Action-Rules: How to Increase Profit of a Company

Zbigniew W. Ras[1,2] and Alicja Wieczorkowska[1,3]

[1] UNC-Charlotte, Computer Science Dept., Charlotte, NC 28223, USA
[2] Polish Academy of Sciences, Inst. of Comp. Science, Warsaw, Poland
[3] Polish-Japanese Institute of Information Tech., 02-018 Warsaw, Poland
ras@uncc.edu or awieczor@uncc.edu

Abstract. Decision tables classifying customers into groups of different profitability are used for mining rules classifying customers. Attributes are divided into two groups: stable and flexible. By stable attributes we mean attributes which values can not be changed by a bank (age, marital status, number of children are the examples). On the other hand attributes (like percentage rate or loan approval to buy a house in certain area) which values can be changed or influenced by a bank are called flexible. Rules are extracted from a decision table given preference to flexible attributes. This new class of rules forms a special repository of rules from which new rules called action-rules are constructed. They show what actions should be taken to improve the profitability of customers.

1 Introduction

In the banking industry, the most widespread use of data mining is in the area of customer and market modeling, risk optimization and fraud detection. In financial services, data mining is used in performing so called trend analysis, in analyzing profitability, helping in marketing campaigns, and in evaluating risk. Data mining helps in predictive modeling of markets and customers, in support of marketing. It can identify the traits of profitable customers and reveal the "hidden" traits. It helps to search for the sites that are most convenient for customers as well as trends in customer usage of products and services. Examples of specific questions include:

1. What are micromarkets for specific products?
2. What is the likelihood that a given customer will accept an offer?
3. What actions improve the profitability of customers?
4. What customers switch to other services?

Even if it is very tempting to use limited tools and seek quick answers to very specific questions, it is important to keep a broad perspective on knowledge. For instance, while the direct data make a particular offer look profitable, a more thorough analysis may reveal cannibalization of other offers and the overall decrease in profit.

Specific questions about profitability must be answered from a broader perspective of customer and market understanding. For instance, customer loyalty can be often as important as profitability. In addition to short term profitability the decision makers must also keep eye on the lifetime value of a customer. Also, a broad and detailed

D.A. Zighed, J. Komorowski, and J. Zytkow (Eds.): PKDD 2000, LNAI 1910, pp. 587-592, 2000.

understanding of customers is needed to send the right offers to the right customers at the most appropriate time. Knowledge about customers can lead to ideas about future offers which will meet their needs.

2 Information Systems and Decision Tables

An information system is used for representing knowledge. Its definition, presented here, is due to Pawlak [2] .

By an information system we mean a pair $S = (U,A)$, where:

1. U is a nonempty, finite set called the universe,
2. A is a nonempty, finite set of attributes i.e. $a:U \longrightarrow V_a$ for $a \in A$, where V_a is called the domain of a.

Elements of U are called objects. In our paper objects are interpreted as customers. Attributes are interpreted as features, offers made by a bank, characteristic conditions etc.

In this paper we consider a special case of information systems called decision tables [2]. In any decision table together with the set of attributes a partition of that set into conditions and decisions is given. Additionally, we assume that the set of conditions is partitioned into stable conditions and flexible conditions. We consider decision tables with only one decision attribute to be seen as “profit ranking” with values being integers. This attribute classifies objects (customers) with respect to the profit for a bank. Date of Birth is an example of a stable attribute. Interest rate on any customer account is an example of a flexible attribute (dependable on bank). We adopt the following definition of a decision table:

A decision table is any information system of the form $S = (U, A_1 \cup A_2 \cup \{d\})$, where $d \notin A_1 \cup A_2$ is a distinguished attribute called decision. The elements of A_1 are called stable conditions, whereas the elements of A_2 are called flexible conditions.

The cardinality of the image $d(U) = \{ k: d(x)=k$ for some $x \in U\}$ is called the rank of d and is denoted by r(d). Let us observe that the decision d determines the partition $CLASS_S(d) = \{X_1,X_2,\ldots,X_{r(d)}\}$ of the universe U, where $X_k = d^{-1}(\{k\})$ for $1 \leq k \leq r(d)$. $CLASS_S(d)$ is called the classification of objects in S determined by the decision d.

In this paper, as we mentioned before, objects correspond to customers. Also, we assume that customers in $d^{-1}(\{k1\})$ are more profitable for a bank than customers in $d^{-1}(\{k2\})$ for any $k2 \leq k1$. The set $d^{-1}(\{r(d)\})$ represents the most profitable customers for a bank. Clearly the goal of any bank is to increase its profit. It can be achieved by shifting some customers from the group $d^{-1}(\{k2\})$ to $d^{-1}(\{k1\})$, for any $k2 \leq k1$. Namely, through special offers made by a bank, values of flexible attributes of some customers can be changed and the same all these customers can be moved from a group of a lower profit ranking to a group of a higher profit ranking.

Careful analysis of decision tables classifying customers can help us to identify groups of customers within any $d^{-1}(\{k1\})$ who may accept attractive offers from a competitive bank. In some of these cases, our action rules will suggest what offers should be sent to these customers moving them to a lower profit ranking (but still profitable for the bank) group instead of loosing them entirely.

3 Action Rules

In this section we describe a method to construct action rules from a decision table containing both stable and flexible attributes.

Before we introduce several new definitions, assume that for any two collections of sets X, Y, we write, $X \subseteq Y$ if $(\forall x\in X)(\exists y\in Y)[\ x\subseteq y\ ]$. Let $S = (U, A_1 \cup A_2 \cup \{d\})$ be a decision table and $B \subseteq A_1 \cup A_2$. We say that attribute d depends on B if $CLASS_S(B) \subseteq CLASS_S(d)$, where $CLASS_S(B)$ is a partition of U generated by B (see [2]). Assume now that attribute d depends on B where $B \subseteq A_1 \cup A_2$. The set B is called d-reduct in S if there is no proper subset C of B such that d depends on C. The concept of d-reduct in S was introduced to induce rules from S describing values of the attribute d depending on minimal subsets of $A_1 \cup A_2$. In order to induce rules in which the THEN part consists of the decision attribute d and the IF part consists of attributes belonging to $A_1 \cup A_2$, subtables $(U, B \cup \{d\})$ of S where B is a d-reduct in S should be used for rules extraction. By Dom(r) we mean all attributes listed in the IF part of a rule r. For example, if r = [(a1,3)*(a2,4) → (d,3)] is a rule then Dom(r) = {a1,a2}. By d(r) we denote the decision value of a rule. In our example d(r) = 3.

If r1, r2 are rules and $B \subseteq A_1 \cup A_2$ is a set of attributes, then r1/B = r2/B means that the conditional parts of rules r1, r2 restricted to attributes B are the same. For example if r1 = [(a1,3) → (d,3)], then r1/{a1} = r/{a1}.

Example 1. Assume that S = ({x1,x2,x3,x4,x5,x6,x7,x8}, {a,c} {b} {d}) be a decision table represented by Figure 1. The set {a,c} lists stable attributes, in {b} we have flexible attributes and d is a decision attribute. Also, we assume that H denotes customers of a high profit ranking and L denotes customers of a low profit ranking.

In our example r(d)=2, r(c)=3, r(a)=2,
$CLASS_S$(d)={{x1,x2,x3,x4,x5,x6},{x7,x8}}, $CLASS_S$({b})={{x1,x3,x7,x8}, {x2,x4},{x5,x6}}, CLASS({a,b})={{x1,x3},{x2,x4},{x5,x6}, {x7,x8}},
$CLASS_S$({a})={{x1,x2,x3,x4},{x5,x6,x7,x8}}, $CLASS_S$({c})={{x1,x3},{x2,x4}, {x5,x6,x7,x8}}, CLASS({b,c})={{x1,x3},{x2,x4},{x5,x6},{x7,x8}}.

So, CLASS({a,b}) $\subseteq$ $CLASS_S$(d) and CLASS({b,c}) $\subseteq$ $CLASS_S$(d). It can be easily checked that both {b,c} and {a,b} are d-reducts in S.

Rules can be directly derived from d-reducts and the information system S. In our example, we get the following rules:

$(a,0)\wedge(b,S) \rightarrow (d,L)$, $(b,S)\wedge(c,0) \rightarrow (d,L)$,
$(a,0)\wedge(b,R) \rightarrow (d,L)$, $(b,R)\wedge(c,1) \rightarrow (d,L)$,
$(a,2)\wedge(b,P) \rightarrow (d,L)$, $(b,P)\wedge(c,2) \rightarrow (d,L)$,
$(a,2)\wedge(b,S) \rightarrow (d,H)$, $(b,S)\wedge(c,2) \rightarrow (d,H)$.

	a	b	c	d
x1	**0**	**S**	**0**	**L**
x2	**0**	**R**	**1**	**L**
x3	**0**	**S**	**0**	**L**
x4	**0**	**R**	**1**	**L**
x5	**2**	**P**	**2**	**L**
x6	**2**	**P**	**2**	**L**
x7	**2**	**S**	**2**	**H**
x8	**2**	**S**	**2**	**H**

Fig. 1

We use information system S to simplify them. We get :

$(a,0) \to (d,L)$, $(c,0) \to (d,L)$,
$(b,R) \to (d,L)$, $(c,1) \to (d,L)$,
$(b,P) \to (d,L)$, $(a,2)\wedge(b,S) \to (d,H)$, $(b,S)\wedge(c,2) \to (d,H)$.

Now, let us assume that $(a, v \to w)$ denotes the fact that the value of attribute a has been changed from v to w. Similarly, the term $(a, v \to w)(x)$ means that a(x)=v has been changed to a(x)=w. Saying another words, the property (a,v) of a customer x has been changed to property (a,w).

Assume now that $S = (U, A_1 \cup A_2 \cup \{d\})$ is a decision table, where A_1 is the set of stable attributes and A_2 is the set of flexible attributes. Assume that rules r1, r2 have been extracted from S and $r1/A_1 = r2/A_1$, d(r1)=k1, d(r2)=k2 and k1< k2. Also, assume that (b1, b2,…, bp) is a list of all attributes in $Dom(r1) \cap Dom(r2) \cap A_2$ on which r1, r2 differ and r1(b1)= v1, r1(b2)= v2,…, r1(bp)= vp, r2(b1)= w1, r2(b2)= w2,…, r2(bp)= wp.

By (r1,r2)-action rule on $x \in U$ we mean a statement:

$[(b1, v1 \to w1) \wedge (b2, v2 \to w2) \wedge \ldots \wedge (bp, vp \to wp)](x) \Rightarrow [(d,k1) \to (d,k2)](x)$.

If the value of the rule on x is true then the rule is valid. Otherwise it is false.

Let us denote by $U^{<r1>}$ the set of all customers in U supporting the rule r1. If (r1,r2)-action rule is valid on $x \in U^{<r1>}$ then we say that the action rule supports the new profit ranking k2 for x.

Example 2. Assume that $S = (U, A_1 \cup A_2 \cup \{d\})$ is a decision table from the Example 1, $A_2=\{b\}$, $A_1 =\{a,c\}$. It can be checked that rules $r1=[(b,P) \to (d,L)]$, $r2=[(a,2)\wedge(b,S) \to (d,H)]$, $r3=[(b,S)\wedge(c,2) \to (d,H)]$ can be extracted from S. Clearly x5, x6 $\in U^{<r1>}$. Now, we can construct (r1,r2)-action rule executed on x:

$[(b, P \to S)](x) \Rightarrow [(d,L) \to (d,H)](x)$.

It can be checked that this action rule supports new profit ranking H for x5 and x6.

Example 3. Assume that $S = (U, A_1 \cup A_2 \cup \{d\})$ is a decision table represented by Figure 2. Assume that $A_1= \{c, b\}$, $A_2 = \{a\}$.

	c	a	b	d
x1	2	1	1	L
x2	1	2	2	L
x3	2	2	1	H
x4	1	1	1	L

Fig. 2

Clearly $r1=[(a,1)\wedge(b,1) \rightarrow (d,L)]$, $r2=[(c,2)\wedge(a,2) \rightarrow (d,H)]$ are optimal rules which can be extracted from S. Also, $U^{<r1>} = \{x1, x4\}$. If we construct (r1,r2)-action rule: $[(a, 1 \rightarrow 2)](x) \Rightarrow [(d,L) \rightarrow (d,H)](x)$.
then it will certainly support new profit ranking H for x1 but only possibly for x4.

Algorithm to Construct Action Rules

```
Input:    Decision table  $S = (U, A_1 \cup A_2 \cup \{d\})$,
          $A_1$ – stable attributes, $A_2$ – flexible attributes, $\lambda1$, $\lambda2$ – weights.
Output:   R – set of action rules.

Step 0.  $R := \emptyset$.
Step 1.  Find all d-reducts $\{D_1, D_2,\ldots, D_m\}$ in S which satisfy the property
            $card[D_i \cap A_1]/card[A_1 \cup A_2] \leq \lambda1$
            (reducts with a relatively small number of stable attributes)
Step 2.  FOR EACH pair $(D_i, D_j)$ of d-reducts (found in step 1) satisfying the
         property $card(D_i \cap D_j)/card(D_i \cup D_j) \leq \lambda2$ DO
         find set $R_i$ of optimal rules in S using d-reduct $D_i$,
         find set $R_j$ of optimal rules in S using d-reduct $D_j$.
Step 3.  FOR EACH pair of rules (r1, r2) in $R_i \times R_j$ having different THEN
         parts DO
         if $r1/A_1 = r2/A_1$, d(r1)=k1, d(r2)=k2 and k1< k2, then
               if (b1, b2,…, bp) is a list of all attributes in $Dom(r1) \cap Dom(r2)$
            $\cap A_2$ on which r1, r2 differ and
               r1(b1)= v1, r1(b2)= v2,…, r1(bp)= vp,
               r2(b1)= w1, r2(b2)= w2,…, r2(bp)= wp
                 then the following (r1,r2)-action rule add to R:
                     if $[(b1, v1\rightarrow w1) \wedge (b2, v2 \rightarrow w2) \wedge\ldots\wedge (bp, vp \rightarrow$
                           $wp)](x)$ then $[(d, k1) \rightarrow (d,k2)](x)$
```

The resulting (r1,r2)-action rule says that if the change of values of attributes of a customer *x* will match the term
$[(b1, v1\rightarrow w1) \wedge (b2, v2 \rightarrow w2) \wedge\ldots\wedge (bp, vp \rightarrow wp)](x)$
then the ranking profit of custumer x will change from *k1* to *k2.*

This algorithm was initially tested on a sampling containing 20,000 tuples extracted randomly from a large banking database containing more than 10 milion customers. We used DataLogicR$^+$ for rule extraction. Both “roughness” and “rule precision threshold” has been set up to 0.85. We found many pairs of rules (r1,r2) meeting the conditions required by the above algorithm for generating (r1,r2)-action rules. For instance, we extracted the following three rules:

$[A79 > 610.70] \wedge [A75 > 640.50] \wedge [A73 > 444.00] \longrightarrow [\text{ PROFIT} = 1],$

$[A79 \leq 610.70] \wedge [A75 > 640.50] \wedge [A78 \leq 394.50] \longrightarrow [\text{ PROFIT} = 1],$

$[A75 \leq 640.50] \longrightarrow [\text{ PROFIT} = 3].$

Attributes A73, A75, A78, A79 are all flexible.

The following action rule can be generated:

if $[A75, (A75 > 640.50) \longrightarrow (A75 \leq 640.50)](x)$

then $[(\text{PROFIT} = 1) \longrightarrow (\text{PROFIT} = 3)]$.

Similar method is used to identify groups of customers who may accept attractive offers from a competitive bank. In such a case, action rules suggest what offers should be sent to all these customers moving them to lower profit ranking (but still profitable for the bank) groups instead of loosing them entirely.

4 Conclusion

Proposed algorithms identify the customers which may move from one profit ranking group to another profit ranking group if values of some flexible attributes describing them are changed. Also, the algorithms show what attribute values should be changed and what their new values should be (what type of offers should be sent by a bank to all these customers).

References

1. Chmielewski M. R., Grzymala-Busse J. W., Peterson N. W., Than S., “The Rule Induction System LERS - a Version for Personal Computers”, in *Foundations of Computing and Decision Sciences*, Vol. 18, No. 3-4, 1993, Institute of Computing Science, Technical University of Poznan, Poland, 181-212.
2. Pawlak Z., “Rough Sets and Decision Tables”, in *Lecture Notes in Computer Science* **208,** Springer-Verlag, 1985, 186-196.
3. Skowron A., Grzymala-Busse J., “From the Rough Set Theory to the Evidence Theory”, *in ICS Research Reports* 8/91, Warsaw University of Technology, October, 1991