An Application of Rough Sets to Monk's Problems Solving

Duoqian Miao[1] and Lishan Hou[2]

[1] Dpt. of Computer Science, Tongji University or
Tongji Branch, National High Performance Computing Center,
Shanghai, 200092, P. R. China, miaoduoqian@163.com
[2] Dpt of Mathematics, Shanxi University,
Taiyuan 030006, P. R. China, hlslisa@163.com

Abstract. In this paper, the main techniques of inductive machine learning are united to the knowledge reduction theory based on rough sets from the theoretical point of view. And then the Monk's problems are resolved again employing rough sets. As far as accuracy and conciseness are concerned, the learning algorithms based on rough sets have remarkable superiority to the previous methods.

1 Introduction

During the 2nd Europe Summer School on Machine Learning, held in Coresendonk Priory of Belgium in 1991, many popular machine learning algorithms at that time were discussed extensively. Which algorithm would be optimal? And did there exist some relations among the algorithms?

As a consequence of these confusions, researchers having attended to that conference created three problems (Monk's problems)[2,4]. They are different in data scale, target concept, with and without noise in the training examples.

However, the comparison to various inductive learning techniques was limited in learning results, learning efficient and so on. The inherent connections among the methods weren't explained from the theoretical angle. In this paper, several representative machine learning algorithms are compared to rough sets and Monk's problems are analyzed and dealt with again. Comparison to further conclusions made in accuracy and conciseness of rules is inspiring.

2 Analyzing Theoretically

In 1982, Prof. Pawlak, put forward that rough sets could be used to study the representation, learning and induction of the uncertain, incomplete and imprecise information.

For the sake of illustration, let us review some basic points of Rough Sets firstly.

Definition 1. Let U be a universe and P, Q be equivalence relations on U, then $POS_P(Q) = \bigcup \underline{P}(X), X \in U/Q$ is called position region of Q employing P in U.

G. Wang et al. (Eds.): RSFDGrC 2003, LNAI 2639, pp. 138–145, 2003.

Definition 2. $DT = \langle U, C \cup D, V, f \rangle$ is called decision table, in which C and D are condition attribute set and decision attribute set respectively, $C \cap D = \emptyset$, U is universe, $V = \cup V_a \left(a \in C \cup D\right)$, V_a is value set of a, $f$is information function, $f{:}U \times \left(C \cup D\right) \rightarrow V$and $f_x\left(a\right) = a\left(x\right)$ for every $x \in U$and $a \in C \cup D$.

Knowledge reduction is one of the most important features of rough sets in terms of previous methods. It is necessary especially deposing a large scale data, otherwise the cost bought about by superfluous knowledge is very huge in both time and space.

2.1 Relations to ID-Family [6,7]

Different knowledge representations lead to different knowledge reduction algorithms. At present there exist several popular reduction algorithms such as X. H. Hu algorithm, algorithms based on Pawlak's attribute importance, algorithms based on discernibility matrix and its improvement, algorithms based on information entropy and so on.

X. H. Hu algorithm is:

```
$Input\ \ DT = \langle U, C \cup D, V, f \rangle$
$Output\ \ \{B \subseteq C | POS_B\left(D\right) = POS_C\left(D\right)\}$
1) Compute relative core $Co$.
2) $B \Leftarrow Co$.
3) For any $c \in C - B$, compute $Sig\left(c, B, D\right)$.
If $Sig\left(c', B, D\right) = \max_{c \in C-B} \{Sig\left(c, B.D\right)\}$, then $B \Leftarrow B \cup \{c'\}$.
4) If $POS_B\left(D\right) = POS_C\left(D\right)$, output $B$ and end; else go to 3).
```

in which:

$$Sig\left(c, B, D\right) = \frac{card\left(POS_{B \cup \{c\}}\left(D\right)\right) - card\left(POS_B\left(D\right)\right)}{card\left(POS_C\left(D\right)\right)}$$

The difference between X.H. Hu algorithm and algorithm based on Pawlak's attribute importance is the definition of the function $Sig\left(c, B, D\right)$. In the latter,

$$Sig\left(c, B, D\right) = \frac{card\left(POS_{B \cup \{c\}}\left(D\right)\right) - card\left(POS_B\left(D\right)\right)}{cardU}$$

By computing simply, we can prove that the orders of attribute importance generated from the two algorithms are same. As a result, the two ideas are equivalent.

ID3 is one of the most important models in inductive machine learning. It bases mainly on the partition of space, and this partition is limited to the structure of decision tree. According a given strategy, chose an attribute a and part U into $\left\{T_a^1, \cdots, T_a^n\right\}$, in which the superscript k of T will be referred to as a label of a in V_a. Successively, take T_a^k as new universe (example sets) and redo the process mentioned above until all the examples belong to the same decision classification. Thus, we get a decision tree.

According to rough sets, U can be parted by every attribute, $U/a, U/b, \cdots$ $a, b \cdots \in C$, and X. H. Hu algorithm just bases on this partition. For the sake of illustration, let us take attribute a as an example. ${}^{U}\!/_{a}$ is corresponding to $\{T_a^1, \cdots, T_a^n\}$ of the decision tree and $T_a^k \in {}^{U}\!/_{a}$, $k = 1, \cdots, n$, in which $n = card\left({}^{U}\!/_{a}\right)$. Therefore, creating a decision is equivalent to partitioning the universe by attributes substantively [5]. So we can say, ID3 is specialization of rough sets under some restrictions, and equivalent of existing heuristic reduction algorithms.

On the basis of ID3, some new algorithms suiting to different requests have been developed through using some strategies. For instance, ID3 with windowing, ID5R that is an incremental decision tree learning algorithm, IDL that uses some heuristics to stimulate a bi-directional search for a tree [3].

2.2 Relations to AQ-Family [4,10]

Discernibility matrix is an important mutation of rough sets. It transformed the reduction problem of a database into the simplification problem of a matrix due to introducing algebra theory. Skowron has proved that the theory of Discernibility matrix is equivalent to rough sets[9].

Let $DT = \langle U, C \cup D, V, f \rangle$ be a decision table, whose corresponding Discernibility matrix is defined as follows:

$$M\left(DT\right) = \left(c_{ij}\right)_{n \times n}, \quad n = card\left(U\right),$$

in which,

$$c_{ij} = \begin{cases} \emptyset & [x_i]_{IND(D)} = [x_j]_{IND(D)} \\ \{a \in C | a\left(x_i\right) \neq a\left(x_j\right)\} & otherwise \end{cases}.$$

In such a way, study to decision table is shifted to discernibility matrix. For every attribute, higher frequency in matrix, more examples are distinguished by it, more importance. The detailed description of algorithm is omitted here.

Comparisons with AQ algorithms are in order.

In fact, selecting seed example in AQ is equivalent to selecting a line from the discernibility matrix. According to the constitution of discernibility matrix, if some element of selected column is empty, its corresponding example belongs to positive examples for a consistent table; removing positive examples in AQ is equivalent to removing corresponding rows of discernibility matrix. The evaluate function is due to the frequency of the attribute in the line. It can be proved that it equals to AQ algorithms reducing directly attribute value without dealing with attributes.

In the same way, as a basic method, AQ-algorithm lead to many effective algorithms appropriate to specific types of problems by adding hypothesis-driven or combining genetic algorithm[4].

Information entropy is another representation of knowledge. Its idea is almost same with X. H. Hu's, but the use of probability leads a higher efficiency.

Table 1. Rules by RS to Monk-1

U	a1	a2	a5	D	Rule
1	1	1	*	1	if a1=1 and a2=1 then d=1
2	*	*	1	1	if a5=1 then d=1
3	2	2	*	1	if a1=2 and a2=2 then d=1
4	3	3	*	1	if a1=3 and a2=3 then d=1

3 Comparisons and Analysis

Monk's problems were created specially to test the capability of learning algorithms. And we will make experiments on Monk's problems with the algorithms mentioned above so as to compare and analyze.

3.1 Monk-1 Problem

In Monk-1, there are 124 training examples, 62 positive and 62 negative and without any noise. Its target concept is "(a1=a2) or (a5=1)".

Employment of Rough Sets. After reducing the attributes, the core is {a1, a2, a5} and the reduct is just the core; Then reduce the superfluous attribute values. The rules which belong to the positive are showed in Table 1: As you know, the value sets of a1 and a2 are both {1, 2, 3}, so we can merge rule 1, 2, 4 together. In other words, the learning result to Monk-1 by rough sets can be described as (a1=a2) or (a5=1), which is coincident entirely with the target concept. It is exciting that the accuracy accounts for 100% at the test sets (216 positive and 216 negative).

Employment of AQ-Family. We are going to handle Monk-1 employing AQ17-HCI, AQ15-GA and AQR [2].

AQ17-HCI is a module employed in the AQ17 attribute based multi-strategy constructive learning system. This model implements a new iterative constructive induction capability in which how attributes are generated based on the analysis of the hypotheses produced in the previous iteration. Rule is (pos16=false) in which pos16 is attribute constructed from the original ones, or intermediate ones, as defined below:

c01⟨:: (a1=1) & (a2=2,3) & (a5⟩1)
c05⟨:: (a1=2) & (a2=1,3) & (a5⟩1)
c08⟨:: (a1=3) & (a2=1,2) & (a5⟩1)
c10⟨:: (a1=1) & (a2=1)
c12⟨:: (a5=1)
c13⟨:: (a1=2) & (a2=2)
c15⟨:: (a1=3) & (a2=3)
pos⟨:: (c10=false) & (c12=false) & (c13=false) & (c15=false)
neg⟨:: (c01=false) & (c05=false) & (c08=false)

Table 2. Results by ID-family to Monk-1

	nodes	leaves	accuracy
ID3	13	28	98.6%
ID3 no windowing	32	62	83.2%
ID5R	34	52	79.7%
IDL	36	26	97.2%

Size embodies the conciseness of rules; Nonempty leaf includes the classification information, thus the number of nonempty leaves equals to that of rules.

Actually, (pos16=false) means any one of (c10=false), (c12=false), (c13=false) and (c15=false) is valid. It is coincident with the target concept and the accuracy is also 100%. But it is easy to see that the rule of AQ17-HCI is quite obscure and can't be used to make decisions directly.

AQ15-GA is the fusion of all subsets of a given attribute set. Each of the selected attribute subsets is evaluated by invoking AQ15 and measuring the recognition rate of the rules produced. The approach traverses the whole space of subset. Huge cost of computing brings about excellent results and its accuracy is 100% too. The AQR algorithm is an implementation of the AQ-family. It produces a rule for each decision class. Monk-1 is a two-class problem, so learning rule is below:

(a2=1) & (a1=1) or (a5=1) or (a3=1) & (a2=2) & (a1=2) or (a2=2) & (a1=2) or (a6=1) & (a2=3) & (a1=3) or (a6=2) & (a1=3) & (a2=3). class '1'.

This rule includes 5 attributes, a1, a2, a3, a5 and a6. But the reduct only includes 3 attributes a1, a2 and a5 according to rough sets, that is, 3 attributes are necessary to keep the capacity of classifying the data. In the rule of the AQR algorithm, there are two irrelevant attributes, so the rule maybe either contain superfluous information or describe the concept too strictly. All these degrade the ability of generation. In this approach, the accuracy is 95.9% and lower than that of the two noted above.

Employment of ID-Family. ID-family is a series of algorithms derived from decision trees via introducing some strategies. Its rules are limited to the structure of the corresponding decision tree[2]. It seems like that ID algorithms describe rules intuitionally, but their accuracy is not satisfying.

It is quite obvious that the rule numbers of ID-family algorithms are much greater than that rough sets. Its accuracy is relatively low, see Table 2.

Remark. We learned rules on Monk-1 using three groups of algorithms. The knowledge reduction of rough sets is very satisfying. Its rule numbers is fewer and accuracy is high. We can say we gained all the knowledge comprised in Monk-1.

Table 3. Results by AQ-group to Monk-2

	AQ17-HCL	AQ17-FCLS	AQ15-GA	AQR
Accuracy	93.1%	92.6%	86.8%	79.7%

Table 4. Results by ID-family to Monk-2

	nodes	leaves	accuracy
ID3	66	110	67.9%
ID3 no windowing	64	110	69.1%
ID5R	64	99	69.2%
IDL	170	107	66.2%

3.2 Monk-2 Problem

In Monk-2, there are 169 training examples, 64 positive and 105 negative. Again, there is no noise. Its target concept is: “exactly two of (a1=1), (a2=1), (a3=1), (a4=1), (a5=1) and (a6=1) are valid”, which is very complex.

After the attribute reduction, the core is {a1, a2, a3, a4, a5, a6}, and the reduct is just the core. Then reduce the superfluous attribute values. The disposal process is similar to that of Monk-1 and 5 rules are produced. The knowledge we obtained is so complicated and different from the target concept in a certain extent. The reason maybe be that too many negative examples exist in Monk-2. The accuracy is only 75%. Thus it can be seen that the reduction algorithms on the basis of rough sets need not special but general examples.

Although the new attributes in AQ17-HCI is intricate and AQ17-FCLS summarizes 18 complicated rules, anyway, the accuracies of AQ-family on Monk-2 is high, showed as Table 3:

Results by ID-family are showed in Table 4:

Obviously, the number of the rules is relatively great and the accuracy is quite low. The algorithm is inferior to others on this problem.

3.3 Monk-3 Problem

In monk-3, there are 122 training examples, 60 negative and 62 positive. The number of negative examples is less than that of positive of examples. There are 5% misclassifications, i.e. noise in the training set. Its target concept is “(a5=3 and a4=1) or (a5⟨⟩4 and a2⟨⟩3)” which can be decomposed to 7 rules.

According to rough sets, the reduct is {a1, a2, a4, a5}, which has more attributes than target concepts by one. The noise led to this.

We get 23 rules in all, in which the values of a1 are well distributed, so we can draw the rule as a5=3 & a4=1 or (a5⟨⟩4 & a2⟨⟩3) or (a5=4 & a4=1) by neglecting a1, which is similar to the target concept. Rough sets is adaptive and rectifiable to some degree. But there is a intersection between the first two rules. We can't tell the class of the examples belonging to the intersection. For

Table 5. Results by AQ-group to Monk-3

	AQ17-HCI	AQ17-FCLS	AQ15-GA	AQR
Accuracy	100%	97.2%	100%	87.0%

Table 6. Results by ID-family to Monk-3

	nodes	leaves	accuracy
ID3	13	29	94.4%
ID3 no windowing	14	31	95.6%
ID5R	14	28	95.2%

example, we don't know the classes of the examples satisfying (a2=3 & a4=1 & a5=3). The reason is the data set of Monk-3 is inconsistent and with noise. Inconsistent data maybe lead to inconsistent rules.

In the test set, there are 12 examples we can't tell the classes and 4 examples we misclassify. The accuracy is 96.4%, which is acceptable.

For comparison let us list the results of AQ-family and ID-family by Table 5 and Table 6.

On Monk-3, the gaps among the accuracies of the three groups of algorithms are not very large, but the rules of rough sets are very concise.

4 Conclusions

In this paper, the inherent connections among the reduction theory of rough sets and several typical machine learning algorithms are narrated in detail; some experiments and comparisons are made on Monk's problems. Monk's problems are specially created for testing the quality of learning methods. So the conclusions on them are believable and valuable.

In Rough Sets, reducing superfluous attributes values can revise the results of the former, for instance, the disposal to a1 in Monk-3. This is another important feature of rough sets.

AQ-family and ID-family are typical algorithms in traditional machine learning. The adaptive AQ-family can be used to different data sets and have good quality, but their rules are complicated and not accessible. ID-family are simple and easily available, but their results are inferior to AQ-family and rough sets.

Acknowledgements. This research has been supported by grant No. 60175016 from the National Natural Science Foundation of China and grant No. 2001001 from the Youth Natural Science Foundation of Shanxi.

References

[1] Z. Pawlak. Rough Sets, Theoretical Aspects of Reasoning about Data. Warsaw, Poland, 1990.

[2] The MONK's Problems, A Performance Comparison of Different Learning Algorithms. 1991.

[3] Cestnik, B., Bratho. I. On Estimating Probabilities in Tree Pruning. Proc. of EWSL 91, Porto, Portugal, March 6–8, 1991.

[4] Quinlan, J.R. Learning Logical Definitions from Relations. Machine Learning 5(3), 239–266.

[5] J. Wang. Contributions on Rough Set Theory to Inductive Machine Learning. Computer Science 2001 285: 5–7.

[6] D.Q. Miao, G.R. Hu. A heuristic algorithm of knowledge reduction. Computer Research and Development, 366: 681–684.

[7] D.Q. Miao, J. Wang. An Information Representation of Concepts and Operations in Rough Sets. Journal of Software, 1999, 102: 113–116.

[8] Y. Yao, T.Y. Lin. A Review of Rough Set Methods. Rough Set and Data Mining. Kluwer Academic Publishers, 1997, 47–71.

[9] Polkowski L., Skowron A. Rough Sets: Perspectives, Rough Sets in Knowledge Discovering. In: Polkowski L, Skowron A, eds, Physica-Verlag, 1998, 1–27.

[10] D.Q. Miao, Lishan Hou. A Heuristic Algorithm foy Reduction of Knowledge Based on Discernibility Matrix. International Conference on Intelligent Information Technology, 2002, Beijing, 276–279.