Yao, Y.Y., Zhao, Y., Wang, J., and Han, S., A model of machine learning based on user preference of attributes, Rough Sets and Current Trends in Computing, 5th International Conference, RSCTC 2006, Proceedings, LNAI 4259, pp. 587-596, 2006.

A Model of Machine Learning Based on User Preference of Attributes

Yiyu Yao[1], Yan Zhao[1], Jue Wang[2] and Suqing Han[2]

[1] Department of Computer Science, University of Regina,
Regina, Saskatchewan, Canada S4S 0A2
E-mail: {yyao, yanzhao}@cs.uregina.ca
[2] Laboratory of Complex Systems and Intelligence Science Institute of Automation,
Chinese Academy of Science, Beijing, China 100080
E-mail: {jue.wang, suqing.han}@mail.ia.ac.cn

Abstract. A formal model of machine learning by considering user preference of attributes is proposed in this paper. The model seamlessly combines internal information and external information. This model can be extended to user preference of attribute sets. By using the user preference of attribute sets, user preferred reducts can be constructed.

1 Introduction

A basic task of machine learning and data mining is to derive knowledge from data. The discovered knowledge in general should be concise, precise, general, easy to understand and practically useful. Typically, knowledge is expressed by using a certain formal language or a representation scheme. It is a crucial issue to select the most suitable features or properties of the objects in a dataset in the machine learning process. This attribute selection problem is studied under many different areas, such as data reduction, feature selection, rule generation, and so on [1, 4–6].

Many proposals have been made regarding the effectiveness of individual attributes, or subsets of attributes. They can be broadly divided into two classes, the approaches based on internal information and the approaches based on external information. Internal information and external information are so called and distinguished with respect to the dataset. Internal information based approaches typically depend on the syntactic or statistical information of the dataset. For example, an attribute weighting function is designed by using attributes' distribution information or prediction power. The most fit attribute is used firstly in the rule construction process. On the contrary, external information based approaches assign weights to attributes, or rank attributes based on external semantics or constraints. It is important to realize that these two classes are complementary to each other. Together, they may provide a realistic model for machine learning and data mining. That is, it is desirable that one can consider both syntactical and semantical information in a unified framework.

A review of existing research in machine learning observes that the major research efforts have been done on the internal information based approaches,

although the external information based approaches may be more meaningful and effective. This may stem from the fact that external information covers a very diverse range, is highly subjective, and usually is not well-defined. Consequently, it may be difficult to build a well-accepted model. In this paper, we only consider very simple cases of external information based on our intuitions. We provide a formal model of machine learning by considering user preference of attributes. The model seamlessly combines internal information and external information.

The rest of the paper is organized as follows. Section 2 discusses the user preference of attributes. Section 3 extends the user preference of attributes to attributes sets. Both qualitative and quantitative representations of these two models are discussed. Section 4 illustrates the usefulness of the proposed model by applying it to reduct construction. The conclusion is made in Section 5.

2 User Preference of Attributes

In many machine learning algorithms, it is implicitly assumed that all attributes are of the same importance from a user's point of view. Consequently, attributes are selected based solely on their characteristics revealed in an information system. This results in a model, which is simple and easy to analyze. At the same time, without considering the semantic information of attributes, the model is perhaps unrealistic. A more applicable model can be built by considering attributes with non-equal importance. This type of external information is normally provided by users in addition to the information system, and is referred to as user judgement or user preference.

User judgement can be expressed in various forms. Quantitative judgement involves the assignment of different weights to different attributes. Qualitative judgement is expressed as an ordering of attributes. In many situations, user judgement is determined by semantic considerations. For example, it may be interpreted in terms of notions that are more intuitive, such as the cost of testing, the easiness of understanding, or the actionability of an attribute. It is virtually impossible to list all practical interpretations of user judgement. In addition, the meaning of a user judgement becomes clear only in a particular context of application. To simplify our discussion, we treat user judgement as a primitive notion. In other words, we only investigate the desirable properties of a user judgement, as well as how to incorporate it into a machine learning process.

A practical issue is how to acquire user preference. One may argue that a user might not be able to precisely and completely express preference on the entire attribute set. For clarity, we simply assume that a user *can* provide such information. This enables us to investigate the real issues without the interference of unnecessary constraints. Practical constraints, although very important, can always be resolved, at least partially, with further understanding of the problem, or the development of additional methods.

2.1 Quantitative user judgement

A simple and straightforward way to represent user judgement of attributes is to assign them with numerical weights. Formally, it can be described by a mapping:

$$w : At \longrightarrow \Re, \tag{1}$$

where At is a finite non-empty set of attributes, and $\Re$ is the set of real numbers. For an attribute $a \in At$, $w(a)$ is the weight of a. The numerical weight $w(a)$ may be interpreted as the degree of importance of a, the cost of testing a in a rule, or times of occurrence of a in a set (which is also called the frequency of a). This naturally induces an ordering of attributes. For example, if the weights are interpreted as costs, a machine learning algorithm should apply, if possible, attributes with lower costs first. Furthermore, one may also apply arithmetic operations on the weights.

The use of numerical weights for attribute importance has been extensively considered in machine learning. In many learning algorithms, a numerical function is used to compute weights of individual attributes based on their distribution characteristics. According to the computed weights, attributes are selected. For example, entropy-theoretic measures have been studied and used for attribute selection [7].

2.2 Qualitative user judgement

A difficulty with the quantitative method is the acquisition of the precise and accurate weights of all attributes. On the other hand, a qualitative method only relies on pairwise comparisons of attributes. For any two attributes, we assume that a user is able to state whether one is more important than, or more preferred to, the other. This qualitative user judgement can be formally defined by a binary relation $\succ$ on At. For any two $a, b \in At$:

$$a \succ b \Longleftrightarrow \text{ the user prefers } a \text{ to } b. \tag{2}$$

The relation $\succ$ is called a preference relation. If $a \succ b$ holds, we say that the user strictly prefers a to b. In contrast to the quantitative representation, the preference does not say anything regarding the degree of preference, namely, how much a is preferred to b.

In the absence of preference, i.e., if both $\neg(a \succ b)$ and $\neg(b \succ a)$ hold, we say that a and b are indifferent. An indifference relation $\sim$ on At is defined as:

$$a \sim b \Longleftrightarrow \neg(a \succ b) \wedge \neg(b \succ a). \tag{3}$$

The indifference of attributes may be interpreted in several ways. A user may consider the two attributes are of the same importance. The indifference may also occur when the comparison of two attributes are not meaningful, as they are incompatible. When both a and b are unimportant, it may not make too much sense to compare them. The indifference represents such an absence of preference. In fact, in many practical situations, one is only interested in expressing

preference on a subset of crucial attributes, and considers all unimportant attributes to be the same.

Based on the strict preference and indifference, one can define a preference-indifference relation $\succeq$ on At:

$$a \succeq b \Longleftrightarrow a \succ b \vee a \sim b. \tag{4}$$

If $a \succeq b$ holds, we say that b is not preferred to a, or a is at least as good as b. The strict preference can be re-expressed as $a \succ b \Longleftrightarrow a \succeq b \wedge \neg(b \succeq a)$.

A user preference relation must satisfy certain axioms in order to represent our intuitive understanding of preference. The following two axioms seem to be reasonable for $\succ$. For any $a, b, c \in At$:

(1). $a \succ b \Longrightarrow \neg(b \succ a)$ (asymmetry);
(2). $(\neg(a \succ b) \wedge \neg(b \succ c)) \Longrightarrow \neg(a \succ c)$ (negative transitivity).

The asymmetry axiom states that a user cannot prefer a to b, and at the same time prefer b to a. The negative transitivity axiom states that if a user does not prefer a to b, nor b to c, then the user should not prefer a to c. If a preference relation $\succ$ on At is asymmetric and negatively transitive, it is called a *weak order*.

A weak order imposes a special structure on the set of attributes. Additional properties of a weak order are summarized in the following lemma [2].

Lemma 1. *Suppose a preference relation $\succ$ on a finite set of attributes At is a weak order. Then,*

- *Exactly one of $a \succ b$, $b \succ a$ and $a \sim b$ relations holds for any two $a, b \in At$;*
- *The indifference relation $\sim$ is an equivalence relation, which induces a partition $At/\sim$ of At;*
- *The relation $\succ'$ on the partition $At/\sim$, defined by $[a]_\sim \succ' [b]_\sim \Longleftrightarrow a \succ b$, is a linear order, where $[a]_\sim$ is the equivalence class containing a.*

A linear order is a weak order in which any two distinct elements are comparable. This lemma implies that if $\succ$ is a weak order, the indifference relation $\sim$ divides the set of attributes into disjoint subsets. Furthermore, for any two distinct equivalence classes $[a]_\sim$ and $[b]_\sim$ of $At/\sim$, either $[a]_\sim \succ' [b]_\sim$ or $[b]_\sim \succ' [a]_\sim$ holds. In other words, it is possible to arrange the attributes into several levels so that attributes in a higher level are preferred to attributes in a lower level, and attributes in the same level are indifferent.

When each equivalence class contains exactly one attribute, the preference relation $\succ$ on At is in fact a linear order itself. The ordering has been considered by some authors [3, 9]. In general, if we do not care how to order attributes in an equivalence class, we can extend a weak order into a linear order such that a is ranked ahead of b if and only if $a \succeq b$. For a weak order, its linear extension may not be unique [2].

Example 1. The main notions of qualitative user preference can be illustrated by a simple example. Suppose a user preference relation $\succ$ is qualitatively specified on a set of attributes $At = \{a, b, c, d\}$ by the following weak order:

$$c \succ a, c \succ b, d \succ a, d \succ b, d \succ c.$$

This relation $\succ$ satisfies the asymmetry and negative transitivity conditions. Because of the absence of preference relation between attribute a and b, we say $a \sim b$. Thus, three equivalence classes $\{d\}, \{c\}, \{a, b\}$ can be found. They can also be written as $[d]_\sim, [c]_\sim, [a]_\sim$ (or $[b]_\sim$), respectively. In turn, they can be arranged as three levels in a linear order:

$$\{d\} \succ' \{c\} \succ' \{a, b\}.$$

If one does not care the order of attributes in an equivalence class, we can extend the weak order of attributes into a linear order of attributes. The given weak order can be extended to two linear orders on At:

$$d \succeq c \succeq b \succeq a,$$
$$d \succeq c \succeq a \succeq b.$$

2.3 Connections between quantitative and qualitative judgements

A quantitative judgement can be easily translated into a qualitative judgement. Given the weights of attributes, we can uniquely determine a preference relation. Suppose $w(a)$ and $w(b)$ represent the importance of $a, b \in At$, a preference relation is defined by:

$$a \succ b \Longleftrightarrow w(a) > w(b). \tag{5}$$

When $w(a)$ and $w(b)$ are the costs of testing attributes $a, b \in At$ in a rule, the following preference relation should be used instead,

$$a \succ b \Longleftrightarrow w(a) < w(b). \tag{6}$$

In general, two attributes may have the same weights. Therefore, the induced preference relation is indeed a weak order.

The translation to a preference relation only preserves the ordering of attributes implied by the relative weights. The additional information given by the absolute weight values is lost.

In the reverse process, a user preference relation can be represented in terms of the weights of attributes. A rational user's judgement must allow numerical measurement.

The following theorem states that a weak order is both necessary and sufficient for a numerical measurement [2]:

Theorem 1. *Suppose $\succ$ is a preference relation on a finite non-empty set At of attributes. There exists a real-valued function $u : At \to \Re$ satisfying the condition:*

$$a \succ b \Longleftrightarrow u(a) > u(b), a, b \in At. \tag{7}$$

if and only if $\succ$ is a weak order. Moreover, u is uniquely defined up to a strictly monotonic increasing transformation.

The function u is referred to as an order-preserving utility function. It provides a quantitative representation of a user preference. That is, the numbers of $u(a), u(b), \ldots$ as ordered by $>$ reflect the order of $a, b, \ldots$ under the preference relation $\succ$.

The utility function also trustfully represents the indifference relation, i.e.,

$$a \sim b \Longleftrightarrow u(a) = u(b), a, b \in At. \tag{8}$$

According to Theorem 1, for a given preference relation, there exist many utility functions. For a utility function, we can only obtain one preference relation. Under the ordinal scale, it is only meaningful to examine the order induced by a utility function. Although numerical values are used, it is not necessarily meaningful to apply them to arithmetic operations.

Example 2. We can easily observe the connections between a preference relation and a set of weights by the running example. Suppose we can define user preference quantitatively on the set $At = \{a, b, c, d\}$. For example, we can define a utility function u_1 as information entropy, therefore, $u_1(a) = 0, u_1(b) = 0, u_1(c) = 0.8, u_1(d) = 1$. We can also define another utility function u_2 as the cost of testing, therefore, $u_2(a) = 2^{10}, u_2(b) = 2^{10}, u_2(c) = 4, u_2(d) = 0$. The two utility functions define two opposite orders for any pair of attributes. They also use different measurement scales. While the utility function u_1 is used, a preference relation is defined by Equation 5; while the utility function u_2 is used, a preference relation is naturally defined by Equation 6. The example identifies that a user preference relations can be induced by more than one utility functions. A utility function can decide a rational user preference.

One can impose addition axioms on user preference. It is then possible to derive quantitative measurements using other scales. Different scales allow more operations [2].

3 User Preference of Attribute Sets

Conceptually, rule learning in an information system can be viewed as two tasks, the selection of a subset of attributes, and the construction of rules using such attributes. The two tasks can in fact be integrated in one algorithm without a clear separation. Ideally, the subset should contain more preferred attributes and avoid including less preferred attributes. In this case, users should be able to express the preference over subsets of attributes. This requires a user preference relation on the power set 2^{At}. In this section, we present the way to derive a preference relation $\succ$ on 2^{At} based on a preference relation $\succ$ on At.

3.1 Basic properties

For simplicity, we use the same symbol to denote the preference relation on At and the preference relation on 2^{At}. Obviously, the relation $\succ$ on 2^{At} needs to satisfy certain conditions.

By definition, $\succ$ on 2^{At} must be an extension of $\succ$ on At. That is,

$$\begin{aligned} &\text{(E1).} \quad \{a\} \succ \{b\} \Longleftrightarrow a \succ b; \\ &\text{(E2).} \quad \{a\} \sim \{b\} \Longleftrightarrow a \sim b; \\ &\text{(E3).} \quad \{a\} \succeq \{b\} \Longleftrightarrow a \succeq b. \end{aligned}$$

Suppose $\succ$ on At is a weak order. For a subset of attributes $A \subseteq At$, the cardinality $|A| = k$, we can arrange the attributes of A into a linear order in the form of $a_1 \succeq a_2 \succeq \ldots \succeq a_k$. According to Theorem 1, this requires the following axiom:

$$\text{(T).} \quad \succ \text{ on } 2^{At} \text{ is a weak order.}$$

The previous axioms may be considered as the basic properties of $\succ$ on 2^{At}. In addition, $\succ$ on 2^{At} must allow quantitative measurements. One may impose on additional conditions, depending on particular applications.

3.2 Qualitative extensions

For a set of attributes, we can arrange them in a linear order based on the preference-indifference relation $\succeq$. This enables us to derive a possible ordering of subsets by consecutively examining attributes one by one. Based on the directions in which attributes are examined, we define two lexical orders. In the left-to-right lexical order, we compare two sets of attributes from left to right, in order to determine which set of attributes is preferred. In the right-to-left lexical order, we compare attributes in the reverse order.

Definition 1. Left-to-right lexical order: *Given two attribute sets* $A : a_1 \succeq a_2 \succeq \ldots \succeq a_m$ *and* $B : b_1 \succeq b_2 \succeq \ldots \succeq b_n$, *let* $t = \min\{m, n\}$. *We say that* A *precedes* B *in the left-to-right lexical order, written* $A \succ B$, *if and only if*

(a) *there exists a* $1 \leq i \leq t$ *such that* $a_j \sim b_j$ *for* $1 \leq j < i$ *and* $a_i \succ b_i$, *or*
(b) $a_i \sim b_i$ *for* $1 \leq i \leq t$ *and* $m < n$.

Definition 2. Right-to-left lexical order: *Given two attribute sets* $A : a_1 \succeq a_2 \succeq \ldots \succeq a_m$ *and* $B : b_1 \succeq b_2 \succeq \ldots \succeq b_n$, *let* $t = \min\{m, n\}$. *We say that* A *precedes* B *in the right-to-left lexical order, written* $A \succ B$, *if and only if*

(a) *there exists a* $0 \leq i < t$ *such that* $a_{m-j} \sim b_{n-j}$ *for* $0 \leq j < i$ *and* $a_{m-i} \succ b_{n-i}$, *or*
(b) $a_{m-i} \sim b_{n-i}$ *for* $0 \leq i < t$ *and* $m < n$.

These two lexical orders represent two extreme views and define two different criteria for selecting the winner of attribute sets. Roughly speaking, the meaning of these two can be interpreted as follows. The left-to-right method focuses on the preferred attributes of the two sets. That is, the winner of all attribute sets is determined by comparing the strongest attributes of individual sets. By the left-to-right lexical order, an attribute set A is preferred to another attribute

set B if and only if (1) the most preferred attribute of A is preferred to the most preferred attribute of B, or, (2) A is a proper subset consisting of the most preferred attributes of B.

On the other hand, the right-to-left method emphasizes the less preferred attributes of the two sets. The winner of all subsets of attributes is determined by comparing the weakest attributes of individual sets. By the right-to-left lexical order, an attribute set A is preferred to another attribute set B if and only if (1) the least preferred attribute of A is preferred to the least preferred attribute of B, or, (2) A is a proper subset consisting of the least preferred attributes of B.

The left-to-right lexical order encourages an optimistic comparison, and the right-to-left lexical order promotes a pessimistic comparison.

Example 3. The running example can be used to illustrate the differences between two lexical orders. Recall that attributes in Example 1 can be arranged as $\{d\} \succ' \{c\} \succ' \{a, b\}$. For two attribute subsets $A : d \succeq c \succeq a$ and $B : d \succeq a$, since $d \sim d$ and $c \succ a$, then A is the winner according to the left-to-right lexical order. At the same time, since $a \sim a$ and $d \succ c$, thus B is the winner according to the right-to-left lexical order.

For two attribute subsets $C : d \succeq c \succeq a$ and $D : c \succeq b$, since $d \succ c$, then C is the winner according to the left-to-right lexical order. On the other hand, since $a \sim b, c \sim c$ and $|D| < |C|$, then D is the winner according to the right-to-left lexical order.

It is essential to note that both lexical orders satisfy Axioms (E1,2,3) and (T), and should be considered as examples of potential extensions of the preference order from At to 2^{At}. They may provide different preference orders based on their criteria, as we just showed in the example. It may be difficult to argue which one is better based solely on theoretical basis. In real applications, we might also need to consider other extensions.

3.3 Quantitative extensions

When user preference is given as weights of attributes, one can first define a preference and then use the previously discussed qualitative methods. The numerical weights also offer addition methods. We can extend the weighting function w on At to a weighting function on 2^{At}. For simplicity, we use the same symbol to denote these two functions. Similarly, the extensions are not unique. For example, for $A \subseteq At$, we consider the following possible extensions:

$$\begin{aligned}
&\text{Additive extension: } w(A) = \textstyle\sum_{a \in A} w(a),\\
&\text{Average extension: } w(A) = \frac{\sum_{a \in A} w(a)}{|A|},\\
&\text{Maximal extension: } w(A) = \max_{a \in A} w(a),\\
&\text{Minimal extension: } w(A) = \min_{a \in A} w(a).
\end{aligned}$$

The extensions are not true or false. They are simply useful or not useful for some purposes. One can interpret the meaningful extensions based on the

physical meaning of the weighting function on At. It is important to note that only some extensions are meaningful in a particular application.

The values of an extension naturally define an order. For example, if $w(a)$ is a cost measurement function, the above extensions are interpreted as the total cost, average cost, maximal cost and minimal cost, respectively. An attribute set with the minimum cost is normally in favour. If $w(a)$ is an information measurement function, $w(A)$ is the joint information of all attributes in the set. Normally, an attribute set with the maximal information gain is in favour. Based on the computed weights, we can order subsets of attributes in a similar way as given by Equations 5 and 6.

4 User preference on reducts

The usefulness of the proposed model can be illustrated by reduct construction. A reduct is the minimal subset of attributes that preserves the discernible information of an information table. Conceptually, internal information determines a set of reducts, and user preference determines an ordering of reducts. By involving user preference in the reduct construction process, we can observe two directions. First is to choose the user preferred reducts while all reducts are available. Second is to construct a user preferred reduct directly. It is obvious that the second approach is more efficient.

Regarding the two lexical orders, we can define an RLR algorithm for computing the winner reduct of the right-to-left lexical order, and an LRR algorithm for computing the winner reduct of the left-to-right lexical order. We define that an attribute set $R' \subseteq At$ is called a super-reduct of a reduct R if $R' \supseteq R$; and an attribute set $R' \subset At$ is called a partial reduct of a reduct R if $R' \subset R$. Given a reduct, there exist many super-reducts and many partial reducts.

An RLR algorithm uses a deletion strategy, that removes the less preferred attributes one by one from the super-reduct, until a reduct is obtained. An RLR algorithm can start from the largest super-reduct At, or a computed super-reduct $A \subseteq At$. An LRR algorithm uses an addition strategy, that adds the most preferred attributes one by one to an empty set, until a reduct is obtained. It is important to note that as long as an attribute is added, it is hard to remove it. Therefore, the addition strategy should be carried out with caution. The general RLR and LRR algorithms are briefly illustrated below.

A general RLR algorithm:
Input: An information table S with At in a linear preference order.
Output: The winner reduct of the right-to-left lexical order.
(1) $R = At$, $CD = At$.
(2) While $CD \neq \emptyset$:
  (2.1) Consider all attributes in CD from right to left, let $CD = CD - \{a\}$;
  (2.2) If $R - \{a\}$ is a super-reduct, let $R = R - \{a\}$.
(3) Output R.

A general LRR algorithm:
Input: An information table S with At in a linear preference order.
Output: The winner reduct of the left-to-right lexical order.
(1) $R = \emptyset,\ CA = At$.
(2) While R is not a reduct and $CA \neq \emptyset$:
(2.1) Consider all attributes in CA from left to right;
(2.2) If $R \cup \{a\}$ is a partial reduct, let $R = R \cup \{a\}$, and $CA = CA - \{a\}$.
(3) Output R.

It is important to note that the deletion strategy and the addition strategy correspond to the RLR algorithm and the LRR algorithm, respectively. The cross effect is not easy to implement, if it is not impossible. The detailed implementation and discussion of these two strategies are presented in our another recent paper [8], and will be addressed more carefully in our following research.

5 Conclusion

We propose a model for machine learning based on user preference of attributes. This model can be extended to user preference of attribute sets. Both qualitative and quantitative representations of user preference on attributes and attribute sets are elaborately explored. With respect to user preference of attribute sets, various of applications, such as the computation of the most preferred reducts can be intensively studied.

References

1. Blum, A.L. and Langley, P., Selection of relevant features and examples in machine learning, *Artificial Intelligence*, 97, 245-271, 1997.
2. Fishburn, P.C., *Utility Theory for Decision-Making*, John Wiley & Sons, New York, 1970.
3. Han, S.Q. and Wang, J., Reduct and attribute order, *Journal of Computer Science and Technology archive*, 19(4), 429-449, 2004.
4. Jain, A., Duin, P. and Mao, J., Statistical pattern recognition: A review, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(1), 4-37, 2000.
5. Kohavi, R. and John, G., Wrappers for feature subset selection, *Artificial Intelligence*, 97(1-2), 273-324, 1997.
6. Swiniarski, R.W. and Skowron, A., Rough set methods in feature selection and recognition, *Pattern Recognition Letters*, 24(6), 833-849, 2003.
7. Yao, Y.Y., Chen, Y.H. and Yang, X.D. A measurement-theoretic foundation for rule interestingness evaluation, *Proceedings of Workshop on Foundations and New Directions in Data Mining in the Third IEEE International Conference on Data Mining (ICDM 2003)*, 221-227, 2003.
8. Yao, Y.Y., Zhao, Y. and Wang, J., On reduct construction algorithms, *Proceedings of the First International Conference on Rough Sets and Knowledge Technology*, 297-304, 2006.
9. Ziarko, W., Rough set approaches for discovering rules and attribute dependencies, in: Klösgen, W. and Żytkow, J.M. (eds.), *Handbook of Data Mining and Knowledge Discovery*, Oxford, 328-339, 2002.