Concept-based Data Mining with Scaled Labeled Graphs

Bernhard Ganter, Peter A. Grigoriev, Sergei O. Kuznetsov, Mikhail V. Samokhin

Technische Universität Dresden
All-Russian Institute for Scientific and Technical Information

Abstract Graphs with labeled vertices and edges play an important role in various applications, including chemistry. A model of learning from positive and negative examples, naturally described in terms of Formal Concept Analysis (FCA), is used here to generate hypotheses about biological activity of chemical compounds. A standard FCA technique is used to reduce labeled graphs to object-attribute representation. The major challenge is the construction of the context, which can involve tens thousands attributes. The method is tested against a standard dataset from an ongoing international competition called Predictive Toxicology Challenge (PTC).

1 Introduction

In [1] we introduced a general construction based on a semilattice of object description, which we called *pattern structure*. An example that we used was related to a lattice on sets of labeled graphs. In general, pattern structures are naturally reduced to formal contexts. In this paper we present a practical data mining approach which uses JSM or concept-based hypotheses. On the data side we use a standard FCA technique, called ordinal scaling [2] for the reduction of labeled graphs to formal contexts. We consider a chemical application in Predictive Toxicology and compare the results to those obtained with the same learning model, but different representation language which used predefined descriptors (attributes) for describing chemical compounds.

2 A Learning Model

2.1 Pattern Structures

In [1] we showed how such an approach is linked to the general FCA framework [2]. In [3] and in [4] we showed how this approach is related to standard machine learning models such as version spaces and decision trees.

Let G be some set, let $(D, \sqcap)$ be a meet-semilattice and let $\delta : G \to D$ be a mapping. Then $(G, \underline{D}, \delta)$ with $\underline{D} = (D, \sqcap)$ is called a *pattern structure*, provided that the set

$$\delta(G) := \{\delta(g) \mid g \in G\}$$

generates a complete subsemilattice $(D_\delta, \sqcap)$ of $(D, \sqcap)$ (e.g., when $(D, \sqcap)$ is complete, or when G is finite), i.e., every subset X of $\delta(G)$ has an infimum $\sqcap X$ in $(D, \sqcap)$ and D_δ is the set of these infima.

If $(G, \underline{D}, \delta)$ is a pattern structure, the derivation operators are defined as

$$A^\diamond := \sqcap_{g \in A}\ \delta(g) \qquad \text{for } A \subseteq G$$

and

$$d^\diamond := \{g \in G \mid d \sqsubseteq \delta(g)\} \qquad \text{for } d \in D.$$

The elements of D are called *patterns*. The natural order on them is given, as usual, by

$$c \sqsubseteq d :\Longleftrightarrow\ c \sqcap d = c,$$

and is called the *subsumption* order.

The operators $(.)^\diamond$ obviously make a Galois connection between the power set of G and $(D, \sqsubseteq)$. The pairs (A, d) satisfying

$$A \subseteq G, \quad d \in D, \quad A^\diamond = d, \quad \text{and} \quad A = d^\diamond$$

are called the *pattern concepts* of $(G, \underline{D}, \delta)$, with extent A and *pattern intent* d. For $a, b \in D$ the *pattern implication* $a \to b$ holds if $a^\diamond \subseteq b^\diamond$. Similarly, for $C, D \subseteq G$ the *object implication* $C \to D$ holds if $C^\diamond \sqsubseteq D^\diamond$.

Since $(D_\delta, \sqcap)$ is complete, there is a (unique) operation $\sqcup$ such that $(D_\delta, \sqcap, \sqcup)$ is a complete lattice. It is given by

$$\sqcup X := \sqcap\{c \in D_\delta \mid \forall_{x \in X}\ x \sqsubseteq c\}.$$

A subset M of D is $\sqcup$*-dense* for $(D_\delta, \sqcap)$ if every element of D_δ is of the form $\sqcup X$ for some $X \subseteq M$. If this is the case, then with

$$\downarrow d := \{e \in D \mid e \sqsubseteq d\}$$

we get

$$c = \sqcup(\downarrow c \cap M) \qquad \text{for every } c \in D_\delta.$$

Of course, $M := D_\delta$ is always an example of a $\sqcup$-dense set.

If M is $\sqcup$-dense in $(D_\delta, \sqcap)$, then the formal context (G, M, I) with I given as $gIm :\Leftrightarrow m \sqsubseteq \delta(g)$ is called a *representation context* for $(G, \underline{D}, \delta)$.

In [1] we proved that for any $A \subseteq G$, $B \subseteq M$ and $d \in D$ the following two conditions are equivalent:

1. (A, d) is a pattern concept of $(G, \underline{D}, \delta)$ and $B = \downarrow d \cap M$.
2. (A, B) is a formal concept of (G, M, I) and $d = \bigsqcup B$.

Thus, the pattern concepts of $(G, \underline{D}, \delta)$ are in 1-1-correspondence with the formal concepts of (G, M, I). Corresponding concepts have the same first components (called *extents*). These extents form a closure system on G and thus a complete lattice, which is isomorphic to the concept lattice of (G, M, I).

2.2 Hypotheses in Pattern Structures

In [5–7] we considered a learning model from [8] in terms of Formal Concept Analysis. This model assumes that the cause of a *target property* resides in common attributes of objects that have this property.

For pattern structures this can be formalized as follows. Let $(G, \underline{D}, \delta)$ be a pattern structure together with an external target property ω. As in the case of standard contexts, the set G of all objects is partitioned into three disjoint sets w.r.t. ω: the sets G_+, G_-, G_τ of positive, negative, and undetermined examples, respectively. For positive examples it is known that they have property ω, for negative examples it is known that they do not have property ω, and for undetermined examples it is not known whether they have or do not have ω. This gives three pattern substructures of $(G, \underline{D}, \delta)$: $(G_+, \underline{D}, \delta_+)$, $(G_-, \underline{D}, \delta_-)$, $(G_\tau, \underline{D}, \delta_\tau)$, where δ_ε for $\varepsilon \in \{+, -, \tau\}$ are restrictions of δ to the corresponding sets of examples. For brevity sake, we shall write just δ instead of δ_ε.

A *positive hypothesis* h is defined as a pattern intent of $(G_+, \underline{D}, \delta)$ that is not subsumed by any pattern from $\delta(G_-)$ (for short: not subsumed by any negative example). Formally: $h \in D$ is a positive hypothesis iff

$$h^\diamond \cap G_- = \emptyset \text{ and } \exists A \subseteq G_+ : A^\diamond = h.$$

A *negative hypothesis* is defined accordingly. A hypothesis in the sense of [9, 8, 7] is obtained as a special case of this definition when $(D, \sqcap) = (2^M, \cap)$ for some set M then we have a standard context (object-attribute) representation.

Hypotheses can be used for classification of undetermined examples as introduced in [8] in the following way. If $g \in G_\tau$ is an undetermined example, then a hypothesis h with $h \sqsubseteq \delta(g)$ is *for the positive classification* of g if h is positive and *for the negative classification* of g if it is a negative hypothesis.

An example $g \in G_\tau$ is classified positively if there is a hypothesis for its positive classification and no hypothesis for the negative classification. g is classified negatively in the opposite case. If there are hypotheses for both positive and negative classification, then some other methods (based on standard statistical techniques) may be applied.

2.3 An example with labeled graphs

Consider a pattern structure based on a given ordered set G of graphs (V, E) with vertex- and edge-labels from the sets $(\mathcal{L}_\mathcal{V}, \preceq)$ and $(\mathcal{L}_\mathcal{E}, \preceq)$. Each labeled graph Γ from G is a quadruple of the form $((V, l), (E, b))$, where V is a set of vertices, E is a set of edges, $l \colon V \to \mathcal{L}_\mathcal{V}$ is a function assigning labels to vertices, and $b \colon E \to \mathcal{L}_\mathcal{E}$ is a function assigning labels to edges. We do not distinguish isomorphic graphs with identical labelings.

The order is defined as follows: For two graphs $\Gamma_1 := ((V_1, l_1), (E_1, b_1))$ and $\Gamma_2 := ((V_2, l_2), (E_2, b_2))$ from G we say that Γ_1 **dominates** Γ_2 or $\Gamma_2 \leq \Gamma_1$ (or Γ_2 is a **subgraph** of Γ_1) if there exists a one-to-one mapping $\varphi : V_2 \to V_1$ such that it

- respects edges: $(v, w) \in E_2 \Rightarrow (\varphi(v), \varphi(w)) \in E_1$,
- fits under labels: $l_2(v) \preceq l_1(\varphi(v))$, $(v, w) \in E_2 \Rightarrow b_2(v, w) \preceq b_1(\varphi(v), \varphi(w))$.

Two small examples are given in Figure 1.

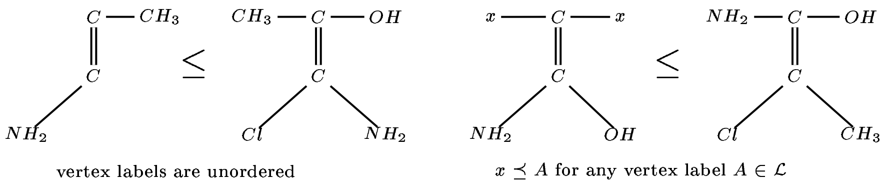


Figure1. With $\mathcal{L}_\mathcal{V} = \{\text{C, NH}_2\text{, CH}_3\text{, OH}, x\}$ we have subgraphs as shown in the diagrams above. In all subsequent examples the label value x will not be admitted and unordered vertex labels will be used

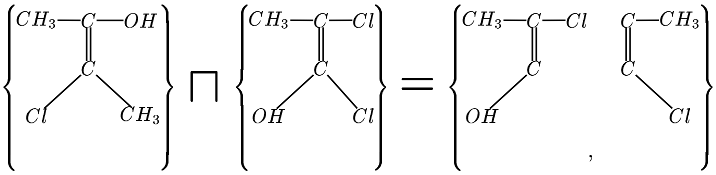
A pattern structure for these graphs then is defined as $(G, \underline{D}, \delta)$, where the semilattice $\underline{D} := (D, \sqcap)$ consists of all sets of subgraphs of graphs from G ("graph sets"), and the meet operation $\sqcap$ on graph sets is defined as follows: For two graphs X and Y from G

$$\{X\} \sqcap \{Y\} := \{Z \mid Zy \leq X, Y, \ \forall Z_* \leq X, Y \ Z_* \not\geq \ Z\},$$

i.e., $\{X\} \sqcap \{Y\}$ is the set of all maximal common subgraphs of X and Y. The meet of non-singleton sets of graphs is defined as

$$\{X_1, \ldots, X_k\} \sqcap \{Y_1, \ldots, Y_m\} := \mathrm{MAX}_{\leq}(\sqcup_{i,j}(\{X_i\} \sqcap \{Y_j\}))$$

for details see [10, 6, 1]. Here is an example of applying $\sqcap$ defined above:



To get an example of such a pattern structure, let $G := G_+ \cup G_-$, Where G_+ consists of the first four graphs $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$ of Figure 2 and $G_- := \{\Gamma_5, \Gamma_6, \Gamma_7\}$.

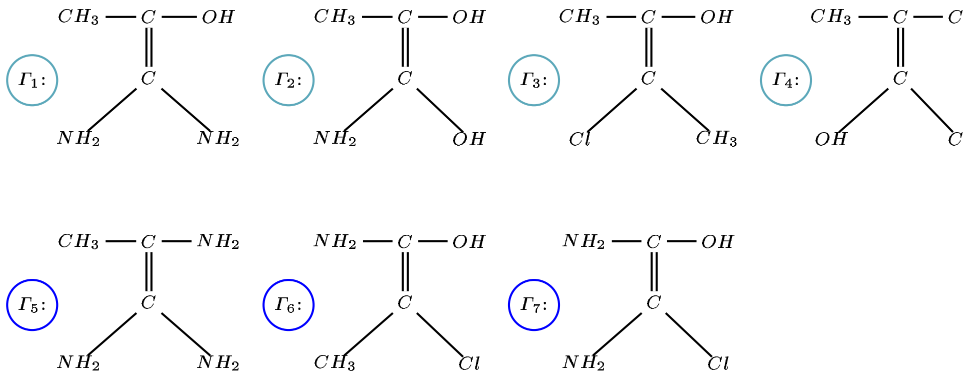


Figure2. Seven labeled graphs for a pattern structure.

The pattern concept lattice of the positive pattern structure $(G_+, \underline{D}, \delta)$ is given in Figure 3.

2.4 Projections and projected hypotheses

Since for some pattern structures (e.g., for the pattern structure given by sets of graphs with labeled vertices) even computing subsumption relation may be NP-hard, in [1] we introduced projection operators to approximate pattern structures. A projection (kernel operator) is a mapping $\psi \colon D \to D$ which is

monotone: if $x \sqsubseteq y$, then $\psi(x) \sqsubseteq \psi(y)$,
contractive: $\psi(x) \sqsubseteq x$, and
idempotent: $\psi(\psi(x)) = \psi(x)$.

Any projection of a complete semilattice $(D, \sqcap)$ is $\sqcap$-preserving, i.e., for any $X, Y \in D$

$$\psi(X \sqcap Y) = \psi(X) \sqcap \psi(Y),$$

which helps us to describe how the lattice of pattern concepts changes when we replace $(G, \underline{D}, \delta)$ by its approximation $(G, \underline{D}, \psi \circ \delta)$. First, we note that $\psi(d) \sqsubseteq \delta(g) \Leftrightarrow \psi(d) \sqsubseteq \psi \circ \delta(g)$. Then, using the basic theorem of FCA (which, in particular allows one to represent every lattice as a concept lattice), we showed how the projected pattern lattice is represented by a context [1]:

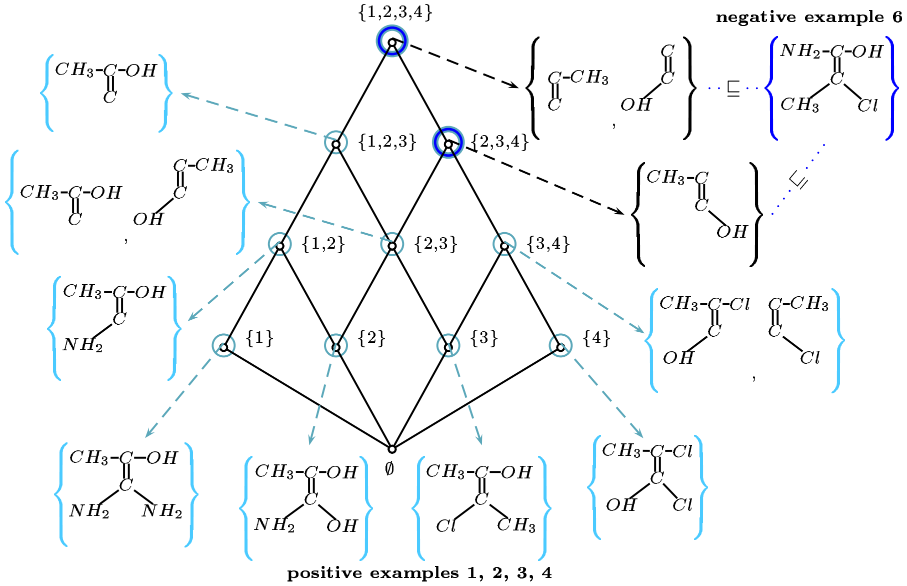


Figure3. The pattern concept lattice of the positive pattern structure

Theorem 1 *For pattern structures $(G, \underline{D}, \delta_1)$ and $(G, \underline{D}, \delta_2)$ the following statements are equivalent:*

1. *$\delta_2 = \psi \circ \delta_1$ for some projection ψ of $\underline{D}$.*
2. *There is a representation context (G, M, I) of $(G, \underline{D}, \delta_1)$ and some $N \subseteq M$ such that $(G, N, I \cap (G \times N))$ is a representation context of $(G, \underline{D}, \delta_2)$.*

The properties of projection allow one to relate hypotheses in the original representation with those approximated by a projection. As in [1] we use the term "hypothesis" to those obtained for $(G, \underline{D}, \delta)$ and we refer to those obtained for $(G, \underline{D}, \psi \circ \delta)$ as *ψ-hypotheses.* There is no guarantee that the ψ-image of a hypothesis will be a ψ-hypothesis. In fact, our definition allows that ψ is the "null projection" with $\psi(d) = \mathbf{0}$ for all $d \in D$ (total abandoning of the data with no interesting hypotheses). However, if $\psi(d)$ is a (positive) hypothesis, then $\psi(d)$ is also a (positive) ψ-hypothesis. If we want to look the other way round, we have the following: if $\psi(d)$ is a (positive) ψ-hypothesis, then $\psi(d)^{\diamond\diamond}$ is a (positive) hypothesis [1].

The set of all hypothesis-based classifications does not shrink when we pass from d to $\psi(d)$. Formally, if d is a hypothesis for the positive classification of g and $\psi(d)$ is a positive ψ-hypothesis, then $\psi(d)$ is for the positive classification of g.

The above observations show that we can generate hypotheses starting from projections. For example, we can select only those that can be seen in the projected data, which is suggested by the following theorem from [1]:

Theorem 2 *For any projection ψ and any positive hypothesis $d \in D$ the following are equivalent:*

1. *$\psi(d)$ is not subsumed by any negative example.*
2. *There is some positive ψ-hypothesis h such that $h^{\diamond\diamond} \sqsubseteq d$.*

An example is shown in Figure 4). We have used the same data as in Figure 3, but the set D of graph sets was restricted to graphs with less then four vertices.

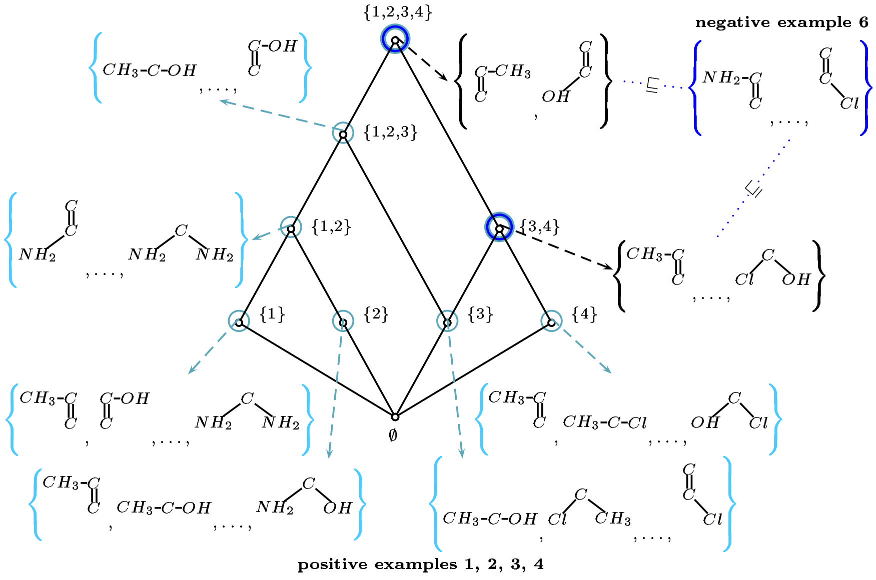


Figure4. The lattice of the projected positive pattern structure

3 Scaling Labeled Graphs and Their Projections

In this section we shall discuss an application of the learning model, introduced in Section 2 to the problem of bioactivity prediction. In our experiments we will use the data from the Predictive Toxicology Challenge (PTC), an ongoing international competition. First, we shall briefly describe the format of PTC (Subsection 3.1). Then we shall discuss k-projections of labeled undirected graphs as a means of approximate representation of graph data.

3.1 Predictive Toxicology Challenge

The program of a workshop on Predictive Toxicology Challenge (PTC) [11], (at the joint 12th European Conference on Machine Learning and the 5th European Conference on Principles of Knowledge Discovery in Databases) consisted in a competition of machine learning programs for generation of hypothetical causes of toxicity from positive and negative examples of toxicity. The organizers (Machine Learning groups of the Freiburg University, Oxford University, and University of Wales) together with toxicology experts (US Environmental Protection Agency, US National Institute of Environmental and Health Standards) provided participants with training and test datasets.

The training dataset consisted of descriptions of 185 molecular graphs of 409 chemical compounds with indication of whether a compound is toxic or not for a particular sex/species group out of four possible groups: male mice, female mice, male rats and female rats. For each group there were about 120 to 150 positive examples and 190 to 230 negative examples of toxicity. The test dataset consisted of 185 substances for which forecasts of toxicity should be made. Twelve research groups (world-wide) participated in PTC, each with up to 4 prediction models for every sex/species group.

3.2 Representation contexts for graph data

The source data of the PTC datasets are molecular graphs. These are graphs in the sense of Subsection 2.3 above, with labeled Vertices and edges. We cant therefore apply the methods from Section 2. Another view is the following (working directly with a representation context): Think of the source data as a many-valued context with a single many-valued attribute “graph”, assigning to each compound its molecular graph. This many-valued context is (ordinaly) scaled one, such that the attributes of the derived context are all (connected) subgraphs of the graphs under consideration and each graph has its subgraps as attributes.

However, generating this context is a rather time-consuming process. The difficulty of generating all subgraphs is due to costly isomorphism and subgraph isomorphism testing (the latter is an NP-complete problem). There are several well-known algorithms for these problems, e.g., that of B. D. McKay [12] for isomorphism testing and the algorithm of J. R. Ullmann [13] for testing subgraph isomorphism. Since the generation of all subgraphs for an arbitrary labeled graph is a computationally hard task, we use k-projections of initial graphs. The notion of projection introduced in Section 2 for general semilattices can be specified for the lattice on graph sets, e.g., as follows.

Definition 1. *Let $\Gamma = ((V,l),(E,b))$ be a labeled graph. The set $S_\Gamma = \{\Gamma_* = ((V_*,l_*),(E_*,b_*)) \mid \Gamma_*$ is connected, $\Gamma_* \leq \Gamma$, $|V_*| \leq k\}$ is called a k-projection of Γ.*

Thus, k-projection of a labeled graph Γ is a set of all subgraphs (up to isomorphism) of Γ with up to k-size set of vertices. Obviously, k-projection satisfies the properties of the kernel operator. When we use k-projections of graphs, then in the corresponding representation context (G, M, I), the set of objects G is a set of chemical compound names, M is a set of subgraphs of molecular graphs with k or less vertices and gIm means that the graph $m \in M$ is a subgraph of the molecular graph of the compound $g \in G$.

So far, we have generated these contexts for the values of k from 1 up to 8. With the growth of k, the number of attributes in the resulting scaled context becomes very large (thousands of attributes), but reduction of attributes (a standard FCA technique) reduces the size of contexts in several times, see Figure 5.

projection size	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
# attributes in full context	22	95	329	1066	3275	9814	28025	76358
# attributes in reduced context	22	72	153	373	812	1548	2637	3981
reducing time (in sec.)	1	1	2	5	16	57	219	883

Figure5. PTC dataset: number of attributes in representation contexts before and after attribute reduction

4 QuDA: Qualitative Data Analysis

After the preprocessing step described above, we had 8 datasets. Each of these datasets was an encoding of the original PTC data by means of 1- to 8-projections, respectively. Our goal was to test whether hypotheses obtained for these encodings can be better than those obtained for other encoding schemes proposed by the PTC participants [11]. To do that we have used the QuDA software [14, 15].

4.1 A brief description of QuDA

History and motivation. *QuDA, a data miners' discovery environment* was developed at the Intellectics group of the Darmstadt University of Technology in 2001-2003. The authors are P. Grigoriev and S. Yevtushenko. This project was started as a "companion" data mining system for the DaMiT tutorial [16] by initiative of the authors.
Usage. QuDA can serve as a personal data mining environment for analyzing mid-sized datasets (up to ten thousands of records). Most of its functionality is also accessible from external applications thus allowing the use of it in integrated solutions. QuDA has an open architecture; it supports scripting and has import/export capabilities for the most commonly used data/model formats.
Functionality. QuDA implements various data mining techniques, including association rule mining, decision tree induction, JSM-reasoning[1], Bayesian learning, and interesting subgroup discovery. It provides also several preprocessing and post-processing utilities, including data cleaning tools, visualization of attribute distributions, dynamic Hasse diagrams, and a ruleset navigator.
Implementation. QuDA is implemented entirely in Java. The system consists of approximately 1700 classes; the source codes take approximately 3.5Mb.
Availability. QuDA is freeware. It is available for download at the DaMiT tutorial site (http://damit.dfki.de) as well as at its own site at the Intellectics group (http://ki-www2.intellektik.informatik.tu-darmtadt.de/˜jsm/QDA).

4.2 Using QuDA for bioactivity prediction: sequential covering

QuDA implements several variants of the learning model described in Section 2. In our experiments we used the strategy of sequential covering with structural similarities. This procedure does not generate the set of all hypotheses, but generates a subset of it, sufficient to explain every of the examples in the training set.

Briefly, this strategy can be described as follows:

1. objects in G_+ are sorted according to some fixed linear order (e.g., in the order they appear in the dataset);
2. the set of hypotheses H is initialized with an empty set;
3. first object g_+ in G_+, which is not covered by any hypothesis in H is selected;
4. a hypotheses h, covering g_+ is found by generalizing its description with descriptions of other objects in G_+ uncovered so far by H;
5. the new-found hypothesis h is added to H and the procedure continues from the step 3 until every object in G_+ is covered by at least one hypothesis in H.

A pseudocode for the main loop of this procedure is given in Figure 6. Figure 7 provides pseudocode for the step 4: finding a hypothesis that explains a particular example, uncovered so far. We provide only the pseudocode for generating *positive* hypotheses. Generating *negative* hypotheses is organized dually.

Although the sequential covering strategy has a number of obvious drawbacks (most notably – its dependence on the selected order of objects), we decided to use this strategy instead of generating all hypotheses for several reasons:

- it has attractive computational complexity: linear in the number of attributes in the representation context (see Section 2), linear in the number of negative examples (G_-), and quadratic in the number of positive examples (G_+);
- in practical experiments on a number of real-world datasets [15] it has shown classification accuracy and recall comparable to those of the strategy where all hypotheses are generated.

```
function SequentialCovering()
{
  $H$ := $\emptyset$;

  $UnexplainedExamples$ := $G_+$;

  for each $g$ in $G_+$
  {
    if $g \in UnexplainedExamples$
    {
       $explanation$ := findExplanation($g$, $UnexplainedExamples$);
       $UnexplainedExamples$ := $UnexplainedExamples \setminus explanation^\diamond$;
       $H$ := $H \cup \{explanation\}$;
    }
  }
  return $H$;
}
```

Figure6. Sequential covering: the main loop

```
function findExplanation($g$, $P$)
{
  $explanation$ := $\delta(g)$;

  for each $g_+$ in $P$
  {
     $candidate$ := $explanation \sqcap \delta(g_+)$
     if $candidate^\diamond \cap G_- = \emptyset$
       $explanation$ := $candidate$;
  }
  return $explanation$;
}
```

Figure7. Sequential covering: `findExplanation` procedure

We conclude this section with a final remark, which aims at making our results reproducible. The sequential covering strategy naturally depends on the order on objects. This order is used in the main loop to select the next object to find an explanation for; at the same time this order determines the sequence in which objects are used for generalization in the `findExplanation` procedure. In our experiments we have used the order in which objects were presented in the source dataset.

5 Experiments with the PTC dataset

One of the standard techniques used in machine learning for comparison of the classification algorithms is the so-called ROC-analysis [2]. Here, results obtained by a learning model are represented by a point on (x, y)-plane, where x stays for the relative number of false positive predictions and y stays for the relative number of true positive predictions. The best (usually unattaianble) point is (0,1) and the strait line from (0,0) to (1,1) corresponds to models that are equivalent to random guess under uniform distribution. When the costs of correct classification and misclassification are not known in advance, "best" models correspond to points lying on the convex hull of leftmost points. ROC-diagrams were used in the Predictive Toxicology Challenge to select the best learning models. Here we also use ROC-diagrams to demonstrate the usage of projected pattern structures for bioactivity prediction in comparison to other encoding schemes and/or other learning models.

5.1 Projected pattern structures "On the ROC"

The results are shown in Figure 8. The following abbreviations are used:

- PR1, PR2, ..., PR8 – the results obtained using 1- to 8-Projection representations, respectively, in combination with sequential covering strategy;
- WAI1, GONZ, KWAI, LEU3 are other "best" models submitted to the Predictive Toxicology Challenge for this animal group.

Note, that the Figure 8 shows both the "old" ROC-curve (composed by LEU3, KWAI, GONZ, and WAI1 models) and the "new" one (composed by LEU3, PR7, PR5, and WAI1 models).

As one can "read" from the ROC-diagram in Figure 8:

- using 1-Projections does not lead to any classifications at all (false positive rate and true positive rate equal zero);
- using 2-Projections results in rather bad classifications: the corresponding point on the ROC-diagram is below the diagonal;
- using 3-Projections results in better classifications: the corresponding point on the ROC-diagram is above the diagonal;
- using 4-Projections results in even better classifications: the corresponding point is above the "old" ROC-curve;
- using 5-Projections occurs to be one of the four "new" best strategies: it results in making 8 true positive predictions with only 2 false positive ones;
- using 6-Projections, however, does not result in better classifications: the number of true positives decreases to 6; the number of false positives remains the same;
- using 7-Projections, with 4 true positives and 1 false positives again appears on the "new" ROC-curve;

[1] This name stands for a group of lattice-based machine learning methods, including the model described in Section 2.

[2] ROC is the abbreviation for Receiver Operating Characteristic, see [17]

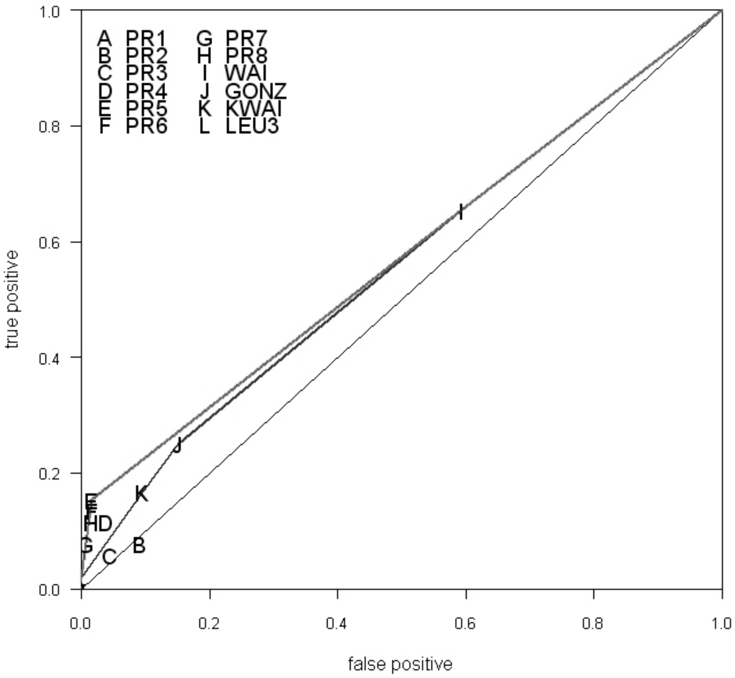


Figure8. Projected pattern structures "On the ROC". Animal group: MR (male rats).

- using 8-Projections increases the number of true positives to 6, but also increases the number of false positives to 2; this strategy is thus strictly worse than using 5-Projections (assuming positive cost of making a true positive classification);

For the FR group (female rats; see Figure 9) the strategies with 4-, 5-, 6-, and 8-Projections occur above the "old" ROC-curve, without, however, making any of the "old" best models (LEU3 and KWAI) lose their positions.

For the other two animal groups (MM and FM, male and female mice) our strategy did not bring any good results (see Figure 10).

6 Conclusions and Further Work

Our practical result is: *In two animal groups of the four the classification accuracy obtained with molecular graph projections and sequential covering strategy appeared to be among the best known.* In the other two groups, however, this was not the case.

Somewhat more interesting, although expected result is the demonstrated *non-monotonicity of the classification accuracy by k-Projections with the growth of k.* At first glance, this result may seem strange, as increasing the size of projections we increase the amount of information available for the learning algorithm. However, in practice this information growth often results in generating more irrelevant hypotheses and thus, in the decrease of classification accuracy.

The most interesting directions of further research are as follows:

- check the proposed approach on other real-world datasets involving graph representation; these include other datasets from the bioactivity prediction domain as

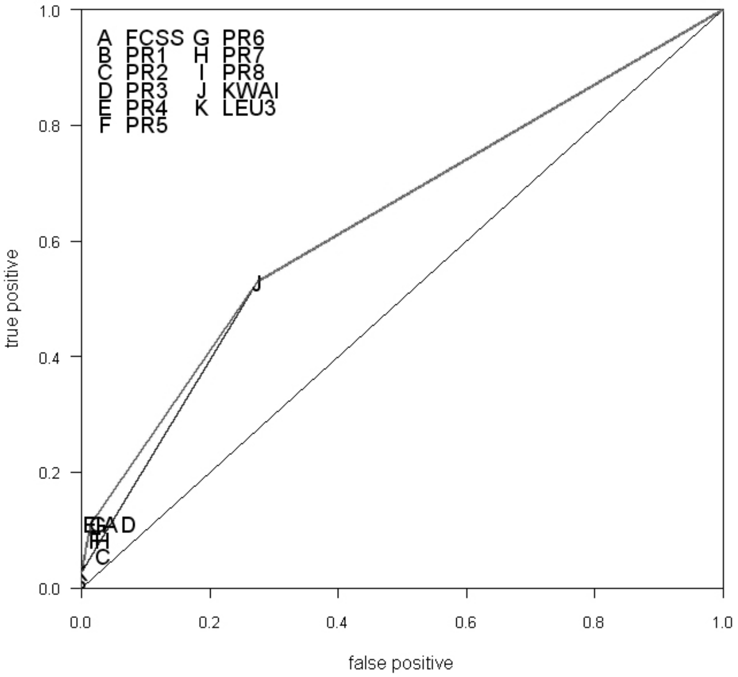


Figure9. Projected pattern structures “On the ROC”. Animal group: FR (female rats).

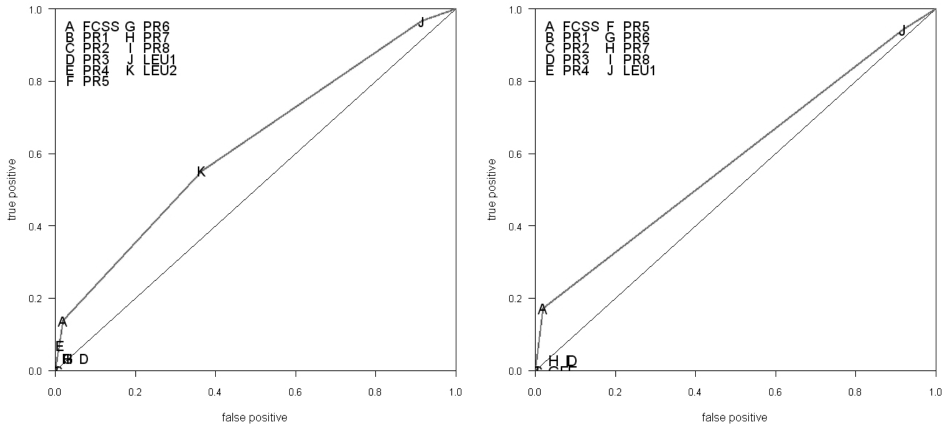


Figure10. Projected pattern structures “On the ROC”. Animal groups: MM and FM (male and female mice).

well as datasets from other domains, e.g. web structure mining, ontology-based text mining, etc.;

- incorporate standard machine learning algorithms for feature selection and/or develop specialized ones to overcome classification accuracy decrease when growing the size of projections;
- in our experiments we have used a trivial conflict resolution technique: if a certain object contained both positive and negative hypotheses it remained undefined; other strategies, e.g. voting, may prove more appropriate.

References

1. B. Ganter and S. Kuznetsov. Pattern Structures and Their Projections. In G. Stumme and H. Delugach, editors, *Proc. 9th Int. Conf. on Conceptual Structures, ICCS'01*, volume 2120 of *Lecture Notes in Artificial Intelligence*, pages 129–142. Springer-Verlag, 2001.
2. B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical foundations.* Springer-Verlag, Berlin Heidelberg New-York, 1999.
3. B. Ganter and S.O. Kuznetsov. Hypotheses and version spaces. In W. Lex A.de Moor and B.Ganter, editors, *Proc. 10th Int. Conf. on Conceptual Structures, ICCS'01*, volume 2746 of *Lecture Notes in Artificial Intelligence*, pages 83–95. Springer-Verlag, 2003.
4. S.O. Kuznetsov. Machine learning and Formal Concept Analysis. In P. Eklund, editor, *Concept Lattices*, volume 2961 of *Lecture Notes in Artificial Intelligence*, pages 287–312. Springer-Verlag, 2004.
5. S.O. Kuznetsov and V.K. Finn. On a model of learning and classification based on similarity operation. *Obozrenie Prikladnoi i Promyshlennoi Matematiki* **3**, 1:66–90, 1996. in Russian.
6. S.O. Kuznetsov. Learning of Simple Conceptual Graphs from Positive and Negative Examples. In J. Zytkow and J. Rauch, editors, *Proc. Principles of Data Mining and Knowledge Discovery, Third European Conference, PKDD'99*, volume 1704 of *Lecture Notes in Artificial Intelligence*, pages 384–392. Springer-Verlag, 1999.
7. B. Ganter and S. Kuznetsov. Formalizing Hypotheses with Concepts. In B. Ganter and G. Mineau, editors, *Proc. 8th Int. Conf. on Conceptual Structures, ICCS'00*, volume 1867 of *Lecture Notes in Artificial Intelligence*, pages 342–356. Springer-Verlag, 2000.
8. V.K. Finn. Plausible Reasoning in Systems of JSM Type. *Itogi Nauki i Tekhniki, Seriya Informatika*, 15:54–101, 1991. in Russian.
9. V.K. Finn. On Machine-Oriented Formalization of Plausible Reasoning in the Style of F. Backon–J. S. Mill. *Semiotika i Informatika*, 20:35–101, 1983. in Russian.
10. S.O. Kuznetsov. JSM-method as a machine learning method. *Itogi Nauki i Tekhniki, ser. Informatika*, 15:17–50, 1991. in Russian.
11. C. Helma, R.D. King, S. Kramer, and A. Srinvasan, editors. *Proc. of the Workshop on Predictive Toxicology Challegnge at the 5th Conference on Data Mining and Knowledge Discovery (PKDD'01) Freiburg (Germany)*, http://www.predictive-toxicology.org/ptc/, 2001, September 7.
12. B.D. McKay. Practical graph isomorphism. *Congressus Numerantium*, 30:45–87, 1981.
13. J.R. Ullmann. An algorithm for subgraph isomorphism. *J. Assoc. Comput. Mach.*, 23:31–42, 1976.
14. P.A. Grigoriev, S.A. Yevtushenko, and G.Grieser. QuDA, a data miner's discovery enviornment. Technical Report AIDA 03 06, FG Intellektik, FB Informatik, Technische Universität Darmstadt, http://www.intellektik.informatik.tu-darmstadt.de/∼peter/QuDA.pdf, September 2003.
15. P. Grigoriev and S. Yevtushenko. JSM-Reasoning as a data mining tool. In *Proceedings of the 8th Russian National Conference on Artificial Intelligence, CAI-2002*, pages 112–122, Moscow, 2002. PhysMathLit. In Russian.
16. DaMiT, the **Da**ta **Mi**ning online **T**utorial. http://damit.dfki.de.
17. F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42:203–231, 2001.