Kovalerchuk, B., Vityaev E., Comparison of relational methods and attribute-based methods for data mining in intelligent systems**,** *1999 IEEE Int. Symposium on Intelligent Control/Intelligent Systems*, Cambridge, Mass, 1999. pp. 162-166. ***Working version***

Comparison of relational methods, attribute-based methods and hybrid methods.

Boris Kovalerchuk
Dept. of Computer Science, Central Washington University,
Ellensburg, WA, 98926-7520, USA
Borisk@cwu.edu

Evgenii Vityaev
Institute of Mathematics, Russian Academy of Science,
Novosibirsk, 630090, Russia
Vityaev@math.nsc.ru

Abstract

Most of the data mining methods in real-world intelligent systems are attribute-based machine learning methods such as neural networks, nearest neighbors and decision trees. They are relatively simple,- efficient, and can handle noisy data. However, these methods have two strong limitations: (1) the background knowledge can be expressed in rather limited form and (2) the lack of **relations** other than "object-attribute" makes the concept description language inappropriate for some applications.

Relational and hybrid data mining methods based on first-order logic are compared with Neural Networks and other benchmark methods on different data sets. These computational experiments show several advantages of relational and hybrid methods.

1. Problem definition and objectives

Relational Data Mining (RDM) combines inductive logic programming (ILP) with probabilistic inference. The combination benefits from noise robust probabilistic inference and highly expressive and understandable first-order logic rules employed in ILP.

Data mining has two major sources to infer rules: database and machine learning technologies. The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience. In recent years many successful machine learning applications have been developed, ranging from data-mining programs that learn to detect fraudulent credit card transactions, to information-filtering systems that learn users' reading preferences, to autonomous vehicles that learn to drive on public [6,8]

Currently statistical and Artificial Neural Network methods dominate in design of intelligent systems and data mining. Alternative **relational (symbolic) machine learning** methods had shown their effectiveness in robotics (navigation, 3-dimensional scene analysis) and drug design (selection of the most promising components for drug design). Traditionally symbolic methods are used in the areas with a lot of non-numeric (symbolic) knowledge. In robot navigation this is relative location of obstacles (on the right, on the left and so on). We discuss the key algorithms and theory that form the core of symbolic machine learning methods for applications with **dominating numerical data**.

Relational Data Mining (RDM) technology is a data modeling algorithm that **does not assume the functional form of the relationship being modeled a priori**. It can automatically consider a large number of inputs (e.g., time series characterization parameters) and learn how to combine these to produce estimates for future values of a specific output variable. Most of the data mining methods are attribute-based machine learning methods such as neural networks, nearest neighbors and decision trees. They are relatively simple, efficient, and can handle noisy data. However, these methods have two strong limitations: (1) the background knowledge can be expressed in rather limited form and (2) the lack of **relations** makes the concept description language inappropriate for some domains [1]. The purpose of a new area of machine learning called ***Inductive Logic Programming*** **(ILP)** is to overcome these limitations. Logic programming provided the solid theoretical basis for ILP. On the other hand at

present existing ILP systems are relatively inefficient and have rather limited facilities for **handling numerical data** [1]. We developed a hybrid ILP and probabilistic technique that handles numerical data efficiently [3,9].

One of the main advantages of ILP over attribute-based learning is ILP's generality of representation for **background knowledge**. This enables the user to provide, in a more natural way, domain-specific background knowledge to be used in learning. The use of background knowledge enables the user both to develop a suitable problem representation and to introduce problem-specific constraints into the learning process. By contrast, attribute-based learners can typically accept background knowledge in rather limited form only [1].

2. Comparison of problem requirements and method capabilities

Dhar and Stein [2] introduced a unified vocabulary for matching computational intelligence problems and methods. A problem is described using a set of requirements (**problem ID profile**). A method is described using its capabilities in the same terms. In [2] this vocabulary was applied for describing and comparing several data mining methods. Neural Networks (NN) are the most common methods in data mining. There are three shortages of NN for forecasting related to: (1) **explainability,** (2) use of **logical relations** and (2) **tolerance for sparse data**. Table 1 presents wider comparison of different data mining methods including Neural Networks.

3. Relational methods

A machine learning type of method, called Machine Methods for Discovering Regularities (MMDR) is applied for forecasting time series. The method expresses patterns in first order logic and assigns probabilities to rules generated by composing patterns. Currently the majority of learning systems for applications concentrate on neural networks, genetic algorithms, and related techniques. In practice, learning systems based on first-order representations have been successfully applied to many problems in chemistry, physics, medicine and other fields [1,6].

As any technique based on first order logic, MMDR allows one to get **human-readable forecasting rules** [1,6,8], i.e. **understandable** in ordinary language in addition to the forecast. A field expert can evaluate the performance of the forecast as well as a forecasting rule.

Also, as any technique based on probabilistic estimates, this technique delivers rules tested on their **statistical significance.** Statistically significant rules have advantage in comparison with rules tested only for their performance on training and test data [6, ch. 5]. Training and testing data can be too limited and/or not representative. If rules rely only on them then there are more chances that these rules will not deliver a right forecast on other data.

What is the motivation to use suggested MMDR method in particular? MMDR uses hypothesis/rule generation and selection process, based on fundamental representative measurement theory [5]. The original challenge for MMDR was the simulation of discovering **scientific laws** from empirical data in chemistry and physics. There is a well-known difference between "black box" models and fundamental models (laws) in modern physics. The last ones have much longer life, wider scope and a solid background.

In this paper we study several types of hypotheses/rules presented in first-order logic. They are simple relational assertions with variables. Mitchell [6] noted the importance that relational assertions "can be **conveniently expressed** using first-order representations, while they are **very difficult** to describe using propositional representations" (pp.275, 283-284).

Many well-known rule learners such as AQ, CN2 are propositional [6,7]. Note that decision tree methods represent a particular type of propositional representation [6, p.275]. Therefore decision tree methods as ID3 and its successor C4.5 fit better to tasks without relational assertions. Mitchell argues and gives examples that propositional representations offer no general way to describe the essential ***relations*** among the values of the attributes [6, pp. 283-284]. Below we follow his example. In contrast with propositional rules, a program using **first-order representations** could learn the following general rule:

IF *Father(x,y)* & *Female(y)*, THEN *Daugher(x,y),*

where x and y are variables that can be bound to any

person. For the target concept $Daughter_{1,2}$ **propositional rule learner** such as CN2 or C4.5, the result would be a collection of very specific rules such as

IF ($Father_1$=*Bob*)&$Name_2$=*Bob*)&$Female_1$=*True*)
THEN $Daughter_{1,2}$=*True*.

Although it is correct, this rule is so specific that it will rarely, if ever, be useful in classifying future pairs of people [6, pp.283-284]. We show that the close problem exists for ARIMA and Neural Networks methods. First-order logic rules have an advantage in discovering relational assertions because they capture relations directly, e.g., *Father(x,y)* in the example above.

In addition, first order rules allow one to express naturally other more general hypotheses not only the relation between pairs of attributes [3]. These more general rules can be as for classification problems as for an interval forecast of **continuous variable**. Moreover these rules are able to catch Markov chain type of models used for time series forecast. We share Mitchell's opinion about the importance of algorithms designed to learn sets of first-order rules that contain variables. "This is significant because first-order rules are much more expressive than propositional rules" [4, p.274].

What is the difference of other machine learning methods dealing with first-order logic [4, 5] and MMDR? From our viewpoint the main accent in other first-order methods [4, 5] is on two computational complexity issues: how wide is the class of hypotheses tested by the particular machine learning algorithms and how to construct a learning algorithm to find deterministic rules. The emphasis of MMDR is on **probabilistic first-order rules and measurement issues**, i.e., how we can move from a real measurement to first-order logic representation. Note that recently Muggleton's team moved to the same probabilistic direction. This is a non-trivial task [3]. For example, how to represent temperature measurement in terms of first-order logic without losing the essence of the attribute (temperature in this case) and without inputting unnecessary conventional properties? For instance, Fahrenheit and Celsius zeros of temperature are our conventions in contrast with Kelvin scale where the zero is a real physical zero. There are no temperatures less than this zero. Therefore incorporating properties of the Fahrenheit zero into first-order rules may force us to discover/learn properties of this convention along with more significant scale invariant forecasting rules. Learning algorithms in the space with those kind of accidental properties may be very time consuming and may produce inappropriate rules.

It is well known that the general problem of rule generating and testing is NP-complete. Therefore the discussion above is closely related to the following questions. What determines the number of rules and when to stop generating rules? What is the justification for specifying particular expressions instead of any other expressions? Using the approach from [3] we select rules which are simplest and consistent with measurement scales for a particular task.

The algorithm stops generating new rules when they become too complex (i.e., statistically insignificant for the data) in spite of possible high accuracy on training data. The obvious other stop criterion is time limitation. Detailed discussion about a mechanism of initial rule selection from measurement theory [3] viewpoint is out of the scope of this paper. A special study may result in a catalogue of initial rules/hypotheses to be tested (learned) for particular applications. In this way any field analyst can choose rules to be tested without generating them. This paper delivers a preliminary list of rules for that catalogue.

The critical issue in applying data-driven forecasting systems is generalization. The "Discovery" system generalizes data through "lawlike" logical probabilistic rules. Discovered rules have similar statistical estimate and significance on training and test sets of studied time series. Theoretical advantages of MMDR generalization are presented in [6, 2]. We use mathematical formalisms of first order logic rules described in [5, 3].

4 Method for discovering regularities

Figure 1 describes the steps of MMDR. On the first step we select and/or generate a class first–order logic rules suitable for a particular task.

The next step is learning the particular first-order logic rules using available training data. Then we test

first-order logic rules on training and test data using Fisher statistical criterion. After that we select

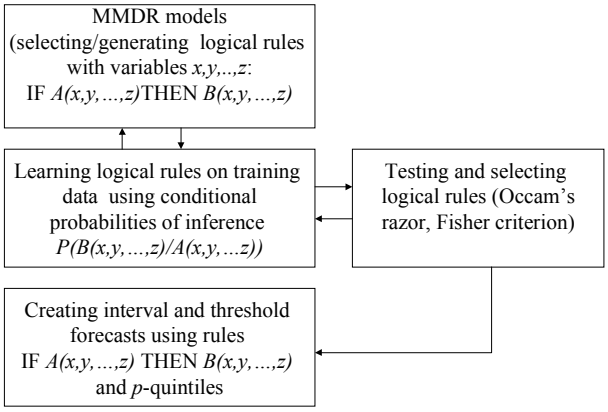

Figure 1. Flow diagram for MMDR: steps and technique applied

statistically significant rules and apply Occam's razor principle: prefer the simplest hypothesis (rules) that fits the data [4, p. 65]. Simultaneously we use the rules' performance on training and test data for their selection. We may iterate back and forth among these three steps several times to find the best rules. The last step is creating interval and threshold forecasts using selected first-order logic rules:

IF *A(x,y,...,z)* THEN *B(x,y,...,z)*.

Conceptually **law-like rules** came from philosophy of science. These rules attempt to mathematically capture the essential features of **scientific laws:**
(1) High level of generalization;
(2) Simplicity (Occam's razor); and,
(3) Refutability.

The first feature -- generalization -- means that any other regularity covering the same events would be less general, i.e., applicable only to the part of events covered by the law-like regularity. The second feature – simplicity--reflects the fact that a law-like rule is shorter than other rules. The law-like rule (R1) is more refutable than another rule (R2) if there are more testing examples which refute (R1) than (R2), but the examples fail to refute (R1).

Formally, we present an IF-THEN rule C as

$$A_1\& \ldots \& A_k \Rightarrow A_0,$$

where the IF part, $A_1\&...\&A_k$, consists of true/false logical statements $A_1,\ldots,A_k$,and the THEN part consists of a single logical statement A_0. Statements A_i are some given refutable statements or their negations, which are also refutable. Rule C allows us to generate sub-rules with a truncated IF part, e.g.

$$A_1\&A_2 \Rightarrow A_0\ ,\ A_1\&A_2\&A_3 \Rightarrow A_0$$

and so on. For rule C its conditional probability

$$Prob(C) = Prob(A_0/A_1\&...\&A_k)$$

is defined. Similarly conditional probabilities

$$Prob(A_0/\ A_{i1}\&...\&A_{ih})$$

are defined for sub-rules Ci of the form

$$A_{i1}\& \ldots \& A_{ih} \Rightarrow A_0.$$

We use conditional probability

$$Prob(C) = Prob(A_0/A_1\&...\&A_k)$$

for estimating forecasting power of the rule to predict A_0.

The **rule is** "**law-like" iff all of its sub-rules have less conditional probability than the rule, and statistical significance of that is established.** Each sub-rule C_i generalizes rule C, i.e., potentially C_i is true for larger set instances [19, chapter 10]. Another definition of "law-like" rules can be stated in terms of generalization.

The **rule is** "**law-like" if and only if it can not be generalized without producing a** statistically significant reduction in **its conditional probability.**

"Law-like" rules defined in this way hold all three mentioned above properties of scientific laws: generality, simplicity, and refutability..
The "Discovery" software searches all chains

$$C_1\ ,\ C_2,\ \ldots,\ C_{m-1},\ C_m$$

of nested "law-like" subrules, where C_1 is a subrule of rule C_2 , $C_1 = sub(C_2)$, C_2 is a subrule of rule C_3, $C_2 = sub(C_3)$ and finally C_{m-1} is a subrule of rule C_m, $C_{m-1} = sub(C_m)$. In addition, they satisfy an important property:

$$Prob(C_1) < Prob(C_2), \ldots, Prob(C_{m-1}) < Prob(C_m).$$

This property is the base for the following **theorem** [9]:

> All rules, which have a maximum value of conditional probability, can be found at the end of such chains.

This theorem basically means that the MMDR algorithm does not miss the best rules. The algorithm stops generating new rules when they become too complex (i.e., statistically insignificant for the data) even if the rules are highly accurate on training data. The Fisher statistical criterion is used in this algorithm for testing statistical significance. The obvious other stop criterion is time limitation.

Theoretical advantages of MMDR generalization are presented in [3,4,9]. This approach has some similarity with the hint approach [20]. We use mathematical formalisms of first-order logic rules described in [21-23]. Note that a class of general propositional and first-order logic rules, covered by MMDR is wider than a class of decision trees [19, pp. 274-275].

5. Performance

We considered a control task with two control actions values u=0 and u=1, set as follows:

$$\text{If } y_t \tilde{} > y_{t-1} \text{ then } u=1 \text{ else if } (y_t \tilde{} = y_{t-1} \text{then } u=0 \text{ else } u=-1),$$

where y_{t-1} is an actual value of the time series for t-1 moment and $y_t \tilde{}$ is the forecasted value of y for t moment. Forecasted values are generated by all four studied methods including Neural Network method and relational MMDR method.

The gain function G is computed after all actual values of y_t for all considered t became known. Gain function $G(u, y_t, y_{t-1})$ depends on u and actual values of output y_t, y_{t-1}:

$$G(u, y_t, y_{t-1}) = y_t - y_{t-1} \text{ if } u=1 \text{ and } -(y_t - y_{t-1}) \text{ if } u=0$$

The adaptive linear method in essence works according to the following fomula:

$$y_t \tilde{} = a y_{t-1} + b y_{t-2} + c.$$

If

"No active control" method ignores forecast y_t

and in all cases generates the same control signal "do nothingn"

If $(y_t \tilde{} > y_{t-1}$ or $y_t \tilde{} \le y_{t-1})$ then u=0 (do nothing).,

Table 1 shows performance of MMDR in conparison with three other methods: Adaptive linear, passive contol, and neural networks. In avarage human-readable and understadable regularities generated MMDR autperformed other methods including widely used neural networks.

Table 1. Simulated gain

Method	Gain (%)		
	Data set 1 (456 instances)	Data set 2 (456 instances)	Average (two data sets, 856 instances)
Adaptive Linear	21.9	18.28	20.09
MMDR	26.69	43.83	35.26
No active control	30.39	20.56	25.47
Neural Network	18.94	16.07	17.5

6. References

1) Bratko, I., Muggleton. S. Applications of inductive logic programming, *Communications of ACM,* vol.38, N. 11,1995, pp.65-70.

2) Dhar V., Stein R. *Intelligent decision support methods*. Prentice Hall, 1997

3) Kovalerchuk, B., Vityaev E. "Discovering Lawlike Regularities in Financial Time Series", *Journal of Computational Intelligence in Finance*, vol.6, No.3, 1998, pp.12-26,

4) Kovalerchuk, B., Triantaphyllou, E., Despande, A., and Vityaev, E. "Interactive Learning of Monotone Boolean Function". Information Sciences, Vol. 94, issue 1-4, 1996, pp. 87-118.

5) Krantz D.H., Luce R.D., and Suppes P., Tversky A. “Foundations of measurement”. vol.1-3, Acad. Press, NY, London, 1971, 1989, 1990

6) Mitchell T. *Machine Learning*, McGraw-Hill, NY, 1997

7) Quinlan J.R.*C4.5: Programs for Machine Learning.* San Mateo, CA; Morgan Kaufmann, 1993.

8) Russel S. and Norvig P. *Artificial Intelligence*. *A modern approach,* Prentice Hall, 1995

9) Vityaev E., Moskvitin A. Introduction to Discovery theory: Discovery system, *Computational Systems,* v.148, Novosibirsk, p.117-163, 1993 (in Russian).

Table 2. Comparison of capabilities of methods

	Dimension	NN	CBR	FL	DR	ST	DT	ILP	PILP
1	Accuracy	H	MH	H		MH	MH	H	H
2	Explainability	L	M	M	H	MH	MH	H	H
3	Response speed	H	MH	H	LM	MH	H	H	H
4	Scalability	M	H	M	M	MH	MH	M	M
5	Compactness	H	LM	H	L		M	M	M
6	Flexibility	H	H	H	M	LM	H	H	
7	Embeddability	H	M	M	L	H	MH	MH	MH
8	Tolerance for complexity	H	M	H	L	LM	M	LM	LM
9	Tolerance for noise in data	MH	M			LM	M	L	H
10	Tolerance for sparse data	L	M				L	L	H
11	Independence from experts	H	MH	M	L	H	M	M	H
12	Development speed	M		M	MH		M	M	M
13	Used computing resources	LM		L			M	M	M
14	Ease of use	M		M				M	M
15	Ease of use of logical relations	LM	L	M	H		L	H	H
16	Ease of use of numerical data	H	H	H	LM		H	LM	LM

The white part of table 3 is based on tables presented in [2]. We generated the rest part to show the importance of first-order logic methods.
Notation:
NN--neural networks; CBR--case-based reasoning;
FL—fuzzy logic; DR—deductive reasoning (expert systems)
ST—statistical methods; DT--decision trees
ILP—inductive logic programming
PILP—probabilistic ILP;
H—high; M—medium; L—low.