2004－ICS－138　(28)
2004／12／6

Multiple-Instance Learning Based Heuristics for Mining Chemical Compound Structure

Cholwich Nattee,† Sukree Sinthupinyo,† Masayuki Numao†
and Takashi Okada††

Inductive Logic Programming (ILP) is a combination of inductive learning and first-order logic aiming to learn first-order hypotheses from training examples. ILP has a serious bottleneck in an intractably enormous hypothesis search space. This makes existing approaches perform poorly on large-scale real-world datasets. In this research, we propose a technique to make the system handle an enormous search space efficiently by deriving qualitative information into search heuristics. Currently, heuristic functions used in ILP systems are based only on quantitative information, e.g. number of examples covered and length of candidates. We focus on a kind of data consisting of several parts. The approach aims to find hypotheses describing each class by using both individual and relational features of parts. The data can be found in denoting chemical compound structure for Structure-Activity Relationship studies (SAR). We apply the proposed method to extract rules describing chemical activity from their structures. The experiments are conducted on a real-world dataset. The results are compared to existing ILP methods using ten-fold cross validation.

1. Introduction

Inductive Logic Programming (ILP)[2] aims to learning first-order rules from examples and background knowledge. ILP combines inductive learning and first-order logic to overcome limitations of inductive learning which is based on propositional logic or attribute-value language. First-order logic representation provides capability to handle data which consist of complicated relations. Such as, data that is scattered over many tables with relations among tables. Though, propositionalization allows attribute-value learning system to handle this kind of data, it causes the number of features to become larger and difficult to be managed. Another advantage of ILP is comprehension. Learning results are given in form of first-order rules which are understandable by human. Nevertheless, a bottleneck of ILP is an intractably enormous search space caused from flexibility of first-order logic.

To reduce the search space size, two techniques are mainly used: language bias and informed search. Language bias aims to define description of learning results to limit possibility in candidate generation. Informed search uses heuristic function to cut unnecessary parts from searching process. In this research, we focus on using heuristic function to limit search space and lead to appropriate rules. Heuristic functions used in the existing ILP systems are based only on quantitative information, such as, the number of examples covered by the considered candidate or length of the candidate. This causes the existing approaches sometimes perform worse than attribute-value learners. To overcome the shortcoming, qualitative information is required, such as the quality of the covered examples should be considered.

We therefore propose an improved heuristic function based on Multiple-Instance Learning (MIL)[1]. MIL is an extended two-class propositional learning approach for data that cannot be labeled individually, albeit several instances of data are gathered and labeled as a bag. Each positive bag may consist of both positive and negative instances. Nevertheless, MIL aims to obtain models that predict instances not bags, thereby rendering itself similar to supervised learning where there are noises in positive examples. Algorithms from MIL solve the ambiguity by using similarity or distance among instances within feature space. Using distance, target concept is an area where several instances from various positive bags are located together and that area is far from instances from negative bags. We derive this basic idea

† The Institute of Scientific and Industrial Research, Osaka University
†† School of Science and Technology, Kwansei Gakuin University

of MIL to evaluate quality of first-order objects consisting of multiple parts. Each object is considered as a bag containing several parts. We evaluate each part using MIL based measure using similarity or distance among parts. Therefore, the part whose features are common compared to parts from various positive objects is evaluated as high value. We evaluate all parts and incorporate obtained values as weights into heuristic function. Then, hypothesis candidate covering high-valued parts is evaluated as high value and selected first.

The paper is organized as follows. In the next section, we present details of proposed method that improves heuristic function used in ILP to efficiently learn rules from objects consisting of multiple parts. We focus on classifying chemical compound according to their structures. The experiments conducted on real-world datasets are then presented in Section 4. Finally, we conclude the paper and consider future directions in Section 5.

2. Background

2.1 FOIL

FOIL[3] is a top-down ILP system for learning function-free Horn clause definitions of a target predicate using background predicates. The learning process in FOIL starts with training examples containing all positive and negative examples, constructs a function-free Horn clause (a hypothesis) to cover some of the positive examples, and removes the covered examples from the training set. Next, it continues to search for the next clause. When the clauses covering all the positive examples have been found, they are reviewed to eliminate any redundant clauses and re-ordered so that all recursive clauses follow the non-recursive ones.

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

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

the training examples covered by this clause are denoted as T_i. The information required for T_i is given as

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

If a literal L_m is selected and yields a new set T_{i+1}, then the similar formula is given as:

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

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

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

T_i^{++} in this equation is the positive example extended in T_{i+1}.

This heuristic function is used over every candidate literal and a literal with largest value is selected. The algorithm will continue until generated clauses cover all positive examples.

2.2 Diverse Density

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

$$DD(x) = \prod_i P(x|B_i^+) \prod_i P(x|B_i^-) \quad (4)$$

$$P(x|B_i^+) = 1 - \prod_j (1 - e^{-\|B_{ij}^+ - x\|^2}) \quad (5)$$

$$P(x|B_i^-) = \prod_j (1 - e^{-\|B_{ij}^- - x\|^2}) \quad (6)$$

where x is a point in the feature space and B_{ij} represents the j^{th} instance of the i^{th} bag in training examples. For the distance, the Euclidean distance is adopted

$$\|B_{ij} - x\|^2 = \sum_k (B_{ijk} - x_k)^2 \quad (7)$$

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

3. Using ILP in Structure-Activity Relationship Studies

The studies of Structure-Activity Relationship aim to find structures in chemical com-

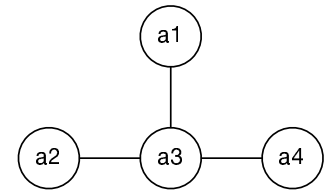
pounds describing their characteristics or activities. Knowledge discovered will be useful for developing new drugs. In recent years, advance in High Throughput Screening (HTS) technology has produced vast amount of SAR data. Once the rules which predict the activities of existing SAR data are found, it significantly helps the screening process. Since each compound consists of multiple parts, we then gain benefits from the improved heuristics for a large-scale data.

The proposed approach incorporates existing top-down ILP system (FOIL) and applies MIL based measure to find common features among parts of positive compounds. The measure is then used as the 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 explain first-order representation used in the paper. After that, the improved heuristic function is then be explained.

3.1 First-Order Representation

To apply ILP for SAR studies, training examples are required to be denoted in form of the first-order logic. Because of flexibility of first-order logic, there are many ways to denote data. We set a 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`$_2$`, ...)` and `relation(Data-ID, Part-ID`$_1$`, Part-ID`$_2$`, ...)`.

Each chemical compound is considered a first-order object. We denote them based on their structure using two predicates: `atom(Compound, Atom, Element)` and `bond(Compound, Atom`$_1$`, Atom`$_2$`, Type)`. Features related to atom and bond are put as parameters of predicate. Predicate `atom` denotes an `Atom` of `Element` in a `Compound`. Predicate `bond` denotes a bond of `Type` consisting of two atoms (`Atom`$_1$ and `Atom`$_2$). Figure 1 shows an example of first-order representation. More features can be used to represent atoms and bonds in real-world dataset. In this research, we consider an atom as a part of compound, and a bond is a relation between two parts. In other words, a compound is a group of atoms relating to each other. An atom as a part is used in the improved heuristic function explained in the following section.




```
atom(c1,a1,c)        atom(c1,a2,c)
atom(c1,a3,c)        atom(c1,a4,c)
bond(c1,a1,a3,1)     bond(c1,a2,a3,1)
bond(c1,a3,a4,1)
```

Fig. 1 Examples of first-order representation of chemical compound

3.2 Improved Heuristics

The original heuristic function used in FOIL is based on information theory. Partially developing hypothesis is evaluated by using the number of positive and negative tuples covered. Hence, FOIL selects the literal that covers many positive tuples but few negative tuples. To make heuristics select better literals, we derive DD to evaluate literals. From Equation 1, T_i^+ and T_i^- denote set of positive and negative tuples respectively. We consider each compound as a bag and each part of compound as an instance in the bag (Figure 3). DD of parts are then computed and used as a weight attached to each part. Therefore, a part with common features among parts from positive compound is given a high-valued weight. The weights are incorporated to the heuristic function by altering $|T_i^+|$ to be the sum of weight. If heuristic value is high, it means that the candidate covers many common parts among positive compounds. The heuristic function is modified as follows.

$$DD_s(T) = \sum_{T_i \in T} DD(T_i) \tag{8}$$

$$I(T_i) = -\log_2 \frac{DD_s(T_i^+)}{DD_s(T_i^+) + |T_i^-|} \tag{9}$$

Nevertheless, we still use the number of negative tuples $|T_i^-|$ in the same way as the original

FindBestRule(**Examples**, **Remaining**)
- Initialize $Beam$ with an empty rule.
- Do
  - $NewBeam \leftarrow \{\}$
  - For each clause C in $Beam$
    - Generate $Candidates$ by adding all possible literals 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) $> \gamma$
- Return R

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

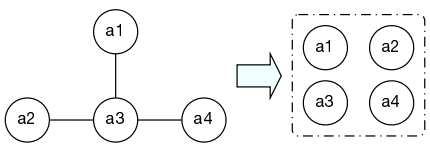


Fig. 3 A bag representation of compound

heuristics, since we know that all parts of negative examples show the same strength. Therefore, it is similar to weighing all negative parts with value 1.

3.3 Algorithm

From the modified function, we implement the prototype system called FOILmp. The system basically uses the same algorithm as FOIL. However, to construct accurate hypotheses, beam search is applied. The algorithm maintains a set of good candidates instead of selecting best candidate at that time. This searching strategy makes the algorithm possible to backtrack to the right direction and finally get to the goal. Moreover, to obtain rules with high coverage, we define the coverage ratio, and the algorithm is set to select only rules covering positive examples higher than that ratio. The modified subroutine for selecting rules is shown in Figure 2. There are two user-defined parameters: ε for minimum accuracy and γ for minimum positive example coverage.

4. Experiments and Discussions

4.1 Datasets

We aim to discover rules describing the activities of dopamine antagonist. Dopamine antagonist dataset contains 1,366 compounds separated into four classes; D1, D2, D3 and D4. They are obtained from MDDR database of MDL Inc. Each compound is originally

```
d1(A) :- atom(A,B,C,D,E,F), E>=3.7, F=3.3,
         bond(A,L,B,H,M,N),
         bond(A,G,H,I,J,K), K=1.5,
         bond(A,O,B,P,Q,R), not_equal(H,P).
```

The rule shows a compound contains an atom `B` with distance to nearest oxygen is larger than $3.7\mathring{A}$, and distance to nearest nitrogen is $3.3\mathring{A}$. From `B`, there are two bonds to `H` and `P`. There is another bond from `H` to `I` of length $1.5\mathring{A}$.

Fig. 5 Rules obtained by FOILmp using data for D1 activity.

described in term of the position in three-dimensional space. Each atom is denoted by element type. Each bond is represented by relation between two atoms and bond type. After preprocessing, three kinds of predicates are used to denote a compound as shown in Figure 4.

4.2 Comparing to existing ILP approaches

We conduct ten-fold cross validation to predict D1, D2, D3, and D4 activities and compare the experimental results with Aleph[5]. Aleph is an ILP system based on inverse entailment. It has adopted several search strategies, such as randomized search which helps improve the performance of the system. In this experiment, we set Aleph to use GSAT[4] where the best results can be generated. Table 1 shows the prediction accuracy computed for both positive and negative examples, and then, for only the positive examples. The table also shows the results of significance test using one-tailed paired t-test. The experimental results show that FOILmp predicts more accurately than Aleph in both accuracy computation methods. The significance tests also show the confidence level in the difference between accuracy. Figure 5 and 6 show the details of rules obtained by FOILmp. We also found that FOILmp generates rule with higher coverage than Aleph.

4.3 Comparing to different parts

In the previous experiment, an atom is used as a part of compound. Its features are then used to compute DD for weighing. We can consider a compound composing of different kind of part, and features of that part can be used to compute DD for weighing. In this experiment, we consider two adjacent bonds as a part of compound. Thus, a new predicate `twobond(compound, twobond,`

`atom(compound, atom, element, o-connect, o-min-dist, n-min-dist)` – describing an atom `atom` in `compound` with element `element`. It forms a bond with oxygen atom if `o-connect` is 1 and has distance `o-min-dist` and `n-min-dist` to the nearest oxygen and nitrogen atom respectively.

`bond(compound, atom1, atom2, bondtype, length)` – describing a bond `bond` in compound `compound`. This bond links atom `atom1` and atom `atom2` together with type `bondtype` and length `length`.

`link(compound, atom1, atom2, length)` – describing a relation `link` in compound `compound`. It links atom `atom1` and atom `atom2` with length `length`.

Fig. 4 Predicates used to describe dopamine antagonist compound.

Table 1 Ten-fold cross-validation test comparing the accuracy on dopamine antagonist data; Superscripts denote confidence levels for the difference in accuracy between FOILMP and Aleph, using a one-paired t-test: * is 95.0%, ** is 99.0%; no superscripts denote confidence levels below 95%.

Activity	FOILMP		Aleph	
	Accuracy(%) (overall)	Accuracy(%) (only positive)	Accuracy(%) (overall)	Accuracy(%) (only positive)
D1	97.0	85.5	96.0*	78.6**
D2	88.1	79.1	86.4*	70.5*
D3	93.4	78.4	93.1	75.1*
D4	88.4	85.1	87.6*	83.2*

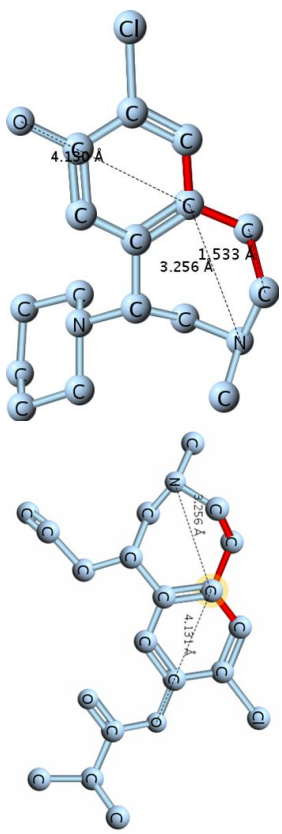


Fig. 6 Structure of compound specified by rule in Figure 5

`bond1, bond2)` is generated and included into the dataset. Figure 7 shows a bag representation which is different from one shown in Figure 3.

However, to compute DD, features of `twobond` are required. As a `twobond` composes of two bonds, features of bonds and atoms related to those bonds are used. In other words, we construct a new feature space for `twobond` based on features of bonds and atoms. This approach is useful when only features of atom or bond are unable to discriminate positive and negative compounds. For instance, if feature of atom is only an element type, they can be found in all compounds, such as carbon, oxygen or nitrogen. One way to solve this limitation is to append some special features like one used in the previous section. The other way is to consider a new part composing of simple parts as `twobond`. From this experiment, different rule is generated as shown in Figure 8 and 9.

5. Conclusion and Future works

We have presented an improved heuristic function for a data consisting of multiple parts. Diverse Density, a measure for MIL data, is applied to weigh parts, so that parts with common features among positive compounds have high-valued weights. The weights representing

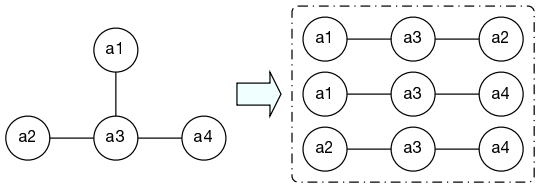


Fig. 7 A bag representation when considering two adjacent bonds as a part.

```
d1(A) :- twobond(A,B,C,D), bond(A,C,E,F,G,H),
         H<1.3, bond(A,D,E,I,J,K), K<1.5,
         K>=1.4, twobond(A,L,M,N),
         twobond(A,O,M,P),
         N\==P, D\==P, C\==P.
```

The rule shows a compound contains two adjacent bonds, `C` of length shorter that 1.3$\AA$, `D` of length between 1.4 and 1.5$\AA$ and three adjacent bonds `M`, `N`, and `P`.

Fig. 8 Rules obtained by FOILMP using `twobond` as a part.

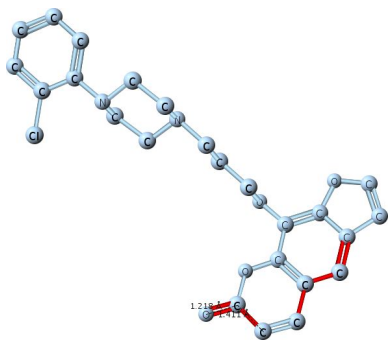


Fig. 9 Structure of compound specified by rule in Figure 8

quality of examples enable ILP to cut off unnecessary searching paths from an enormous search space and produce more efficient rules.

For future works, scaling factor of features should be considered in DD computing to produce more suitable heuristics. We plan to evaluate the proposed approach on other domains.

Acknowledgments

This research was supported by the Active Mining Project (Grant-in-Aid for Scientific Research on Priority Areas, No.759).

References

1) 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).

2) Muggleton, S. and Raedt, L. D.: Inductive Logic Programming: Theory and Methods, *Journal of Logic Programming*, Vol. 19,20, pp. 629–679 (1994).

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

4) Selman, B., Levesque, H. J. and Mitchell, D.: A New Method for Solving Hard Satisfiability Problems, *Proceedings 10th National Conference on Artificial Intelligence*, pp. 440–446 (1992).

5) Srinivasan, A.: The Aleph Manual (2001). http://web.comlab.ox.ac.uk/oucl/research/areas-/machlearn/Aleph/.