Data Mining and Knowledge Discovery, 14:225-243, 2007

Evaluation of ordinal attributes at value level

Marko Robnik-Šikonja*, Koen Vanhoof**

*University of Ljubljana,
Faculty of Computer and Information Science,
Tržaška 25, 1001 Ljubljana, Slovenia
Marko.Robnik@fri.uni-lj.si
** University of Hasselt, Department of Economics,
Universitaire Campus, 3590 Diepenbeek, Belgium
koen.vanhoof@uhasselt.be

Abstract

We propose a novel context sensitive algorithm for evaluation of ordinal attributes which exploits the information hidden in ordering of attributes' and class' values and provides a separate score for each value of the attribute.

Similar to feature selection algorithm ReliefF, the proposed algorithm exploits the contextual information via selection of nearest instances. The *ordEval* algorithm outputs probabilistic factors corresponding to the effect an increase/decrease of attribute's value has on the class value.

While the ordEval algorithm is general and can be used for analysis of any survey with graded answers, we show its utility on an important marketing problem of customer (dis)satisfaction. We develop a visualization technique and show how we can use it to detect and confirm several findings from marketing theory.

Keywords: attribute evaluation, ordinal attributes, attribute values, visualization, marketing

1 Introduction

Feature (attribute) evaluation is an important component of many machine learning tasks, e.g., feature subset selection, constructive induction, decision and regression tree learning. Scores assigned to attributes during evaluation, also provide an important information to the domain expert trying to get an insight into the problem domain.

In this work we study a subclass of feature evaluation, namely evaluation of conditionally strongly dependent ordinal attributes where each of the individual attribute's values may be dependent on other attributes in a different manner.

Our work was originally motivated by customer (dis)satisfaction analysis which is an important problem from marketing research. However, the results presented are applicable to a much broader set of problems. The algorithms and visualizations developed could be used in almost any survey analysis where the answers are given in graded or ordinal manner. To facilitate comprehension of our ideas we will give background, examples and motivation from (dis)satisfaction analysis throughout the work, but bear in mind that the approach and results are much more general.

1.1 Marketing research

A typical approach in practical marketing research of customer (dis)satisfaction is to define a number of features of the product/service and then to conduct a survey on a representative sample of customers where the customers rate their satisfaction with each of the features and also express their overall (dis)satisfaction. While all types of captured answers (features) can be integrated with our approach, we are interested only in those questions of the survey which correspond to the product/service features. We consider them attributes in our data set and the overall (dis)satisfaction corresponds to the class value. The goal of feature analysis in marketing research is manifold:

1. identify features which influence the overall (dis)satisfaction most,
2. identify type of features: marketing research differentiates mostly between three types of important features:
   (a) *basic features* are taken for granted by customers. High score in these features does not significantly increase the overall satisfaction, while a low score usually causes dissatisfaction.
   (b) *performance features* are important features not taken for granted; they usually have a positive correlation with overall satisfaction: the higher the score the bigger the effect on the overall satisfaction,
   (c) *excitement features* usually describes properties of product/service which are normally not very important to the users, but can cause excitement (and boost in satisfaction) if the score is very high.
3. identify those attribute values (thresholds) which have positive/negative impact on overall satisfaction, and
4. identify typical behavior patterns of attribute values:
   (a) *upward reinforcement*: the value of a feature has a positive effect on overall satisfaction,
   (b) *downward reinforcement*: the value of a feature has a negative effect on overall satisfaction,
   (c) *anchoring*: the value of a features acts as an anchor on overall (dis)satisfaction and prevents its change,
   (d) *compensation* is a type of behavior characteristic for subsets of features, namely low score in one of the features is compensated with high score in one or more others.

There are many different feature evaluation algorithms in data mining which can evaluate, rank and/or select the most informative features (goal 1). For problems with highly dependent features as is predominantly the case in the marketing research, the most suitable heuristics are probably ReliefF (Kononenko, 1994) and CM (Hong, 1997). Other goals (2-4) remain mostly untouched by current work in machine learning and data mining.

1.2 Feature evaluation

The attribute evaluation measure we need for these type of problems would ideally possess the following properties.

- *Context sensitivity*: typically the attributes in such studies are highly conditionally dependent and have to be evaluated in the context of other attributes.
- *Ability to handle ordered attributes and ordered class* and to use the information the ordering contains. The order of attribute's values contains information which is comparable but not the same as values of numerical attributes, e.g., attribute values poor, good, very good, excellent are ordered in expressing satisfaction but this ordering is not necessarily linear.
- *Awareness of the meaning implied by the ordering* of attributes and the positive (negative) correlation of changes between attribute and class value (e.g, if the value of the attribute increases from poor to good, we have to be able to detect both positive and negative correlation to the change of class' value).
- *Ability to handle each value of the attribute separately*, e.g., for some attributes the value of good and very good have identical neutral impact on the class, value poor may have strong negative, and value excellent highly positive impact.
- *Visualization* of the output which will enable domain experts to identify type of features and the impact of their individual values.

The goal of this work is to provide an algorithm which will address goals 2,3 and 4. We present a feature evaluation algorithm which satisfies the conditions itemized above and the visualization method which presents the results in such a way that the domain expert can identify the type of the features and the impact of individual values. Our approach is based on probabilistic factors, namely for each value of the attribute we estimate two conditional probabilities: the probability that the class value increases given the increase of the attribute value, and the probability that class value decreases given the decrease of attribute value. To take context of other attributes into account we estimate these factors in the local context, from the most similar instances. We call these conditional probabilities reinforcement factors as they approximate behavior patterns described in goal 4 (upward reinforcement, downward reinforcement, and anchoring). The visualization of these factors gives clear clues about the role of each feature, its type, the importance of each value and the threshold values.

1.3 Related work

We concisely review existing feature evaluation measures and explain their properties which are relevant to our problem. The problem of feature (attribute) evaluation has received much attention in the literature. There are several measures for evaluation of attributes' quality. For classification problems the most popular are e.g., Gini index (Breiman et al., 1984), gain ratio (Quinlan, 1993), MDL (Kononenko, 1995), and ReliefF (Robnik-Šikonja and Kononenko, 2003). The first three are impurity based and measure quality of attribute according to the purity of class value distribution after the split on the values of that attribute. They evaluate each attribute separately, are not aware of the ordering of the attribute's values and cannot provide useful information for each individual value of the attribute. ReliefF on the other hand is context sensitive (by measuring how the attribute separates similar instances) and could be adapted to handle ordered attributes (by changing the definition of its similarity measure), but cannot provide information for each value separately and does not differentiate between the positive and negative changes of the attribute and their impact on the class value. By converting ordered nominal attributes into numeric ones we could use RReliefF (regression version of ReliefF) which, at the cost of assuming linear ordering, naturally handles positive and negative changes of attribute, but cannot separate between positive and negative impact on class value; also the extraction of information for each individual attribute's value is not possible in the RReliefF.

In short, to our knowledge, no existing attribute evaluation measure is suitable for our problem, but they do provide useful hints as how to handle presented problems.

1.4 Notation and organization

Throughout the paper we use a notation where each of the n learning instances I is represented by an ordered pair $(\mathbf{x}, y)$, where each vector of attributes $\mathbf{x}$ consists of individual attributes A_i, $i = 1, ..., a$, (a is the number of attributes) and is labeled with one of the ordered class values y_j, $j = 1, ..., c$ (c is the number of class values) $y_1 < y_2 < ... < y_c$. Each discrete ordered attribute A_i has values $A_{i,1}$ through A_{i,m_i} (m_i is the number of values of attribute A_i), $A_{i,1} < A_{i,2} < ... < A_{i,m_i}$. We use $y(I_t)$ in a functional form when we refer to the class value of the t-th instance and $I_{t,i}$ when we refer to the value of the attribute A_i for the t-th instance. We write $p(y_j)$ as the probability of the class value y_j.

The paper is organized into 5 sections. In Section 2 we formally define reinforcement factors for values of attributes, and propose a new attribute evaluation algorithm ordEval for their approximation. In Section 3 we define two artificial data sets to show the utility of ordEval on it and present two visualization method for the output of the new algorithm. Section 4 contains some background information on (dis)satisfaction analysis, a short description of our marketing research data sets, and an evaluation of our algorithm and its visualizations on these data sets. Section 5 summarizes and gives some ideas for further work.

2 Ordered attribute values evaluation

A common thread of goals 2-4 is a need to observe not the feature as a whole, but rather the effect a single value of the attribute may have. The problem we are tackling is to evaluate ordered attribute values in the context of other attributes where the class values are also ordered. Assume for a moment that we could observe the inner workings of the decision process which forms relationship between the features and the class value. In other words, assume that we can observe a causal effect the change of an attribute's value has on the class value. By measuring such an effect we could reason about the importance of the attribute's values and type of the attribute, we could determine which values are thresholds for change of behavior, and we could characterize the behaviors. This is of course impossible, but we can use our data sample and try to approximate this reasoning.

Without loss of the generality we can assume that for important attributes a positive change in the value of attribute should be positively correlated with the positive change in the class value (if this is not the case for one of the attributes, we can reverse ordering of the attribute). For a randomly selected instance we find the most similar instance (similarity is defined as the distance in the space of features). Note that this similar instance will have many similar attribute values and only a few different ones. If the class value of the similar instance is different as well, then we assume that the cause of these differences are the differences in attributes' values and we give these values some credit for that - but only if the sign of the differences in class and attribute is the same (otherwise we are dealing with noise, or some other attribute was responsible for these differences).

To wrap these idea into algorithm we need some definitions. Let R be a randomly selected instance and S its most similar instance. Let j be the value of attribute A_i at instance R. We observe the necessary changes of class value and attributes (A_i in particular) which would change S to R. If these changes are positive (increase of class and/or attribute values), let

- $P(C^p_{i,j})$ be a probability that the class value of R is larger than the class value of its most similar neighbor S. $P(C^p_{i,j})$ is therefore the probability that the positive change in similar instance's class value is needed to get from S to R.
- $P(A^p_{i,j})$ be a probability that j (the value of A_i at R) is larger than the value of A_i at its most similar neighbor S. By estimating $P(A^p_{i,j})$ we gather evidence of the probability that the similar instance S has lower value of A_i and the change of S to R is positive.
- $P(C^pA^p_{i,j})$ be a probability that both the class and j (the value of A_i at R) are larger than the class and attribute value of its most similar neighbor S. With $P(C^pA^p_{i,j})$ we estimate the probability that positive change in both the class and A_i value of similar instance S is needed to get the values of R.

Similarly we define for negative changes which would turn S into R (decrease of class and/or attribute values). Let

- $P(C^n_{i,j})$ be a probability that the class value of R is smaller than the class value of its most similar neighbor S. $P(C^n_{i,j})$ is therefore the probability that the negative change in similar instance's class value is needed to get from S to R.
- $P(A^n_{i,j})$ be a probability that j (the value of A_i at R) is smaller than the value of A_i at its most similar neighbor S. By estimating $P(A^n_{i,j})$ we gather evidence of the probability that the similar instance S has larger value of A_i and the change of S to R is negative.
- $P(C^nA^n_{i,j})$ be a probability that both the class and j (the value of A_i at R) are smaller than the class and attribute value of its most similar neighbor S. With $P(C^nA^n_{i,j})$ we estimate the probability that negative change in both the class and A_i of similar instance S is needed to get the values of R.

We define two factors, upward and downward reinforcement, which measure the upward/downward trends exhibited in the data. The upward reinforcement of the i-th attribute's value j is

$$U_{i,j} = P(C^p_{i,j}|A^p_{i,j}) = \frac{P(C^pA^p_{i,j})}{P(A^p_{i,j})} \tag{1}$$

This factor reports the probability that a positive class change is caused by positive attribute change. This intuitively corresponds to the effect the positive change in attribute's value has on the class. Similarly we define downward reinforcement as

$$D_{i,j} = P(C^n_{i,j}|A^n_{i,j}) = \frac{P(C^n A^n_{i,j})}{P(A^n_{i,j})} \tag{2}$$

It reports the effect the decrease of attribute's value has on the decrease of class' value. Analogously with numerical features we could say that U and D are similar to the partial derivatives of the prediction function,.

Note that U and D contain a great deal of information we need if we want to fulfill the goals stated in Section 1. What remains is to estimate them reliably. To meet this end we borrow some ideas from feature evaluation algorithm ReliefF (Robnik-Šikonja and Kononenko, 2003). Specifically we randomly sample the instances (in case of small data set we can take all instances) and search for several most similar instances and then weight their contribution with their distance from the randomly selected instance.

For each value of each attribute we count the number of positive, negative and no changes which result in increase, decrease, and no change in the class' value. As we are only interested in a trend the attribute changes cause (class value increase or decrease) and not in exact class value we have to define the distance function appropriately. The complete algorithm ordEval is in Figure 1.

We first set counters of co-occurring changes to zero (line 3). We randomly select an instance (line 5) and search for its k nearest instances (line 6). For each of the nearest instances we update our counters for all the attributes depending on the class and attribute values of the randomly selected instance and the near instance (lines 9-18): if the attribute value of the near instance is lower than the attribute value of the random instance (line 9) then the change is positive and we update A^p for given value of the attribute (line 10); if additionally the class value of the near instance is lower than class value of random instance (line 11) then the change in both class and attribute is positive and we update C^pA^p value for given attribute and its value of random instance (line 12). Similarly we do for negative changes in attribute and class (lines 14-18). We repeat the whole process for a pre-specified number of iterations. Conservatively we can set this number to equal the number of instances available, but similarly to ReliefF we get useful results even if we run only a few iterations (e.g., sampling without replacement for $m = \log n$). The upward and downward enforcement factors for all the values of attributes are computed in lines 24 and 25, respectively.

The updates depend on the values of attribute A at the random instances R and its near instance N. The simplest form of update function $w(R,N)$ (see Eq. (3)) takes into account only the number of nearest instances (k). The idea is to average the results of k nearest instances.

$$w(R,N) = \frac{1}{k} \tag{3}$$

In our experiments we used value $k = 10$ as this is the default value of most k-nearest neighbor classification studies. Depending on the nature of the problem other values could be more suitable, but this is not the topic of this work. For discussion see (Duda et al., 2001). A more sophisticated version of the updated function takes also the distance between the instances into account: closer instances should have greater impact. Exponentially decreasing weighted contribution of instances ranked by distance is recommended by (Robnik-Šikonja and Kononenko, 2003)) in this case, see this reference for detailed explanation of this.

Algorithm ordEval
Input: for each instance a vector of attribute values and the class value
m - the number of iterations, for small sample set to the number of instances n
Output: $U_{i,j}$ and $D_{i,j}$ for all attributes' values

for i = 1 **to** a **do** // for all the attributes
    for j = 1 **to** m_i **do** // and their values
        $A^p_{i,j} = C^p A^p_{i,j} = A^n_{i,j} = C^n A^n_{i,j} = 0.0$ // initialization
for s = 1 **to** m **do begin** // for prespecified number of samples
    randomly select an instance R_s
    find k nearest instances $\mathscr{N}$, closest to R_s
    for each near instance $N_u \in \mathscr{N}$ **do begin**
        for i = 1 **to** a **do begin** // for all the attributes
            if $N_{u,i} < R_{s,i}$ **then begin**
                $A^p_{i,R_{s,i}}$ += $w(R_{s,i}, N_{u,i})$
                if $y(N_u) < y(R_s)$ **then**
                    $C^p A^p_{i,R_{s,i}}$ += $w(R_{s,i}, N_{u,i})$
            end
            else if $N_{u,i} > R_{s,i}$ **then begin**
                $A^n_{i,R_{s,i}}$ += $w(R_{s,i}, N_{u,i})$
                if $y(N_u) > y(R_s)$ **then**
                    $C^n A^n_{i,R_{s,i}}$ += $w(R_{s,i}, N_{u,i})$
            end
        end // for attributes
    end // for near instances
end // for random instance R_i
for i = 1 **to** a **do** // for all the attributes
    for j = 1 **to** m_i **do begin** // and their values
        $U_{i,j} = C^p A^p_{i,j} \,/\, A^p_{i,j}$
        $D_{i,j} = C^n A^n_{i,j} \,/\, A^n_{i,j}$
    end

Figure 1: Pseudo code of OrdEval algorithm

Missing entries which frequently occur in these types of problems can simply be excluded from computation. The similarity of instances is computed as Manhattan distance over all the attributes:

$$d(I_t, I_u) = \sum_{i=1}^{a} \mathrm{diff}(A_i, I_t, I_u). \tag{4}$$

The single attribute distance for two instances is computed with the function $\mathrm{diff}(A_v, I_t, I_u)$. For ordinal attributes, when we do not have better domain knowledge available we can assume linear ordering:

$$\mathrm{diff}(A_i, I_t, I_u) = \frac{|I_{t,i} - I_{t,i}|}{m_i}. \tag{5}$$

The computational complexity of the algorithm is of the order $O(m \cdot n \cdot a)$. The main computational burden within each of the m iterations is the search for the nearest instances (line 6), for which we have to compute the distances to all the instances ($O(n \cdot A)$). If the number of features is low we can reduce this computation by the use of smarter data structures or we can investigate performance of approximate algorithms (Duda et al., 2001). Each iteration of the algorithm is independent and here we see a path to parallelization of the algorithm.

Our algorithm does not cover all the information which could be extracted with the help of class and feature value changes in the nearest neighbor context. Important aspect could also be hidden in the amount of positive and negative changes resulting from no changes in attribute's value. This could be a useful hint about the coherency in the data, noise, overall importance of the attribute as well as the amount of dependency between the attributes. Also important is the anchoring effect uncovered by our algorithm, but this remains as further work.

3 Visualization of enforcement factors

While $U_{i,j}$ and $D_{i,j}$ factors contain all important information, we want to make them more comprehensive and extract trends hidden in them. In other words, we want to present and visualize these results in a way which will enable domain experts to see behavior patterns of the attributes and possibly to determine for each individual attribute the values where upward/downward reinforcement is significant.

To test the behavior of our algorithm we first define simple artificial problem which is described by three important and one random feature. The important feature correspond different feature types from the marketing theory (basic B, performance P, and excitement E). The values of all features are randomly generated integer values from 1 to 5. The dependent variable for each instance (class) is the sum of its features' effects, which we scale to the integers 1-5.

$$C = b(B) + p(P) + e(E)$$

Basic features are taken for granted by customers; high score in these features does not significantly increase the overall satisfaction, while a low score causes dissatisfaction. We define effect of B as

$$b(B) = \left\{ \begin{array}{cl} -1 & ; B <= 3 \\ 1 & ; B > 3 \end{array} \right\}.$$

Performance features have positive correlation with overall satisfaction: the higher the score the bigger the effect on the overall satisfaction, low score causes dissatisfaction. We define effect of P as

$$p(P) = P - 2.$$

Excitement features describe properties of product/service which are normally not very important to the users, but can cause excitement (if the score is very high). We define the effect of E as

$$e(E) = \left\{ \begin{array}{cl} 0 & ; E <= 4 \\ 2 & ; E = 5 \end{array} \right\}.$$

Table 1: The output of ordEval algorithm for attributes in FT data set.

attribute	value	$U_{i,j}$	$D_{i,j}$
B	1	0.000	0.020
	2	0.045	0.066
	3	0.020	0.994
	4	1.000	0.006
	5	0.072	0.000
P	1	0.000	0.107
	2	0.083	0.922
	3	0.909	0.441
	4	0.493	0.438
	5	0.485	0.000
E	1	0.000	0.023
	2	0.008	0.039
	3	0.010	0.074
	4	0.003	0.984
	5	0.972	0.000
R	1	0.000	0.021
	2	0.016	0.006
	3	0.032	0.049
	4	0.010	0.003
	5	0.033	0.000

We generated 1000 instances for this FT (Feature Type) data set. Table 1 shows the upward and downward enforcement factors the ordEval algorithm (Figure 1) returned for this data set.

Of course the extreme values $U_{i,1}$ and $D_{i,5}$ are 0 (there is no smaller value than 1 and larger than 5). We can observe that the algorithm has captured the important landmarks of the features:

- for basic attribute B the strong upward enforcement for value 4 (moving from lower value to 4, strongly increases class value), and downward enforcement of value 3 (decreasing value to 3 strongly decreases class value),
- for performance attribute P the strongest jump is upward enforcement to 3, but 4 and 5 are strong too; similarly for downward enforcement values 2, 3 and 4 are important,
- in excitement feature E the jump from 4 to 5 and back is detected in upward and downward enforcement, respectively,
- random feature R shows no important values.

The visualization we propose is conveying the information with the slope of the line segments between attribute values. The upward and downward reinforcement numbers represent the steepness of the line segment between two consecutive feature values (coefficient of the straight line between the two values) e.g., 0 is horizontal line, 1 corresponds to 45 degrees angle (the maximum), 0.5 $\approx$ 26.6 degrees angle, etc. We visualize $U_{i,j}$ and $D_{i,j}$ for all the values of one attribute in a single graph. Figure 2 shows enforcements factors for the whole FT problem. Feature values are displayed on horizontal axis, and the cumulative enforcement on the vertical axis. Upward reinforcement factors (red line) cover the upper part of the graph and downward reinforcement factors (blue line) reside in the lower part of the graph. Arrows indicate the direction

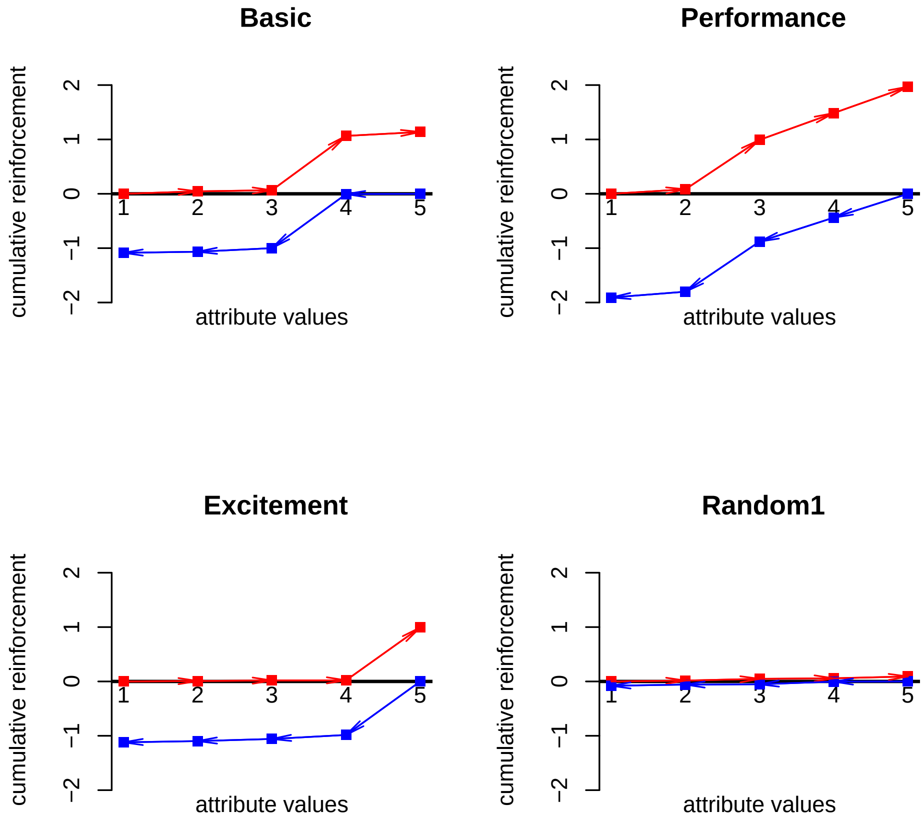

Figure 2: The slope visualization of reinforcement factors for FT data set.

of the change. Beside slopes note also the total cumulative height of the slopes as it is indicative of the overall importance of the feature: for Performance attribute several values are important, and the cumulative value gets close to 2, for Excitement there is a single important value, so the cumulative slope is around 1, and the Random attribute stays close to 0.

We have shown the work of our algorithm and its visualization on the artificial data sets. The approach taken so far has been general and can be applied to any survey with graded answers. In next Section we describe a case study on two customer (dis)satisfaction data sets which initiated our interest in evaluation of ordered attribute values.

4 Motivation problem: customer satisfaction analysis in marketing

Over the last forty years, consumer (dis)satisfaction has taken a prominent position in the marketing research literature, e.g., (Anderson, 1973; Anderson and Sullivan, 1993; Cardozo, 1965; Churchill and Surprenant, 1982). This attention is justified since consumer (dis)satisfaction (directly or indirectly) impacts upon repurchase intention (Szymanski and Henard, 2001), consumer retention (Anderson, 1994; Mittal and Kamakura, 2001) and eventually upon firm performance (Anderson et al., 1994). Consumer (dis)satisfaction is a summarizing response that results from a consumer's post-consumption cognitive and affective evaluation of a product or service performance given pre-purchase expectations (Anderson and Sullivan, 1993; Oliver, 1993; Tse and Wilton, 1988).

Focusing on the antecedents of consumer (dis)satisfaction, two main issues dominate today's discourse: the expectancy-disconfirmation theory and the nature of the relationship between consumer (dis)satisfaction and its antecedents.

First, the expectancy-disconfirmation paradigm is a dominant framework for explaining consumer (dis)satisfaction (Oliver, 1997; Szymanski and Henard, 2001). In its basic format, the model proposes that consumers' overall (dis)satisfaction response is the result of two cognitive processes (Oliver, 1997). In the first, consumers form pre-purchase expectations on the performance of a product or service. In the second, consumers evaluate the actual performance of the product and compare this perceived performance to their expectations. If performance meets expectations, consumers experience confirmation of their expectations. If performance is greater than expected, they experience positive disconfirmation; if performance is less than expected, consumers experience negative disconfirmation (see (Oliver, 1997; Yi, 1991) for reviews).

Second and related to the first is the debate on the nature of the relationship between consumer (dis)satisfaction and its antecedents. Initially, the effects of the antecedents and in particular of attribute-level performance on consumer (dis)satisfaction were assumed linear and symmetric (Mittal et al., 1998; Sethi and King, 1999; Spreng et al., 1996). Only recently, marketing scholars have questioned this double assumption on the basis of economic and psychological theory as well as on a better empirical insight in the satisfaction response function (Anderson and Sullivan, 1993).

The proposed attribute evaluation method attempts to extend the knowledge on the relationship between consumer (dis)satisfaction and its main antecedents. More specifically, we try to quantify and visualize the relationship between attribute-level (dis)satisfaction and overall (dis)satisfaction.

We report performance of our methods on one recent business-to-business (B2B) and one recent consumer-to-business (C2B) customer satisfaction study. For the business-to-business study the product involved is a high-tech product. Requirements are specified by the customer, the product is produced and delivered on demand. The whole process from order to delivery can take two or three months. The database provides the satisfaction scores of customers, who are active and have on-going orders. They reported their (dis)satisfaction with 11 product/service attributes as well as their overall satisfaction with the product. Overall satisfaction and attributes were measured on a 5-point scale. This data set is small (less than 100 records).

The consumer-to-business study is based on a study of a main European player in the entertainment sector. The survey is hierarchically organized. Satisfaction has been measured as general (overall satisfaction), on different dimensions (like personnel, administration, communication...) and on aspects of the dimensions (like personnel friendliness, clearness of invoice, ...). Dimensions and aspects are measured on a 10-point scale. The data set contains over 4000 instances.

Due to confidentiality we can not go into details or give all the results. Therefore we will give some examples from both data sets. The chosen attributes/dimensions are attributes/dimensions that occur in most customer satisfaction data sets.

4.1 Results and evaluation

In this section we report on the findings and types of behavior our algorithm can discover and give some relations to marketing literature. We first look at the results obtained for "information about promotions" (Figure 3), "price" (4), and "communication" (Figure 5) from C2B study.

These attributes can be classified as performance attributes. An increase (decrease) of the attribute level influences satisfaction (dissatisfaction). Several attributes of the B2B study show a very similar pattern of a basic attribute. The examples given are "product quality" (Figure 6), "technical support" (Figure 7), and "on-time delivery" (Figure 8).

A basic attribute behaves like a threshold or stepwise function: it creates (dis)satisfaction when (not) fulfilled. For example 'quality of the product' (Figure 6) is a basic requirement that should obtain the quality level 4 or higher. The threshold value can differ from customer to customer, so the pattern for the upward reinforcement is not always horizontal. The attribute "Delivery on time" (Figure 8) has low reinforcements figures. At first sight this is surprising but it is typical for the product under consideration.

Most of the attributes of the customer-to-business study and some of the business-to-business show a mixed pattern. Next figures give the results for "quality of delivery" (Figure 9) and "accuracy of administration" (Figure 10) from B2B study, and "personnel" (Figure 11) and "employee friendliness" (Figure 12) from C2B study. We can observe that only high values of attribute influence satisfaction and a decrease of the values influences dissatisfaction.

Prospect theory in general, and its assumptions of loss aversion and diminishing sensitivity in particular (Einhorn and Hogarth, 1981; Mittal and Kamakura, 2001) proposes an asymmetric S-shaped relationship between attribute-level performance and overall (dis)satisfaction. Indeed, it has been observed that the marginal contribution of attribute-level performance on overall (dis)satisfaction decreases with its size and that losses have more impact than gains. This is confirmed by evidence and theory on memory accessibility: negative information is more perceptually salient, is given more weight, and creates a stronger response than positive information (Mittal and Kamakura, 2001; Peeters and Czapinski, 1990). Our results show higher values for the downward reinforcement than for upward reinforcement. All these figures demonstrate clearly the asymmetric and non-linear nature of the relationship between consumer (dis)satisfaction and the attribute under consideration. Marketing managers are specifically interested in the height and the shape of the curves and the position of the breakpoints.

5 Conclusions

We have presented the algorithm ordEval for evaluation of ordered attributes' values. The algorithm exploits the information hidden in the ordering of class and attribute values and their inherent correlation. Based on nearest neighbor paradigm and probability theory the algorithm is context sensitive, able to handle ordered attributes and ordered classes, aware of the information the ordering contains, able to handle each value of the attribute separately, and provides output which can be effectively visualized. The

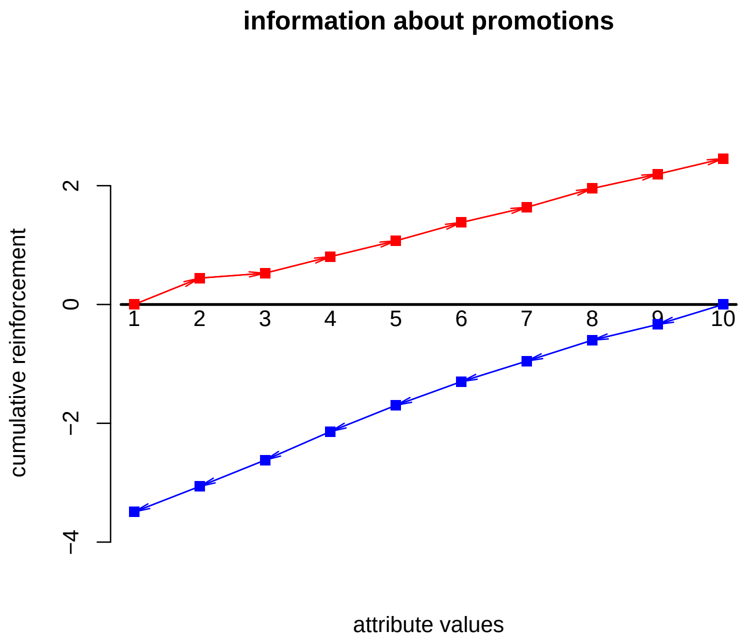

Figure 3: The results for "information about promotions" in C2B data set.

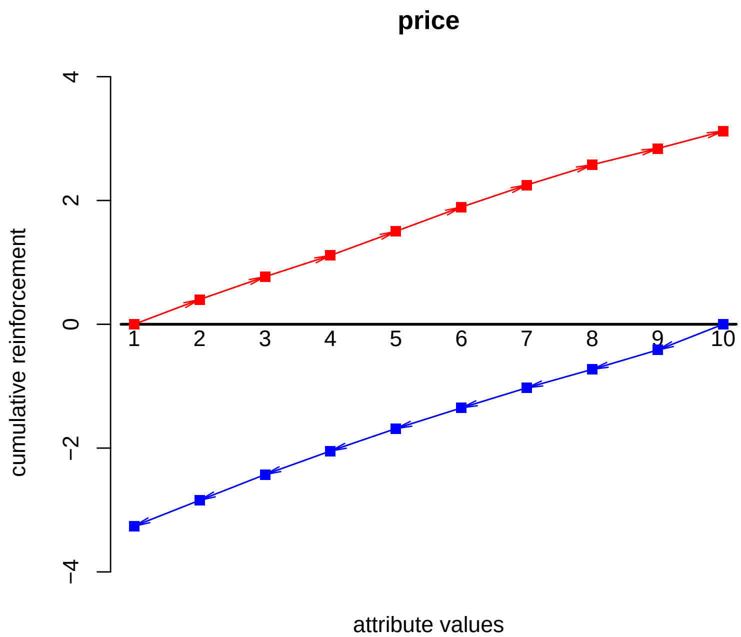

Figure 4: The results for "price" in C2B data set.

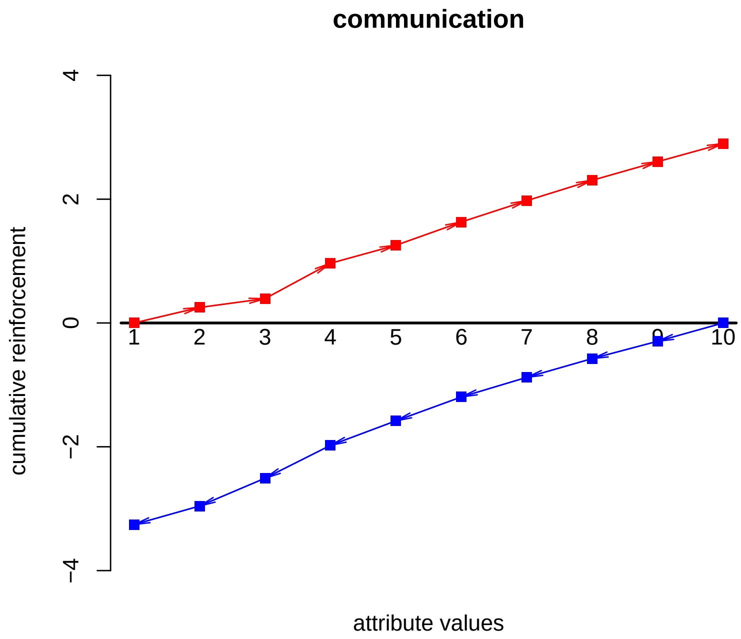

Figure 5: The results for ”communication” in C2B data set.

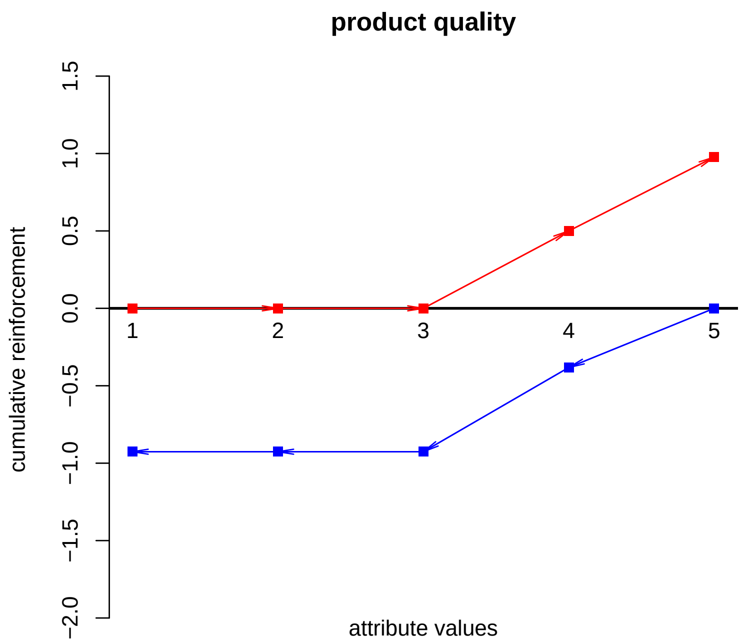

Figure 6: The results for ”product quality” in B2B data set.

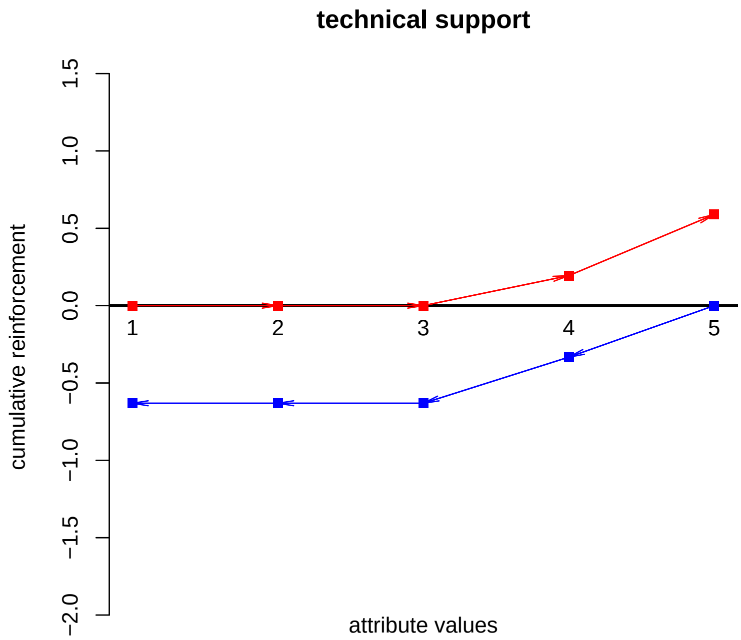

Figure 7: The results for ”technical support” in B2B data set.

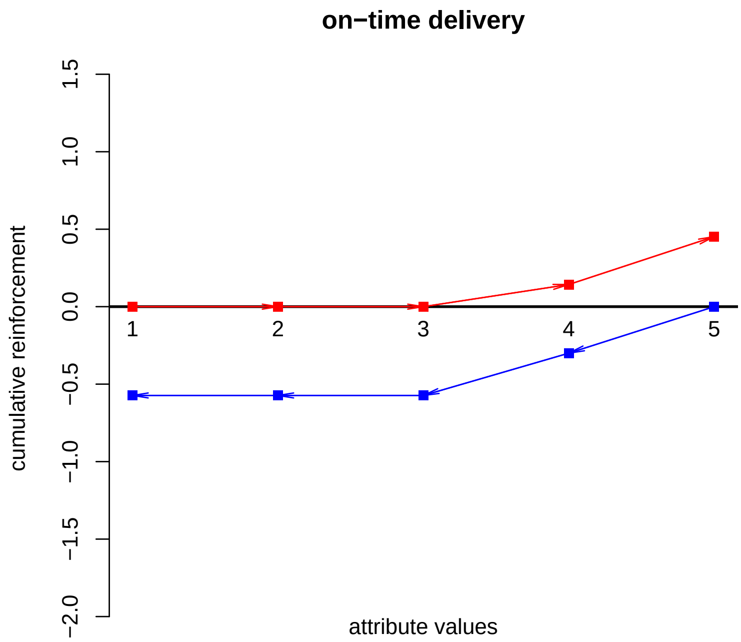

Figure 8: The results for ”on-time delivery” in B2B data set.

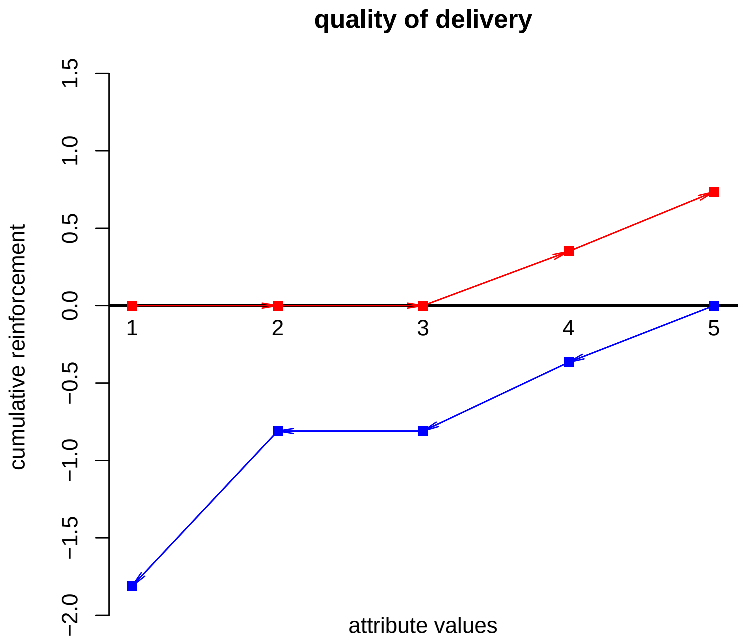

Figure 9: The results for ”quality of delivery” in B2B data set.

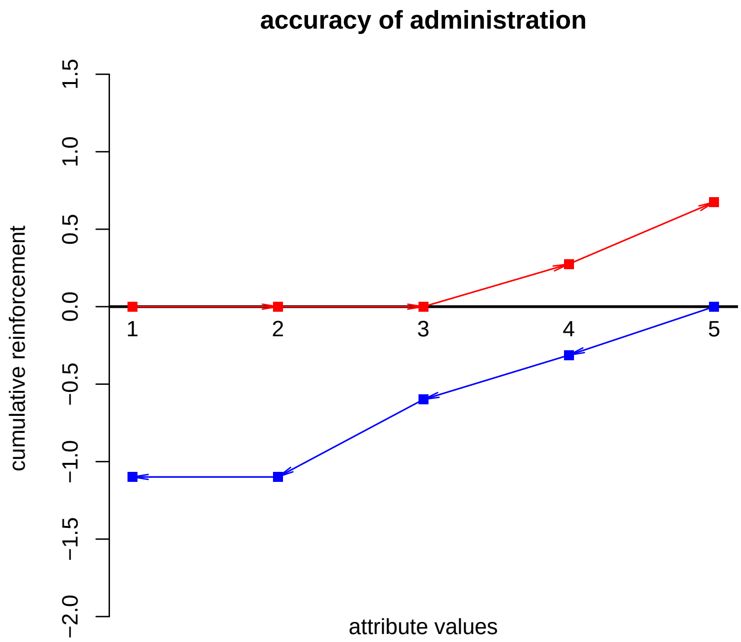

Figure 10: The results for ”accuracy of administration” in B2B data set.

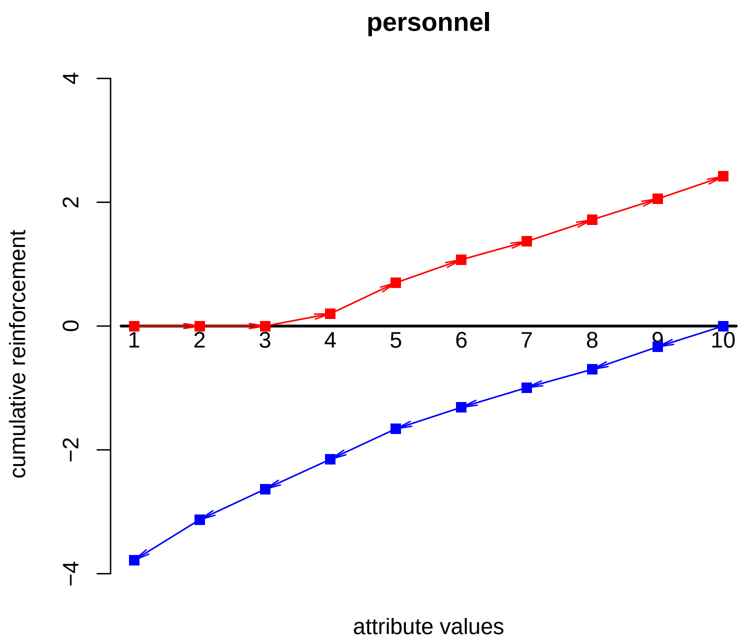

Figure 11: The results for ”personnel” in C2B data set.

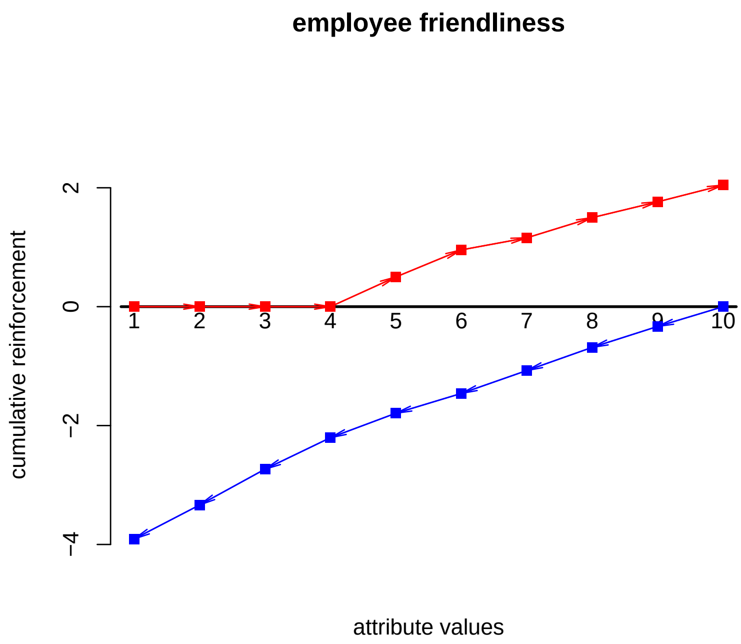

Figure 12: The results for ”employee friendliness” in C2B data set.

visualizations we developed turned out highly useful in our marketing research case study.

In marketing research, the potential of the ordered attribute evaluation to unravel the decision-making heuristics of customers when 'deciding' on a certain level of (dis)satisfaction seems to outperform that of more traditional statistical models. This is due to the power of the method to allow for non-linear and asymmetric effects as well as to the fact that researchers should not a priori postulate the roles the different attributes will take. Although the algorithm appears analytically complex, it may yield parsimonious results. This paper illustrates and confirms earlier advice that managers should identify the 'optimal' performance level for each attribute. The goal should be to optimize, not to maximize attribute-level performance at a level where the payoff in terms of overall customer (dis)satisfaction is maximized. This optimal level can be determined by analyzing the different figures offered by our method.

It is our belief, that the ordered attribute evaluation can be used in fields other than marketing. The algorithm and its visualizations can be useful in any survey analysis where the answers are graded. So far we have used only part of information hidden in the difference between class and feature values of similar instances. Other effects in marketing (such as anchoring) and new applications in other fields may demand definition of additional factors and development of novel visualization techniques.

References

E. W. Anderson. Cross-category variation in customer satisfaction and retention. *Marketing Letters*, 5 (January):19–30, 1994.

E. W. Anderson and M. Sullivan. The antecedents and consequences of customer satisfaction for firms. *Science*, 12(2):125–143, 1993.

E. W. Anderson, C. Fornell, and D. Lehmann. Customer satisfaction, market share and profitability: Findings from Sweden. *Journal of Marketing*, 58 (July):63–66, 1994.

R. E. Anderson. Consumer dissatisfaction: The effect of disconfirmed expectancy on perceived product performance. *Journal of Marketing Research*, 10 (February):38–44, 1973.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and regression trees*. Wadsworth Inc., Belmont, California, 1984.

R. N. Cardozo. An experimental study of customer effort, expectations, and satisfaction. *Journal of Marketing Research*, 2 (August):244–249, 1965.

G. A. J. Churchill and C. Surprenant. An investigation into the determinants of consumer satisfaction. *Journal of Marketing Research*, 19:491–504, 1982.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd edition)*. John Wiley & Sons, 2001.

H. J. Einhorn and R. M. Hogarth. Behavioral decision theory: Processes of judgement and choice. *Annual Review of Psychology*, 32:53–88, 1981.

S. J. Hong. Use of contextual information for feature ranking and discretization. *IEEE transactions on knowledge and data engineering*, 9(5):718–730, 1997.

I. Kononenko. Estimating attributes: analysis and extensions of Relief. In L. De Raedt and F. Bergadano, editors, *Machine Learning: ECML-94*, pages 171–182. Springer Verlag, Berlin, 1994.

I. Kononenko. On biases in estimating multi-valued attributes. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1034–1040. Morgan Kaufmann, 1995.

V. Mittal and W. Kamakura. Satisfaction, repurchase intention, and repurchase behavior: Investigating the moderating effect of customer characteristics. *Journal of Marketing Research*, 38 (February):131–142, 2001.

V. Mittal, W. J. Ross, and P. Baldasare. The asymmetric impact of negative and positive attribute-level performance on overall satisfaction and repurchase intentions. *Journal of Marketing*, 62 (January):33–47, 1998.

R. L. Oliver. Cognitive, affective and attribute bases of the satisfaction response. *Journal of Consumer Research*, 20:418–430, 1993.

R. L. Oliver. *Satisfaction: A Behavioral Perspective on the Consumer*. McGraw-Hill, New York, 1997.

G. Peeters and J. Czapinski. Positive-negative asymmetry in evaluations: The distinction between affective and informational negativity effect. *European Review of Social Psychology*, 1:33–60, 1990.

J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, 1993.

M. Robnik-Šikonja and I. Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning Journal*, 53:23–69, 2003.

V. Sethi and R. King. Nonlinear and noncompensatory models in user information satisfaction measurement. *Information Systems Research*, 10(1):87–96, 1999.

R. A. Spreng, S. McKensey, and R. Olshavsky. A reexamination of the determinants of consumer satisfaction. *Journal of Marketing*, 60 (July):15–32, 1996.

D. M. Szymanski and D. Henard. Consumer satisfaction: A meta-analysis of the empirical evidence. *Journal of the Academy of Marketing Science*, 29(1):16–35, 2001.

D. K. Tse and P. Wilton. Models of consumer satisfaction formation: An extension. *Journal of Marketing Research*, 25 (May):204–212, 1988.

Y. Yi. A critical review of consumer satisfaction. In V. A. Zeithaml, editor, *Review of Marketing*, pages 68–123. American Marketing Association, Chicago, 1991.