Inductive Logic Programming for Multiple-Part Data: Applications on Structure-Activity Relationship Studies

Cholwich Nattee — The Institute of Scientific and Industrial Research, Osaka University
cholwich@ai.sanken.osaka-u.ac.jp

Sukree Sinthupinyo — (affiliation as previous author)
sukree@ai.sanken.osaka-u.ac.jp

Masayuki Numao — (affiliation as previous author)
numao@sanken.osaka-u.ac.jp

keywords: inductive logic programming, multiple-instance learning, structure-activity relationship studies

Summary

Inductive Logic Programming (ILP) becomes interesting when the expressive power of first-order representation provides comprehensibility to learning result and capability to handle more complex data consisting of their relations. Nevertheless, the bottleneck for learning first-order theory is enormous hypothesis search space which causes inefficient performance by the existing learning approaches compared to the propositional approaches. This paper introduces an improved ILP approach capable of handling more efficiently a kind of data called multiple-part data, i.e., one instance of data consists of several parts as well as relations among parts. This approach tries to find hypothesis describing class of each training example by using both individual and relational characteristics of its part which is similar to finding common substructures among the complex relational instances. The multiple-part data can be found in various domains especially on Structure-Activity Relationship (SAR) studies which aim to generate hypotheses describing activities or characteristics of chemical compounds from their own structures. Each compound is composed of atoms as parts, and various kinds of bond as relations among atoms. We then apply the proposed algorithm for SAR studies by conducting experiments on two real-world datasets: mutagenicity in nitroaromatic compounds and dopamine antagonist compounds. The experiment results were compared to the previous approaches in order to show the performance of proposed approach.

1. Introduction

Inductive learning of first-order theory from examples is interesting because first-order representation provides comprehensibility to the learning results and capability to handle more complex data consisting of relations. Yet, the bottleneck for learning first-order theory is enormous hypothesis search space. Moreover, heuristic functions applied in the existing ILP approaches use only quantitative information to select an appropriate candidate, i.e., using only the number of training examples covered without considering the quality. This makes existing approaches sometimes perform worse than propositional approaches. Except from defining heuristic function, language bias is one of techniques used in order to reduce the search space. It is widely used in many ILP systems. However, this research focuses on proposing a heuristic function.

We introduce a novel learning approach focusing on a kind of data called *multiple-part data*, i.e., one instance of data consists of several parts as well as relations among parts. The objective of learning from multiple-part data is to find hypothesis for describing class of each example by using part characteristics individually and characteristics of relations among parts. This is similar to finding common substructures among instances in the same class.

Though the existing first-order theory learning approaches can handle this kind of data due to the power of first-order representation, there is a limitation in efficiency of results since numerous parts within one example make the search space become larger but contains similar hypothesis candidates. Thus, the search heuristics cannot lead to good hypotheses. In order to solve this problem, we propose an approach that weights each part according to its characteristics correlating to parts from other examples in the same class. This makes parts with common characteristics be given higher weights than the uncommon parts, and makes the search heuristics discriminate more efficiently. We adopt this weighting technique from the concept of multiple-instance learning which is an extended two-class propositional learning approach for data that are unable to be labeled individually, albeit several instances of data are gathered and labeled as a group. Each positive group may consist of both positive and negative instances. Nevertheless, the multiple-instance learning aims to learn to predict instances not groups, thereby rendering itself similar to supervised learning with noises in positive examples. Most learning algorithms for multiple-instance data solve this ambiguity by using similarity of data within the feature space to find the area where several instances from various positive groups are located together and negative group instances are far. This method is modified and used as the weighting technique to evaluate each part of the multiple-part data containing similarity among parts before incorporating the weights into search heuristics to find hypothesis that may consist of relations among parts.

To evaluate the proposed approach, we conducted an experiment on SAR studies for chemical compound structures. This is a promising process because the knowledge discovered will be useful for developing new drugs. These studies aim to predict the activity of compound from its structure. In recent years, the advance in High Throughput Screening (HTS) technology has produced vast amount of SAR data. Therefore, once the rules to predict activities of existing SAR data are found, it will significantly help screening process. SAR data represented by chemical compound structure can be categorized as multiple-part data. Because we aim to find substructures that predict activity of a compound, we apply the proposed system to learn hypotheses from this kind of data. We compare the learning results with the previous approaches in order to evaluate the performance.

This paper is mainly divided into two parts. We first introduce the multiple-part data, and describe the proposed approach. Then, we conduct the experiments for SAR studies on two chemical compound datasets. The experiment results are compared to the existing approaches to evaluate its performance. Finally, we conclude the paper and consider our future works.

2. Background

2・1 FOIL

FOIL [Quinlan 90] is a top-down ILP system which learns function-free Horn clause definitions of a target predicate using background predicates. Learning process in FOIL starts with training examples containing all positive and negative examples. The algorithm used in FOIL for constructing a function-free Horn clause consists of two main loops. In outer loop, a Horn clause partially covering the examples is constructed, and covered examples are removed from the training set. While in inner loop, partially developed clauses are iteratively refined by adding a literal one by one. Heuristic function is used to select the most appropriate clause. FOIL maintains covered examples in the form of *tuple* which is the substitutions (i.e., bindings of variables) of the clause under given example. Multiple tuples can be generated from one example.

FOIL uses a heuristic function based on the information theory for assessing usefulness of a literal. It provides effective guidance for clause construction. Purpose of this heuristic function is to characterize a subset of the positive examples. From the partial developing clause below

$$R(V_1, V_2, \ldots, V_k) \leftarrow L_1, L_2, \ldots, L_{m-1}$$

training tuples covered by this clause are denoted as T_i. The information required for T_i is calculated from T_i^+ and T_i^- which denote positive and negative tuples covered by the clause, respectively.

$$I(T_i) = -\log_2 \frac{|T_i^+|}{|T_i^+| + |T_i^-|} \tag{1}$$

If a literal L_m is selected and added, a new set of covered tuples T_{i+1} is created, then similar formula is given as

$$I(T_{i+1}) = -\log_2 \frac{|T_{i+1}^+|}{|T_{i+1}^+| + |T_{i+1}^-|} \tag{2}$$

From above, a heuristic used in FOIL is calculated as an amount of information gained when applying a new literal L_m;

$$Gain(L_m) = |T_i^{++}| \times (I(T_i) - I(T_{i+1})) \qquad (3)$$

T_i^{++} is the positive tuples included in T_i and extended in T_{i+1}. This heuristic function is used over all candidate literals, and the literal with the largest value is selected and added to the partial developed clause in inner loop.

2·2 Multiple-Instance Learning

For the supervised learning problem, we try to design and create algorithms that are able to generate model from training examples to predict correct labels of unseen data, and each instance of training examples has to be labeled beforehand. However, this framework may not be suitable for some applications. Dietterich et al. then proposed the extended framework for supervised learning to handle more ambiguities called *Multiple-Instance Learning* [Dietterich 97]. In the new framework, unlabeled instances are grouped into a bag labeled as positive or negative. A positive bag contains at least one positive instance, otherwise labeled as negative. From this set-up, target concept can be found from feature space where instances from various positive bags gather together. Dietterich et al. proposed an algorithm that tries to find target concept by first constructing a rectangle in the feature space, then reducing its size until it covers instances from positive bags only.

After this framework and algorithm were presented, various approaches were proposed. Some of them extended the existing supervised learning algorithm. Wang and Zucker applied k-NN algorithm for multiple-instance problem [Wang 00]. Chevaleyre and Zucker proposed generic framework for extending propositional rule learner to handle multiple-instance data [Chevaleyre 01]. They implemented the extension of Ripper. Gärtner et al. proposed a new kernel function for multiple-instance learning [Gärtner 02]. Maron et al. proposed the original approach for multiple-instance learning using Diverse Density (DD) [Maron 98]. This approach is applied in the proposed system. We then explain this approach in detail.

Diverse Density

Diverse Density (DD) algorithm aims to measure a point in an n-dimensional feature space for multiple-instance domains. DD at point p in the feature space shows how many *different* positive bags have an instance near p, and how *far* the negative instances are from p. Thus, DD value is high in the area where instances from various positive bags are located together, and rather far from instances from negative bags. It can be calculated as

$$DD(x) = \prod_i (1 - \prod_j (1 - exp(-\|B_{ij}^{+} - x\|^2))) \cdot \prod_i \prod_j (1 - exp(-\|B_{ij}^{-} - x\|^2)) \qquad (4)$$

where x is a point in feature space and B_{ij} represents j^{th} instance of i^{th} bag in training examples with + and − denoting positive and negative bags respectively. For the distance, the Euclidean distance is adopted

$$\|B_{ij} - x\|^2 = \sum_k (B_{ijk} - x_k)^2 \qquad (5)$$

Each instance is represented by a set of attributes and subscript k in B_{ijk}, x_k denotes k^{th} attribute of the instance.

In the previous approaches, several searching techniques were proposed for determining value of features or area in the feature space maximizing DD value.

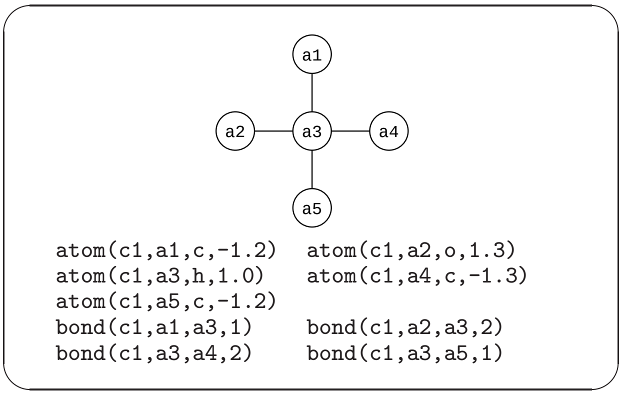
3. Proposed Method

We present top-down ILP system that is able to learn more efficiently hypotheses from set of examples, each consisting of several small parts, or when trying to predict class of data from the common substructure. The proposed system incorporates existing top-down ILP system (FOIL) and applies multiple-instance based measure to find common characteristics among parts of positive examples. This measure is then used as a weight attached to each part of the example and the common parts among positive examples are attached with high-valued weights. With these weights and heuristic function based on example coverage, the system generates more precise and higher coverage hypotheses from training examples. Next, we define multiple-part data, and then, explain modified heuristics.

3·1 Multiple-Part Data

In this section, we define multiple-part data and multiple-part learning problem.

Multiple-part data consists of at least one component with part-of relations between each component and the whole data as well as relations among parts. Because of flexibility of the first-order logic, there are many ways to denote multiple-part data. We set a



```
atom(c1,a1,c,-1.2)   atom(c1,a2,o,1.3)
atom(c1,a3,h,1.0)    atom(c1,a4,c,-1.3)
atom(c1,a5,c,-1.2)
bond(c1,a1,a3,1)     bond(c1,a2,a3,2)
bond(c1,a3,a4,2)     bond(c1,a3,a5,1)
```

Fig. 1 An example of multiple-part data

common way for data representation to make preprocessing easier. A part is denoted using only one predicate. The first two parameters denote the identification of data and part. The rest parameters are used for attributes. For denoting a relation between parts, we use one predicate for one relation in similar manner to a part. The predicate is written as

- *part(Data-ID, Part-ID, Attr-1, Attr-2, ...).*
- *relation(Data-ID, Part-ID*$_1$*, Part-ID*$_2$*, ...).*

For better understanding, we explain the multiple-part data using chemical compound structure data as shown in Figure 1. The predicate `atom` denotes a part of multiple-part data, while `bond` shows a relation between two parts (atoms).

Given a training example $\mathbb{E} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where x_i is a multiple-part data composing of $P_i \subseteq \mathbb{P}$ and $R_i \subseteq \mathbb{P} \times \mathbb{P}$ where $\mathbb{P}$ is a set of part, $\mathbb{R}$ is a set of relation between parts and y_i is a class (simply positive or negative). Multiple-part learning problem aims to approximate a function $f(P, R) = y$ where $P \subseteq \mathbb{P}$ and $R \subseteq \mathbb{R}$. This learning problem is considered a special case of supervised learning since it aims to find the class of data from substructure or subset of parts and relations. For example, in case of chemical compound data, we want to find substructures that are common among compounds with the same label or class, such as a group of atoms and bonds including their features. This problem is different from the traditional supervised learning that aims to predict class from the whole characteristics of data like predicting class of compound from its weight, or some special value computed.

3・2 *Modified Heuristic Function*

Heuristic function is used to control the way algorithm explores hypothesis space. FOIL (equation 1 and 3) adapts this function based on information theory that counts the number of positive and negative examples covered by partially developed clause. With this, FOIL selects the literal that covers many positive tuples but few negative tuples. In order to help heuristics select better literals, we apply DD value to each tuple, and we have to adapt heuristic function and the parts with high DD values are selected first, making the hypothesis cover common characteristics among parts from positive examples.

From equation 1, T_i^+ and T_i^- denote the set of positive and negative tuples respectively, as DD value can be used to show the importance of the part of data by representing each instance of multiple-part data as a bag and each part as an instance in the bag. The distance between two parts is calculated by first constructing a vector p for each part where p_i denotes *Attr-i* in the *part* predicate explained in the previous section. Then, equation 5 is used to calculate distance between two attribute vectors. From Figure 1, distance between `atom(c1,a1,c,-1.2)` and `atom(c1,a2,o,1.3)` is computed by constructing two vectors $[c, -1.2]$ and $[o, 1.3]$ and using equation 5. To compute DD value of each atom, distances between all atom pairs are calculated first. Then, x in equation 4 is assigned to the vector of atom being considered. $||B_{ij}^+ - x||^2$ and $||B_{ij}^- - x||^2$ are obtained from the computed distances.

We incorporate DD values to heuristic function by altering $|T_i^+|$ to be the sum of tuple DD values. If this sum is high, it means that the literal can cover more common parts among positive examples. Thus, the heuristic function is adapted as follows:

$$DD(t) = \frac{\sum_k (t_k)}{m}, t =< t_1, t_2, \ldots, t_m > \quad (6)$$

$$DD_s(T) = \sum_{t_i \in T} DD(t_i) \quad (7)$$

$$I(T_i) = -\log_2 \frac{DD_s(T_i^+)}{DD_s(T_i^+) + |T_i^-|} \quad (8)$$

$$Gain(L_m) = DD_s(T_i^{++}) \times (I(T_i) - I(T_{i+1}))(9)$$

This function weighs each part with DD value and uses the sum of these weights to select the literal, while the original heuristic function weighs all parts with the same value as 1. Nevertheless, we still use the number of negative tuples $|T_i^-|$ in the same way as the original heuristics, because we know that all parts of negative examples show the same strength. Therefore, it weighs all negative parts with value 1.

```
FindBestRule(Examples, Remaining)
  • Initialize $Beam$ and add a rule with empty body.
  • Do
    ○ $NewBeam \leftarrow \{\}$
    ○ For each clause $C$ in $Beam$
      ∗Generate $Candidates$ by selecting a possible literal and
       adding to $C$.
      ∗For each new clause $nC$ in $Candidates$
        ·Calculate $heuristic$ of $nC$ using DD values.
        ·Append $nC$ to $NewBeam$.
    ○ $Beam \leftarrow$ Best $BeamWidth$ clauses in $NewBeam$
    ○ $R \leftarrow$ Best clause in $Beam$
  • Until (Accuracy($R$) $> \varepsilon$ and PositiveCoverage($R$) $> \gamma$) or
    (Gain($R$) $\leq 0$)
  • Return $R$
```

Fig. 2 Algorithm for finding the best rule from the remaining positive examples.

From the above function, one problem is left to be considered. Each tuple may consist of more than one part. The algorithm has to calculate DD value of a relation among parts, e.g. a bond makes each tuple contains two atoms. We then have to select the weight to represent each tuple from DD value of the parts. We solve this problem by simply selecting average DD value in the tuple as the weight of tuple (equation 6).

3·3 Algorithm

From this modified function, we implement the prototype system called FOILmp (**FOIL** for **M**ultiple-**P**art data). This system basically uses the same algorithm as proposed in [Quinlan 90]. Nevertheless, in order to construct accurate hypotheses, beam search is applied so that the algorithm maintains a set of good candidates instead of selecting the best candidate at that time. This searching method enables the algorithm to backtrack to the right direction and finally get to the goal. Moreover, in order to obtain rules with high coverage, we define coverage ratio, and the algorithm is set to select only the rules covering positive examples higher than the coverage ratio. The modified subroutine for selecting rules is shown in Figure 2. There are two user-defined parameters: ε for the minimum accuracy and γ for the minimum positive example coverage.

4. Experiments

We conducted experiments on two datasets for SAR: Mutagenicity and Dopamine antagonist data. To evaluate performance of the proposed system, these experiments are conducted in ten-fold cross validation manner and we compare the results to the existing approaches.

4·1 Data

In this research, we aim to discover rules describing the activities of chemical compounds from their structures. Two kinds of SAR data were studied: mutagenesis data [Srinivasan 94] and dopamine antagonist data.

Mutagenicity data aims to test mutagenicity in nitroaromatic compounds which are often known to be carcinogenic and cause damage to DNA. These compounds are found in automobile exhaust fumes and are common intermediates used in chemical industry. In this dataset, 230 compounds were obtained from the standard molecular modeling package QUANTA. Two predicates (atm and bond) are used to denote each compound:

- *atm(comp, atom, element, type, charge)*, stating that there is the atom *atom* in the compound *comp* that has element *element* of *type* and partial charge *charge*.
- *bond(comp, atom1, atom2, type)*, describing that there is a bond of *type* between the atoms *atom1* and *atom2* in the compound *comp*.

The background knowledge in this dataset is already formalized in the form of multiple-part data, and thus, no preprocessing is necessary.

For dopamine antagonist activity data, the details of data and experiment can be found in [Nattee 04].

4·2 Experimental Results and Discussion

We first compare performance of FOILmp using all data which consist of 125 compounds for positive class and 63 compounds for negative class. In this experiment, beam search is applied in FOILmp by setting the beam size = 1 (hill-climbing search) and 3 subsequently. The results are compared to the existing results described in [Srinivasan 94]. Figure 3 shows performance tables.

From the performance tables, it can be seen that even FOILmp with hill-climbing search strategy can learn from this dataset better than PROGOL with accuracy 83.0% for FOILmp and 79.8% for PROGOL. When compared to the regression technique based on the model called *logM* [Srinivasan 94], FOILmp with the beam size = 3 still performs worse than the regression model that can predict at 89.9% whereas FOILmp can predict at only 87.2%. However, these experimental results show the advantage of FOILmp as follows: to use the regression model, a human expert is required to choose useful features to construct the model and the results are based on features diffi-

		Predicted		
		active	inactive	
Actual	active	100	25	125
	inactive	13	50	63
		113	75	188

(a) PROGOL [Srinivasan 94]

		Predicted		
		active	inactive	
Actual	active	114	11	125
	inactive	8	55	63
		122	66	188

(b) Regression [Srinivasan 94]

		Predicted		
		active	inactive	
Actual	active	100	25	125
	inactive	7	56	63
		107	81	188

(c) FOILMP (beam size=1)

		Predicted		
		active	inactive	
Actual	active	109	16	125
	inactive	8	55	63
		117	71	188

(d) FOILMP (beam size=3)

Fig. 3 Performance tables for Mutagenicity Data comparing FOILMP to PROGOL and the regression technique.

```
Rule1:
active(A) :- atm(A,B,C,D,E),D=27,
             atm(A,F,G,H,I),H=27,
             neq(E,I).
Accuracy = 92.5%   Coverage = 49.6%
This rule shows that the molecule contains
two atoms of Type 27 which means a carbon
atom that merges two six-numbered aromatic
rings. However, these two atoms has
different charge.

Rule2:
active(A) :- atm(A,B,C,D,E),E>0.81,
             atm(A,F,G,H,I),H=29.
Accuracy = 90.9%   Coverage = 16.0%
This rule is similar to Rule1 but the first
atom has charge greater than 0.81 and
the second atom is of Type 29.

Rule3:
active(A) :- atm(A,B,C,D,E),D=32,
             atm(A,F,G,H,I),I<-0.4,
             bond(A,F,J,K),neq(B,F),
             eq(C,G),eq(D,H).

Accuracy = 77.8%   Coverage = 5.0%
This rule shows that there are two atoms
of Type 32 which occurs in an amide group.
But the second atom has charge less than
0.4 and there is also a bond from this
atom to another atom.
```

Fig. 4 Examples of rules obtained by FOILMP

cult to be comprehended by other chemists. Figure 4 shows example of rules obtained from FOILMP. We found that FOILMP obtains rules with high coverage, such as, the first rule can cover around 50% of the all positive examples.

Table 1 shows experimental results on Mutagenicity data. Prediction accuracy on test examples using ten-fold cross validation is compared to the existing approaches (FOIL and PROGOL). It shows that the proposed method can predict more accurately than the existing approaches.

For Dopamine Antagonist data, we conducted tenfold cross validation to predict D1 activity by using D2, D3, and D4 as negative examples. However, we compare the experimental results with Aleph [Srinivasan 01] since PROGOL cannot perform well on this dataset. Aleph is ILP system based on inverse entailment and similar algorithm with PROGOL, but Aleph adopts several search strategies such as randomized search that help improve performance of the system. In this experiment, we set Aleph to use GSAT, one of the randomized search algorithms that can generate the best results compare to the other search strategies. Table 2 shows prediction accuracy computed for both positive and negative examples and for only the positive examples. The experiment results show that FOILMP predict more accurately than Aleph in both accuracy computation methods. Details of the results can be found in [Nattee 04].

Table 1 Ten-fold cross validation test comparing the accuracy on Mutagenicity data

Approach	Accuracy
The proposed method	0.82
PROGOL	0.76
FOIL	0.61

In summary, we found that FOILMP performs more accurately than the previous approaches, moreover, it can also obtain rules with higher coverage.

5. Related Work

In recent years, many researches were made to learn from chemical compound structure data because learning results can be applied directly to produce new drugs for curing some difficult diseases. Muggleton, Srinivasan and King [Srinivasan 94, King 95, Srinivasan 96] proposed the approach that applies PROGOL to predict several datasets including mutagenic-

Table 2 Ten-fold cross validation test comparing accuracy on Dopamine antagonist data

Approach	Accuracy (overall)	Accuracy (only positive)
FOILMP	96.12%	79.35%
Aleph	95.78%	75.84%

ity of chemical compounds used in our experiments.

King et al. also discussed whether propositional learner or ILP is better for learning from chemical structure [King 95]. Actually, the first-order representation can denote chemical structure without losing any information. Since denoting the relational data using propositional logic is beyond its limit, some special techniques are required, e.g., for relations among parts, we may use only average value of features or use domain-related knowledge to calculate a new feature for categorization. However, a propositional learner can perform better than a learner using first-order representation because ILP learners have some restrictions from the logic theory. However, comparing only accuracy may not be good assessment because chemist's natural inclination is related to chemical structure and the learning results from ILP is comprehensible to chemists.

However, King et al. [King 95] reviewed four case studies related to SAR studies: inhibition of dihydrofolate reductase by pyrimidines, inhibition of dihydrofolate reductase by triazines, design of tacrine analogues, and mutagenicity of nitroaromatic and heteroaromatic compounds. The experimental results are compared with two propositional learner: *Regression*, a linear regression technique and *CART*, a decision tree learner. They found that with these chemical structure data, propositional learners with limited number of features in one instance are sufficient to all problems. However, when more complex chemical structures and background knowledge are added, propositional representations become unmanageable. Therefore, first-order representations would provide more possibility with more comprehensible results.

From multiple-instance learning problem, Wiedmann et al. [Weidmann 03a, Weidmann 03b] proposed an extension of the process for determining labels of a bag. This is a generalization of the assumption used in multiple-instance learning where a bag is labeled as positive if there exists at least one positive instance. The author proposed the idea of two-level classification for handling generalized multiple-instance problem. However, the research focuses on learning from propositional data representation.

McGovern and Jensen [McGovern 03] proposed extension of diverse density and chi-squared measure to relational data using the metric based on the found maximal common subgraph. This work is similar to the proposed approach that also aims to handle ambiguous relational data. Because each bag is represented in form of graph, relations between bags cannot be denoted. Nevertheless, the proposed metric is interesting enough to apply modified version of the metric to our proposed method.

Zucker et al. [Chevaleyre 00, Zucker 01] applied multiple-instance learning on Mutagenesis data and proposed the extension of decision tree and propositional rule learner for multiple-instance learning. The molecular properties, such as *logP* and *lumo*, are applied in the experiments, but comparison to the proposed method is difficult to make. However, the dataset is transformed into propositional representation due to the user-defined setting. It limits usage of relations in the hypotheses.

6. Conclusion and Future Works

We presented extension of FOIL for handling multiple-part data more efficiently by using Diverse Density from multiple-instance learning to evaluate parts and parts with common characteristics among positive examples have high-valued weight and help enable the searching process to generate better results. We conducted experiments on chemical compound data for structure-activity relationship studies. The experimental results showed that the proposed method can predict test examples more accurately than the previous ILP approaches.

For future works, scaling factor of the feature should be considered in heuristic value calculation in order to produce more suitable heuristics. Because the proposed approach works only in the top-down ILP system such as FOIL, it is better to adopt this approach to other kinds of system such as the one with bottom-up approach. We plan to evaluate the proposed system to other domains. Moreover, as the proposed approach mainly focuses on a part, it is difficult to incorporate relational information into the heuristic function. We plan to overcome this limitation in our future works.

♢ References ♢

[Chevaleyre 00] Chevaleyre, Y. and Zucker, J.-D.: Solving multiple-instance and multiple-part learning problems with decision trees and decision rules: Application to the mutagenesis problem, Technical report, LIP6-CNRS, University Paris VI (2000)

[Chevaleyre 01] Chevaleyre, Y. and Zucker, J.-D.: A Framework for Learning Rules from Multiple Instance Data, in *Proc. 12th European Conf. on Machine Learning*, Vol. 2167 of *LNCS*, pp. 49–60, Springer (2001)

[Dietterich 97] Dietterich, T. G., Lathrop, R. H., and Lozano-Perez, T.: Solving the Multiple Instance Problem with Axis-Parallel Rectangles, *Artificial Intelligence*, Vol. 89, No. 1-2, pp. 31–71 (1997)

[Gärtner 02] Gärtner, T., Flach, P. A., Kowalczyk, A., and Smola, A. J.: Multi-Instance Kernels, in *Proc. 19th International Conf. on Machine Learning*, pp. 179–186, Morgan Kaufmann (2002)

[King 95] King, R. D., Sternberg, M. J. E., and Srinivasan, A.: Relating Chemical Activity to Structure: An Examination of ILP Successes, *New Generation Computing*, Vol. 13, No. 3-4, pp. 411–433 (1995)

[Maron 98] Maron, O. and Lozano-Pérez, T.: A Framework for Multiple-Instance Learning, in Jordan, M. I., Kearns, M. J., and Solla, S. A. eds., *Advances in Neural Information Processing Systems*, Vol. 10, The MIT Press (1998)

[McGovern 03] McGovern, A. and Jensen, D.: Identifying Predictive Structures in Relational Data Using Multiple Instance Learning, in *Proceedings of the 20th International Conference on Machine Learning (ICML-2003)* (2003)

[Nattee 04] Nattee, C., Sinthupinyo, S., Numao, M., and Okada, T.: Knowledge Discovery using Inductive Logic Programming in Dopamine Antagonist Structure Data, Technical report, I.S.I.R., Osaka University (2004), `http://libra.ai.sanken.osaka-u.ac.jp/usr/cholwich/techreport.pdf`

[Quinlan 90] Quinlan, J. R.: Learning Logical Definitions from Relations, *Machine Learning*, Vol. 5, No. 3, pp. 239–266 (1990)

[Srinivasan 94] Srinivasan, A., Muggleton, S., King, R., and Sternberg, M.: Mutagenesis: ILP experiments in a non-determinate biological domain, in Wrobel, S. ed., *Proc. 4th International Workshop on Inductive Logic Programming*, Vol. 237, pp. 217–232, Gesellschaft für Mathematik und Datenverarbeitung MBH (1994)

[Srinivasan 96] Srinivasan, A., Muggleton, S., Sternberg, M. J. E., and King, R. D.: Theories for Mutagenicity: A Study in First-Order and Feature-Based Induction, *Artificial Intelligence*, Vol. 85, No. 1-2, pp. 277–299 (1996)

[Srinivasan 01] Srinivasan, A.: The Aleph Manual (2001), `http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/`

[Wang 00] Wang, J. and Zucker, J.-D.: Solving the Multiple-Instance Problem: A Lazy Learning Approach, in *Proc. 17th International Conf. on Machine Learning*, pp. 1119–1125, Morgan Kaufmann, San Francisco, CA (2000)

[Weidmann 03a] Weidmann, N. B.: Two-level Classification for Generalized Multi-instance Data (2003)

[Weidmann 03b] Weidmann, N., Frank, E., and Pfahringer, B.: A Two-level Learning Method for Generalized Multi-instance Problems, in *Proceedings of the European Conference on Machine Learning (ECML-2003)*, pp. 468–479 (2003)

[Zucker 01] Zucker, J.-D.: Solving Multiple-Instance and Multiple-Part Learning Problems with Decision Trees and Rule Sets. Application to the Mutagenesis Problem, in *Proceedings of Canadian Conference on AI 2001*, pp. 204–214 (2001)

〔担当委員：山川 宏〕

Received January 15, 2004.

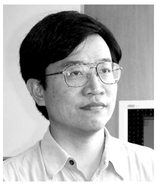
Author's Profile

Cholwich Nattee (Member)

Dr. Nattee is a research associate in the Department of Architecture for Intelligence, the Institute of Scientific and Industrial Research, Osaka University. He received his B.Eng. degree in computer engineering in 1998 from Chulalongkorn University, Thailand and his D.Eng. degree in computer science from Tokyo Institute of Technology in 2004. His current research interests include Machine Learning, Knowledge Discovery in Databases and Data Mining.

Sukree Sinthupinyo

Dr. Sinthupinyo is a postdoctoral researcher in the Department of Architecture for Intelligence, the Institute of Scientific and Industrial Research, Osaka University. His current research is Knowledge Discovery in Databases, Multi-class Inductive Logic Programming. He worked as a system engineer in the Department of Dentistry, Mahidol University, Thailand (1995-1996) and then worked as a lecturer in the Department of Computer Science, Faculty of Science and Technology, Thammasat University, Thailand (1996-2003).



Masayuki Numao (Member)

Dr. Numao is a professor in the Department of Architecture for Intelligence, the Institute of Scientific and Industrial Research, Osaka University. He received a bachelor of engineering in electrical and electronics engineering in 1982 and his Ph.D. in computer science in 1987 from Tokyo Institute of Technology. He was working in the Department of Computer Science, Tokyo Institute of Technology from 1987 to 2003, and was a visiting scholar at CSLI, Stanford University from 1989 to 1990. His research interests include Artificial Intelligence and Machine Learning. He is a member of Information Processing Society of Japan, Japanese Society for Artificial Intelligence, Japanese Cognitive Science Society, Japan Society for Software Science and Technology, and the American Association for Artificial Intelligence.