ILP-based Rule Induction for Predicting Carcinogenicity

Hayato Ohwada, Masahiro Koyama, Yu Hoken

Faculty of Sci. and Tech., Science University of Tokyo

1 Introduction

This paper describes an ILP-based approach to carcinogenicity prediction and its initial result. The approach covers practical aspects on ILP, including ILP system design, bias setting, and learning method. The goal of this study is to clarify the applicability of our original ILP system as an AI tool to the challenging problem rather than the predictive value, usability and scientific insight.

2 Materials

In this study, our ILP system (GKS) implemented in Prolog languages was used for carcinogenicity prediction. The data set we selected was KULeuven, because of the Prolog syntax. For our experiments, carcinogenicity definitions classified into "EE", "IS" and "E" were removed. We constructed positive examples from "CE", "SE" and "P", and negative examples from "NE" and "N". According to the problems ("MM", "FM", "MR" and "FR"), the numbers of positive and negative examples are shown below.

	MM	FM	MR	FR
Positive	128	143	152	121
Negative	207	206	192	230
Total	335	349	344	351

3 Engineering effort

For background knowledge, we selected 33 functional groups, "atom" and "distance_charges". To deal with inequalities about "fg_distance" and "distance_charges" efficiently, we introduced the following predicates:

```
gteq(X,Y) :-
```

```
        not(var(X)), not(var(Y)),
        float(X), float(Y),
        X ≥ Y, !.

gteq_total_fg(A,C) :-
        fg_distance(A,B,_), gteq(B,C).

gteq_sum_dis(A,C) :-
        fg_distance(A,_,B), gteq(B,C).

gteq_distance(A,C) :-
        distance_charges(A,_,_,_,_,B), gteq(B,C).
```

In addition, "fg_numbers" having great many arguments is replaced by the following concise descriptions:

```
number_of(aromatic,A,B) :-
        fg_number(A,B,_,_,...,_), B > 0.
```

4 ILP system

Our ILP system (called GKS) is based on inverse entailment employed in PROGOL. First, GKS generates the most specific clause with respect to a selected positive example, and searches for a hypothesis among the lattice constructed by the clause. In the normal learning mode of GKS, this is the same as PROGOL. However, PROGOL-based search is not reasonable for the carcinogenesis prediction. In fact, enormous time is needed for learning and we could not produce rules. In contrast, GKS effectively handles constraints that are regarded as conditional statements to exclude negative examples, and produces a plausible combination of constraints using FOIL-like hill-climbing search. This combination is achieved in a lazy manner; constraints are combined in evaluating a hypothesis. This is similar to Srinivasan's idea to deal with numerical constraints. In our attempt, predicates with no output argument such as functional groups and inequalities are handled as constraints that are combined by hill-climbing heuristics to avoid combinatorial explosion.

5 Bias setting

For search bias, GKS takes as input the parameter values of "depth", "positive", "clause_size" and "ratio" where "depth" is the allowable depth of variable connectivity for input-output relationship, "positive" is the minimum number of positive examples covered by a hypothesis, "clause_size" is the maximum num-

ber of literals in the hypothesis, and "ratio" is the maximum fraction of the numbers of positive and negative examples. We put the values of the parameters as depth=2, positive=3, clause_size=10, and changes the value ratio to produce rules with different strength. Another bias is that the maximum number of the "atom" literal is three. This restriction is due to the functional group definitions.

6 Learning method

The following is the learning algorithm that exchanges positive and negative examples to produce different types of rules. This is due to the fact that a large number positive examples could not be explained by the produced rules in the normal learning method (i.e. covering positive examples and excluding negative examples). In that case, usual treatment is that a permissible ratio negative examples covered should be relaxed. However, the resulting rules are more general, covering a larger number of negative examples. In contrast, we focus on negative rules that cover negative examples, and handle rule strength obtained by combining the numbers of positive and negative examples the rule covers. Based on this metrics, we expect that examples are correctly classified.

```
1     let ES be a set of "active" facts;
2     let NS be a set of "inactive" facts;
3     for (each ratio) {
4        let ES be the positive examples;
5        let NS be the negative examples;
6        run GKS;
7        let NS be the positive examples;
8        let ES be the negative examples;
9        run GKS;
10    }
```

The produced set of rules are applied by the function "Classify(e)":

```
1   Classify (e)
2      let S be a set of rules;
3      while (S is not empty) {
4         Select r from S whose ratio is the lowest;
5         if (active(e) is deducible form r)
6            return "active";
7         if (inactive(e) is deducible form r)
8            return "inactive";
9         remove r from S;
10     }
```

7 Results

We employed 10-fold cross validation. The classification accuracy for each problem is as follows:

	MM	FM	MR	FR
Average	0.6411	0.6369	0.5559	0.6543

8 Concluding Remarks

In this study, different parameter settings was employed to produce rules with different strength (accuracy). This rule strength can be used to not only realize reasonable classifiers but also evaluate the output of the classifiers. We hope that this feature will be preferable for toxicology researchers.