]Note that the frequent subgraphs were identified by mining *only* the graphs of the chemical compounds in the training set.

for the problem of classifying chemical compounds are described in the rest of this section.

3.2.1 Classification Rules

Our first classification method uses single features as Classification Rules (CR). The primary motivation for using such a simple scheme was the fact that (i) it allowed us to evaluate the class discriminating ability of the discovered frequent subgraphs, and (ii) such a scheme has been shown to be well-suited for problems with small positive classes [12].

For each feature i discovered by FSG, we compute its confidence to the positive class, as follows. Let n_i^+ and n_i^- be the number of times the ith feature occurs in the transactions of the positive and negative class, respectively. Then, the *confidence* of the ith feature to the positive class c_i^+ is defined as $n_i^+/(n_i^+ + n_i^-)$. Note that the confidence will be a number between zero and one. A value of zero means that this feature never occurs in the positive class whereas a value of one indicates that this feature only occurs in the positive class.

Now, for each example j in the test-set let S_j be the set of features that it supports. Now, our algorithm assigns j to the positive class based on the confidence to the positive class of the rules in S_j. In particular, we implemented two different schemes. The first scheme identifies the feature in S_j that has the highest confidence in the positive class and assigns j to the positive class as long as this ***maximum confidence*** is above a certain threshold. The technique is inspired from Holte's 1R methods [3]. The second scheme looks at the ***average confidence*** of all the features in S_j, and again assigns j to the positive class if this average confidence to the positive class is above a certain threshold.

Note that the value of the threshold controls true positive rate of the classifier and should be chosen depending on the cost model of the classifier.

3.2.2 Support Vector Machines

Support vector machines is a state-of-the-art classification technique based on pioneering work done by Vapnik *et al*, [18]. This algorithm is introduced to solve two-class pattern recognition problems using the Structural Risk Minimization principle [18]. Given a training set in a vector space, this method finds the *best* decision hyperplane that separates two classes. The quality of a decision hyperplane is determined by the distance (referred as margin) between two hyperplanes that are parallel to the decision hyperplane and touch the closest data points of each class. The *best* decision hyperplane is the one with the maximum margin. By defining the hyperplane in this fashion, SVM is able to generalize to unseen instances quite effectively. The SVM problem can be solved using quadratic programming techniques [18]. SVM extends its applicability on the linearly non-separable data sets by either using soft margin hyperplanes, or by mapping the original data vectors into a higher dimensional space in which the data points are linearly separable. The mapping to higher dimensional spaces is done using appropriate kernel functions, resulting in efficient algorithms. A new example is classified by representing the point the feature space and computing its distance from the hyperplane.

One of the advantages of using SVM is that it allows us to directly control the cost associated with the miss-classification of examples from the different classes [15]. This allow us to associate a higher cost for the miss-classification of positive instances; thus, biasing the classifier to learn a model that tries to increase the true-positive rate, at the expense of increasing the false positive rate.

4 Experimental Evaluation

In this section we experimentally evaluate our approach for classifying chemical compounds on two publicly available chemical compound datasets.

Toxicology Dataset (PTC) This dataset was first used as a part of the Predictive Toxicology Evaluation Challenge [17] which was organized as a part of PKDD/ECML 2001 Conference[3]. It contains data published by the U.S. National Institute for Environmental Health Sciences, the data consists of bio-assays of different chemical compounds on rodents to study the carcinogenicity (cancer inducing) properties of the compounds [17]. The goal being to estimate

[3] $http://www.informatik.uni-freiburg.de/ml/ptc/$

the carcinogenicity of different compounds on humans. Each compound is evaluated on four kinds of laboratory animals (*male Mice, female Mice, male Rats, female Rats*), and is assigned four class labels each indicating the toxicity of the compound for that animal. There are four classification problems one corresponding to each of the rodents and will be referred as *MM, FM, MR* and *FR*.

AIDS Dataset The second dataset is obtained from the National Cancer Institute's DTP AIDS Anti-viral Screen program [8, 13] [4]. Each compound in the dataset is evaluated for evidence of anti-HIV activity. The screen utilizes a soluble formazan assay to measure protection of human CEM cells from HIV-1 infection [20]. Compounds able to provide at least 50% protection to the CEM cells were re-tested. Compounds that provided at least 50% protection on retest were listed as *moderately active* (CM, confirmed moderately active). Compounds that reproducibly provided 100% protection were listed as *confirmed active* (CA). Compounds neither active nor moderately active were listed as *confirmed inactive* (CI). We have formulated three classification problems on this dataset, in the first problem we consider only *confirmed active* (CA) and *moderately active* (CM) compounds and then build a classifier to separate these two compounds; this problem is referred as *CA/CM*. For the second problem we combine *moderately active* (CM) and *confirmed active* (CA) compounds to form one set of *active* compounds, we then build a classifier to separate these *active* and *confirmed inactive* compounds; this problem is referred as *(CA+CM)/CI*. In the last problem we only use *confirmed active* (CA) and *confirmed inactive* compounds and build a classifier to categorize these two compounds; this problem is referred as *CA/CI*.

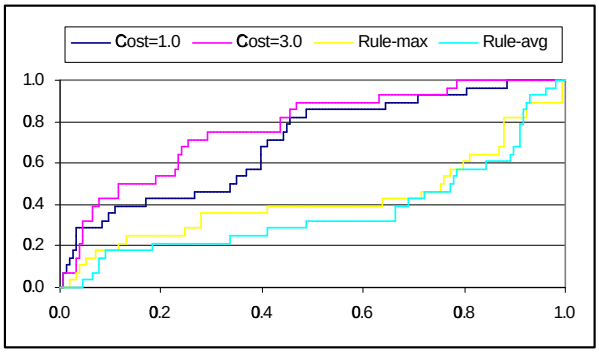
Dataset characteristics Table 1 displays some important characteristics of these two datasets. It is worth noting that DTP-AIDS is an extremely large dataset containing well over 40,000 compounds, with each compound having on an average 45 vertices. The right hand side of the table displays the class distribution for different classification problems, for each problem the table displays the percentage of positive class found in the dataset for that classification problem.

Table 1: Dataset Characteristics

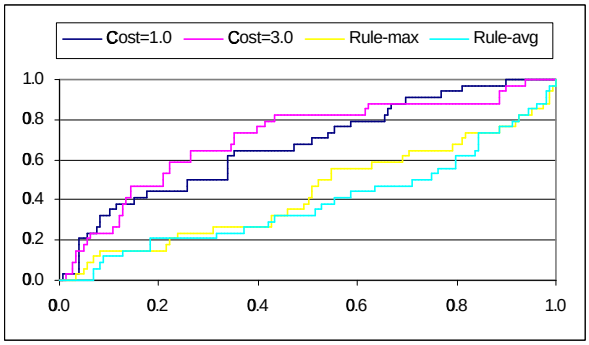
	Toxic	Aids	*Class Dist.* (% +ve class)	
Number of compounds	417	42,687	**Toxicology**	
Avg. Number of vertices	25	46	Male Mice	38.3%
Avg. Number of edges	26	48	Female Mice	40.9%
Number vertex labels	24	85	Male Rats	44.2%
Number edge labels	4	3	Female Rats	34.4%
Max. Number vertices	106	438	**DTP AIDS**	
Min. Number vertices	2	2	CA/CM	28.1%
			(CA+CM)/CI	3.5%
			CA/CI	1.0%

Evaluation Methodology Since the classification problem on both the dataset is cost sensitive *i.e.*, the misclassification cost for each class label could be different, using *accuracy* to judge a classifier would be incorrect. To get a better understanding of the classifier performance for different cost settings we plot the ROC curve [16] for each classifier. ROC curve plots the false positive rate (X axis) versus the true positive rate (Y axis) of a classier; it displays the performance of the classifier without regard to class distribution or error cost. Two classifiers are compared by comparing their ROC curves, if the ROC curve of one classifier completely subsumes the other ROC curve, then that classifier has superior performance for any cost model of the problem. Another qualitative way to compare the performance is to compare the area under the ROC curve for each classifier. Since it is infeasible to plot ROC curves for all the classifiers we tabulate the area under the curve for each experiment and wherever possible display the ROC curve for the classifier.

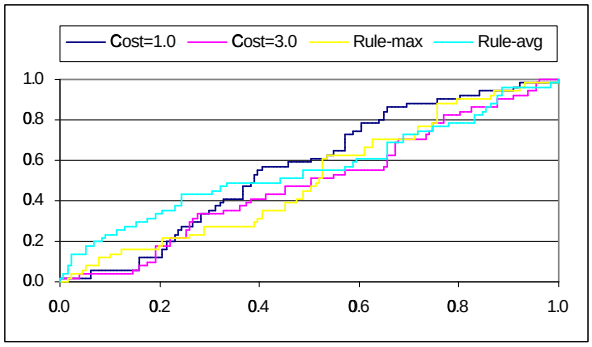
[4] http://dtp.nci.nih.gov/docs/aids/aids_data.html



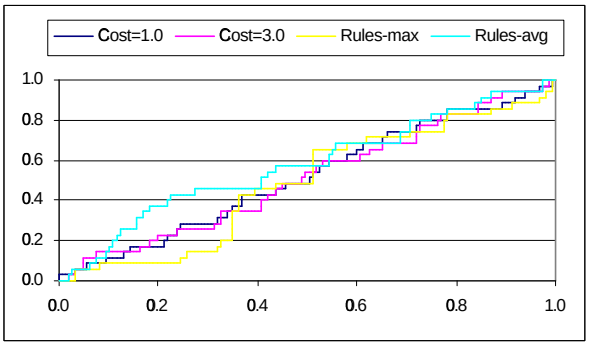
(a) Male Mice ROC for SVM & Rule Classifier



(b) Female Mice ROC for SVM and Rule Classifier



(c) MaleRats ROC for SVM & Rule Classifier



(d) Female Rats ROC for SVM & Rule Classifier

Figure 3: ROC Curves for Different Classification Problems in Toxicology Dataset

4.1 Evaluation on Toxicology Dataset

In this section we conduct three sets of experiments on the Toxicology dataset. For each set of experiments we use the two classifiers, SVM and classification rules for evaluating the performance.

Varying the Misclassification Cost The first set of experiments were conducted by changing the misclassification cost in the SVM classifier so as to associate higher misclassification cost for incorrectly classifying positive examples. The results of the experiments using SVM and classification rules classifier are displayed in Table 2. Each cell in the table indicates the area under the ROC curve for that classifier and misclassification cost value. The actual ROC curves for some classifiers are plotted in Figure 3.

Table 2: Results on the Toxicology Dataset

Support Vector Machines				
Mis. Cost	*MM*	*FM*	*MR*	*FR*
1.0	0.6992	0.6729	**0.5650**	0.5170
1.5	0.7333	0.7073	0.4956	0.5147
2.0	0.7528	**0.7206**	0.4833	0.5199
3.0	**0.7723**	0.7103	0.4883	0.5108
4.0	0.7590	0.6763	0.4860	0.5537
5.0	0.7314	0.6640	0.4887	**0.5747**
Classification Rules				
Order Crit.	*MM*	*FM*	*MR*	*FR*
Max. Conf.	0.4242	0.4343	0.5071	0.4869
Avg. Conf.	0.3634	0.3847	0.5586	0.5771

From the results we see that the misclassification cost parameter does influence the performance of the classifier (area under the curve). However the optimal misclassification cost that achieves maximum area under the curve is

different for each classification problem. This could be explained by the differences in the class distribution for each of the four classification problems. Also, if we compare the performance of SVM with the classification rules classifier, we observe that SVM outperforms classification rules on *male Mice* and *female Mice* dataset, and achieves comparable performance for the *male Rats* and *female Rats* dataset.

Support Sensitivity We performed a set of experiments to evaluate the sensitivity of the proposed approaches on the value of support threshold (σ) used to find frequent structures. We experimented with support thresholds of 3.0%, 5.0%, 7.0%*and*10.0%. These results for the SVM and the classification rule based schemes are shown in Table 3. The various SVM results were obtained using a single misclassification cost value of 3.0.

Table 3: Support Sensitivity on Toxicology Dataset

Classifier	*Support Threshold = 3.0%*			
	MM	*FM*	*MR*	*FR*
SVM	0.7520	0.6649	0.5253	0.5848
CR Max. Conf.	0.6428	0.5459	0.4985	0.4734
CR Avg. Conf.	0.6486	0.5928	0.4828	0.5129
Classifier	*Support Threshold = 5.0%*			
	MM	*FM*	*MR*	*FR*
SVM	0.7723	0.7103	0.4883	0.5108
CR Max. Conf.	0.4242	0.4343	0.5071	0.4869
CR Avg. Conf.	0.3634	0.3847	0.5586	0.5771
Classifier	*Support Threshold = 7.0%*			
	MM	*FM*	*MR*	*FR*
SVM	0.7516	0.7264	0.5260	0.4900
CR Max. Conf.	0.4000	0.4479	0.5079	0.4785
CR Avg. Conf.	0.3541	0.3663	0.5595	0.5876
Classifier	*Support Threshold = 10.0%*			
	MM	*FM*	*MR*	*FR*
SVM	0.7124	0.5827	0.5887	0.4657
CR Max. Conf.	0.4404	0.5403	0.5032	0.5166
CR Avg. Conf.	0.3476	0.4386	0.5396	0.5646

The performance of all the classifiers is significantly influenced by the support threshold used during frequent subgraph generation. However, the relationship is not uniform across the different classification problems. We observe that on *female Rats* classification problem we get better performance at lower values of support threshold, whereas the performance on the *male Rats* problem improves as the value of support threshold is increased. This variation in performance of different classification models is in line with the observations made in [17]. This behavior suggests that a more sophisticated feature selection strategy, instead of using the support threshold to select features, will lead to even better performance.

Focusing on Positive Class Most of the chemical structure datasets have a skewed class distribution with very few examples from the positive class. One way to increase the importance to features belonging to positive class is to run the FSG algorithm only on the example of positive class, as outlined in Section 3.1. Since there are fewer positive class examples, focusing only on the positive class drastically cuts down the time required to build the classifier. The results of our experiments are show in Table 4. We experimented with different support thresholds while keeping misclassification cost fixed at 3.0.

The difference in performance, when subgraphs are discovered only in the positive class versus when they are discovered in the whole dataset, is very small for SVM & maximum confidence rule classifiers. However, the average confidence rule classifier displays some improvement.

Comparison with Predictive Toxicology Challenge Contestants Figure 4 compares the performance of our classifiers with those submitted by the PTC contestants. Each plot contains contains one curve for SVM classifier, two curves for the Classification Rules classifier and about thirty points, each point representing the classification model submitted by contestants of the Predictive Toxicology Challenge.

Observing the plots we can deduce that on *male Mice* and *female Mice* problems our SVM classifier is superior to

Table 4: Focusing on Positive Class Only

Classifier	*Support Threshold = 5.0%*			
	MM	*FM*	*MR*	*FR*
SVM	0.7505	0.7130	0.5055	0.4851
CR Max. Conf.	0.4260	0.4300	0.5052	0.4873
CR Avg. Conf.	0.3952	0.3897	0.5665	0.6002
Classifier	*Support Threshold = 7.0%*			
	MM	*FM*	*MR*	*FR*
SVM	0.7115	0.7327	0.5219	0.4787
CR Max. Conf.	0.4010	0.4469	0.5070	0.4783
CR Avg. Conf.	0.3915	0.3796	0.5749	0.6080
Classifier	*Support Threshold = 10.0%*			
	MM	*FM*	*MR*	*FR*
SVM	0.6776	0.5670	0.5334	0.4995
CR Max. Conf.	0.4462	0.5397	0.5044	0.5166
CR Avg. Conf.	0.3970	0.4960	0.5469	0.5502

all the classifier submitted by the contestants, on the other hand for *male Rats* and *female Rats* classification problems a handful of contestants have their ROC points outside our ROC curve. It should be noted that our classification scheme is based solely on the features obtained from chemical structure and does not make use of any additional domain specific features that can only improve the performance.

4.2 Evaluation on the DTP-AIDS Dataset

Table 5 displays the results of our two classifier on the DTP-AIDS dataset. Since the class distribution for the two classification problems on the AIDS dataset is quite different we have experimented with different sets of misclassification costs for each dataset. Figure 5 displays the ROC curve for some of the classifiers shown in Table 5.

Table 5: Results on the DTP-AIDS Dataset

Support Vector Machines					
Mis. Cost	*CA/CM*	*Mis. Cost*	*(CA+CM)/CI*	*Mis. Cost*	*CA/CI*
1.0	0.7740	1.0	0.7420	1.0	0.86831
1.5	0.7802	1.5	0.7504	1.5	0.86761
2.0	0.7860	15.0	0.7864	15.0	0.90233
2.5	0.7816	35.0	0.7783	35.0	0.90972
3.0	0.7841	50.0	0.7731	50.0	0.91220
15.0	0.7566	100.0	0.7468	100.0	0.91379
Classification Rules					
Order Crit.	*CA/CM*	*Order Crit.*	*(CA+CM)/CI*	*Order Crit.*	*CA/CI*
Max. Conf.	0.6702	Max. Conf.	0.6310	Max. Conf.	0.69735
Avg. Conf.	0.6611	Avg. Conf.	0.6017	Avg. Conf.	0.67681

We find that the results of the SVM classifier on the AIDS dataset are significantly better that those obtained by the Classification Rules scheme. The misclassification cost parameter used in SVM classifier does influence the performance of the classifier, especially the shape of the ROC curve; with each problem having a different optimal value of the misclassification cost. Comparing the performance across the three classification problems, we find that the performance on the *CA/CI* problem is better than performance on the problems *CA/CM* and *CA+CM/CI*.

High Confidence Subgraphs We now present frequent subgraphs that have high confidence on the positive class. Figure 6 displays the top 5 highest confidence subgraphs discovered for three classification problems *CA/CM*, *(CA+CM)/CI* and *CA/CI*.

5 Conclusion & Future Work

In this paper we present an automated scheme for classifying chemical compounds. The highlights of our scheme are that it requires very little domain knowledge and can easily handle large chemical compound datasets. We have evaluated our scheme on two datasets and we show that for some classification problems our classifier outperforms

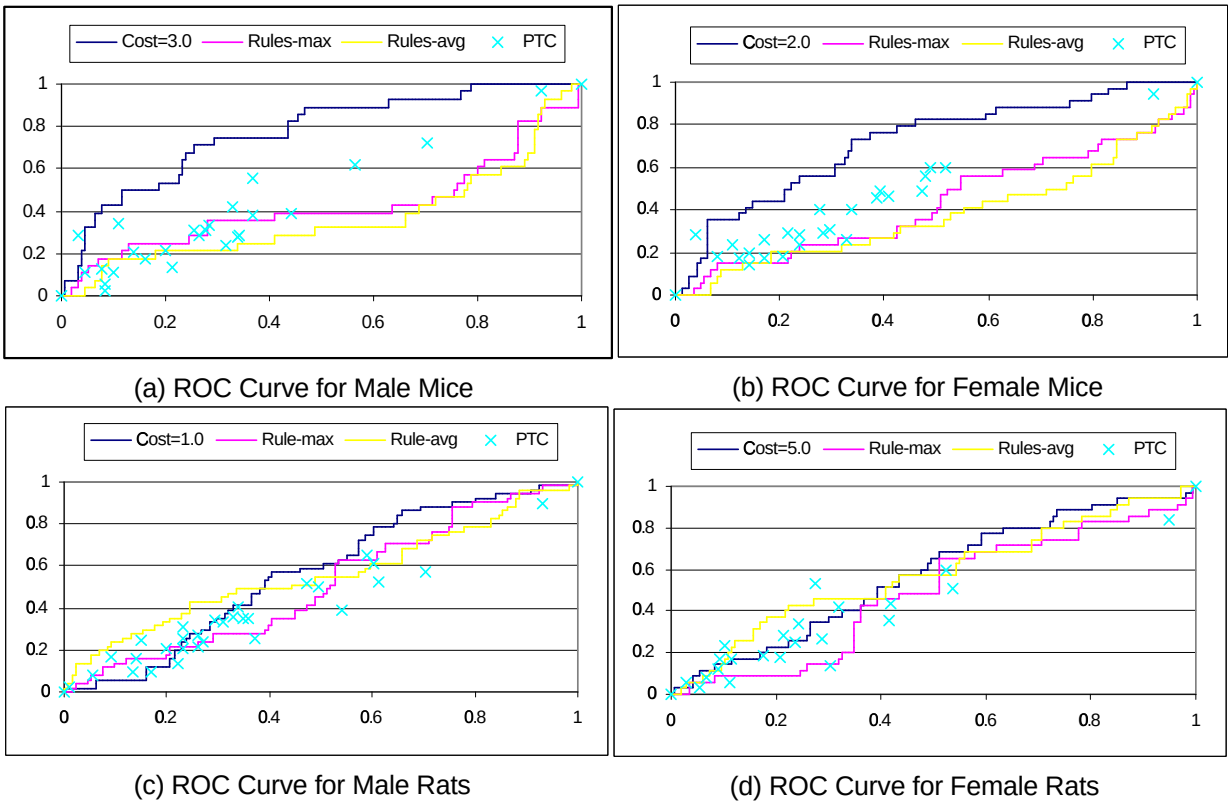


Figure 4: Comparison of graph classification schemes with other contestants of PTC, each plot displays the result for SVM and two Classification rule based schemes.

the contestants of Predictive Toxicology Challenge.

We feel that since the performance of the classifier varies a lot with respect to the support threshold, and to some extent how frequent subgraphs are discovered, a sophisticated feature selection scheme can lead to substantial improvements in the performance. Another way to improve the performance of the classifier is to extend the feature space with well known non-graphical features.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of the 20th Int. Conf. on Very Large Databases (VLDB)*, 1994.

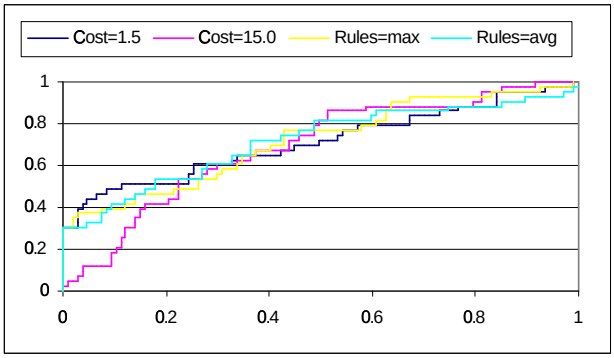
[2] A. An and Y. Wang. Comparisons of classification methods for screening potential compounds. In *IEEE International Conference on Data Mining*, 2001.

[3] H. R. C. Very simple classification rules perfrom well on most commonly used datasets. *Machine Learning*, 1993.

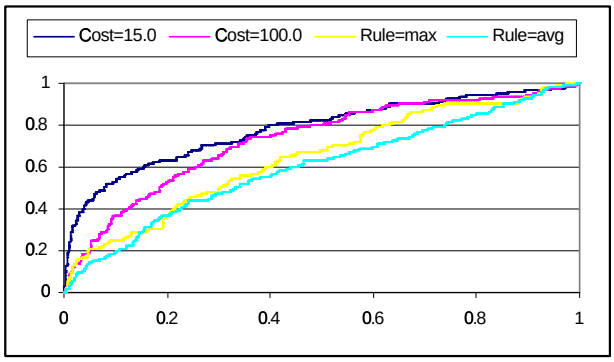
[4] R. N. Chittimoori, L. B. Holder, and D. J. Cook. Applying the subdue substructure discovery system to the chemical toxicity domain. In *Proceedings of the Florida AI Research Symposium*, 1999.

[5] W. D. Smiles 1. introduction and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28, 1988.

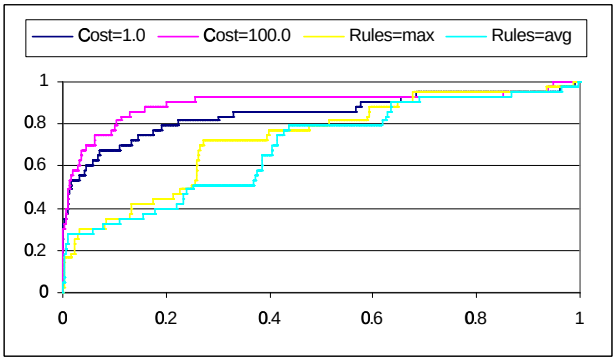
[6] L. Dehaspe, H. Toivonen, and R. D. King. Finding frequent substructures in chemical compounds. In *4th International Conference on Knowledge Discovery and Data Mining*, 1998.



(a) ROC for AIDS -- Active (CA)/ Mod. Active(CM)



(b) ROC for AIDS -- Active(CA+CM) / Inactive(CI)



(c) ROC for AIDS -- Active(CA) / Inactive(CI)

Figure 5: ROC curves for *CA/CM* and *(CA+CM)/CI* classification problems.

[7] M. Deshpande and G. Karypis. Using conjunction of attribute values for classification. Technical report, University of Minnesota, Dept. of Computer Science, 2002.

[8] dtp.nci.nih.gov. Dtp aids antiviral screen dataset.

[9] S. K. Durham and G. M. Pearl. Computational methods to predict drug safety liabilities. *Current Opinion in Drug Discovery & Development*, 2001.

[10] J. Gonzalez, L. Holder, and D. Cook. Applicatio of graph based concept learning to the predictive toxicology domain. In *Predictive Toxicology Challenge, Workshop at the 5th PKDD*, 2001.

[11] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In 4^{th} *PKDD*, 2000.

[12] M. Joshi, R. Agarwal, and V. Kumar. Mining needles in haystack: Classification of rare classes via two-phase rule induction. In *ACM SIGMOD*, 2001.

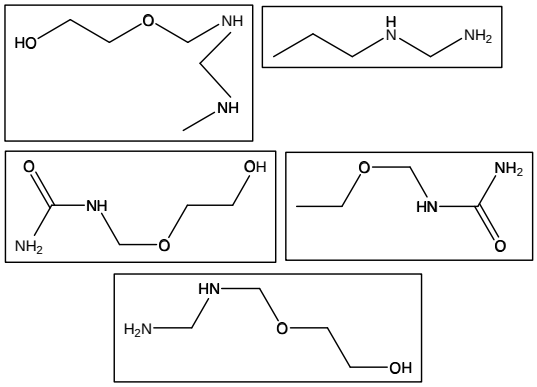
[13] S. Kramer, L. De Raedt, and C. Helma. Molecular feature mining in hiv data. In *7th International Conference on Knowledge Discovery and Data Mining*, 2001.

[14] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *IEEE International Conference on Data Mining*, 2001.

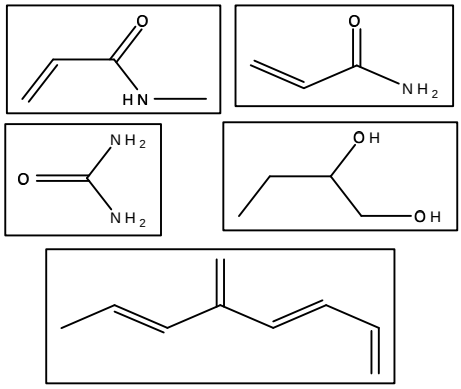
[15] K. Morik, P. Brockhausen, and T. Joachims. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *International Conference on Machine Learning*, 1999.

[16] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3), 2001.

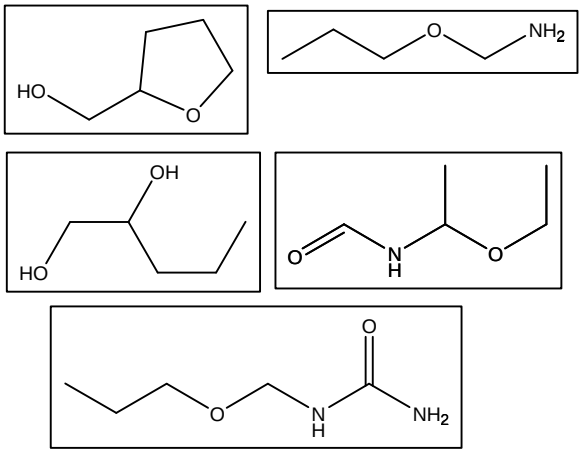
[17] A. Srinivasan, R. D. King, S. H. Muggleton, and M. Sternberg. The predictive toxicology evaluation challenge. In 15^{th} *IJCAI*, 1997.



(a) Top 5 highest confidence subgraphs from problem CA/CI.



(b) Top 5 highest confidence subgraphs from problem (CA+CM)/CI.



(c) Top 5 highest confidence subgraphs from problem CA/CI.

Figure 6: Top 5 highest confidence subgraphs discovered during classification problems: CA/CM, (CA+CM)/CI, and CA/CI

[18] V. Vapnik. *Statistical Learning Theory*. John Wiley, New York, 1998.

[19] X. Wang, J. T. L. Wang, D. Shasha, B. Shapiro, S. Dikshitulu, I. Rigoutsos, and K. Zhang. Automated discovery of active motifs in three dimensional molecules. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, 1997.

[20] O. Weislow, R. Kiser, D. L. Fine, , J. P. Bader, R. H. Shoemaker, and M. R. Boyd. New soluble fomrazan assay for hiv-1 cyopathic effects: appliication to high flux screening of synthetic and natural products for aids antiviral activity. *Journal of National Cancer Institute*, 1989.