

Learning Decision Rules from Positive and Negative Preferences

Eyke Hüllermeier and Jürgen Beringer

Marburg University, Germany

{eyke,beringer}@informatik.uni-marburg.de

Abstract

The paper first presents a general framework of experience-based decision making, in which hypothetical preferences related to new decision problems are induced from known preferences related to already encountered problems. A particular case of this framework, in which a decision maker distinguishes between acceptable and non-acceptable actions (positive and negative preferences), is then studied in more detail. In order to learn acceptable decision models, a generalization of decision tree induction is proposed. Finally, some experimental results are presented for this learning method.

Keywords: decision making, machine learning, decision trees.

1 Introduction

The classic approach to decision making under uncertainty relies on the paradigm of *expected utility theory* (EUT) [6]. This axiomatic framework has a great theoretical appeal but appears rather restrictive from a practical point of view. Particularly, it assumes the decision maker to have very detailed information at its disposal: A complete list of possible *states* of the environment, a probability distribution over these states, a list of potential acts, and a numerical utility value for all act–state pairs.

A more realistic approach may proceed from the assumption that the behavior of a decision maker (agent) is guided by its *experiences* from previously encountered decision problems. This idea is a main motivation underlying *case-based decision theory* as introduced in [2]. In this approach, experience is represented in the form of *cases* (p, a, u) , where p is a decision problem, a is an action that has been chosen by the agent to solve that problem, and u is the resulting utility. Faced with a new decision problem, the agent estimates the utility of potential actions on the basis of their performance in situations similar to the current one.

More generally, *experience-based decision making* (EBDM) can be seen as the exploitation of experience from past decision problems in order to support future decisions. Needless to say, EBDM thus defined is closely connected with *learning from data*. In particular, case-based decision theory can actually be seen as case-based learning – a well-known technique in the field of machine learning – applied to decision making.

In EBDM, the agent’s information is typically limited and does not allow for deriving provably optimal decisions. In this connection, the concept of “bounded rationality” turns out to be useful. The latter is closely related to H. SIMON’s idea of “satisficing” [7]: Roughly speaking, a satisficing agent tries to choose actions which are satisfactory, but it does not necessarily try to maximize utility. Usually, EBDM-agents are satisficing decision makers.

Apart from limited information, satisficing decision making is motivated by limited computational resources: More often than not the use of a, say, *deliberative* decision model requiring a kind of complete utility analysis will be too expensive. Rather, an agent might be interested in, say, *reactive* or *compiled* decision models that allow for example to make decisions in real-time.

In the following section, we introduce a general framework in which EBDM is realized by *generalizing* the agent’s preferences (over actions) from past decision problems to potential future problems. In Section 3, we consider a special case of this framework in which the agent makes a crude distinction between positive and negative preferences resp. acceptable and non-acceptable actions. We propose a learning method for generating efficient decision rules for this particular setting. Some experimental results for this learning method follow in Section 4.

2 EBDM through Generalizing known Preferences

Let \mathcal{X} denote a set of potential decision problems resp. situations an agent might encounter, and \mathcal{A} a set of actions it can choose from. We assume that the agent’s experience can be represented as a collection (memory) \mathcal{M} of *empirical preferences* of the form

$$a \preceq_x b \quad (1)$$

with $a, b \in \mathcal{A}$ and $x \in \mathcal{X}$. The meaning of a preference (1) is as follows: In the situation x , action b is at least as good as action a . We assume that \preceq_x is transitive (as a relation over \mathcal{A}) for all x , and that \mathcal{M} is transitively closed. Moreover, we define the strict part of \preceq_x as usual, i.e. $a \prec_x b$ if $a \preceq_x b$ and $b \not\preceq_x a$.

A preference (1) may descend from different sources. For example, the agent might have tried both actions, finding that b yields an outcome at least as good as a . Alternatively, a satisficing agent might have tried only b , but since it was satisfied with the result, it prefers b to all other actions not tried so far.

We consider experience-based decision making as generalizing beyond a collection of preferences (1), i.e. as deriving *hypothetical* preferences from empirical ones. To illustrate, suppose that an agent found $a \preceq_x b$ for all situations $x \in \mathcal{M}$ having a certain property P . Relying on a kind of *inductive* reasoning, the agent might then expect that $a \preceq_x b$ will hold for *all* $x \in \mathcal{X}$ having property P .

The perhaps most general learning task related to EBDM could now be stated as follows: Induce, on the basis of the empirical preferences \mathcal{M} , a *preference function* that maps each situation $x \in \mathcal{X}$ to a preference relation over \mathcal{A} .

Subsequently, we shall concentrate on a slightly different problem which is related to the idea of a compiled decision model: Induce a *decision function* (decision model)

$$\Delta : \mathcal{X} \rightarrow \mathcal{A} \quad (2)$$

that prescribes an action $\Delta(x)$ for all decision problems x . Thus, we assume that the agent is first of all interested in a model that prescribes how to act in a situation, i.e. that predicts the “top-action” rather than a complete preference relation.

Let \mathcal{H} denote the class of candidate decision models (the hypothesis space in machine learning). There are two main quality criteria for a decision model $\Delta \in \mathcal{H}$. Firstly, the *decision quality* depends on the extent to which the model is in agreement with the empirical preferences. In fact, a partial order on \mathcal{H} can be derived from the preferences in \mathcal{M} as follows: $\Delta \preceq_Q \Delta'$ if there is no preference $(a \prec_x b) \in \mathcal{M}$ such that $a = \Delta'(x)$ and $b = \Delta(x)$. That is, Δ' is at least as good as Δ if it never prescribes an action that is known to be worse than the one prescribed by Δ .

The second criterion concerns the *efficiency* or *complexity* of a decision model. The aspect of efficiency becomes especially relevant if deliberation time is costly or strictly limited as e.g. in real-time decision making. The efficiency criterion gives rise to a second ordering relation over \mathcal{H} : $\Delta \preceq_E \Delta'$ if Δ' is at least as efficient as Δ .

The two relations \preceq_Q and \preceq_E can be combined into an overall preference relation \preceq over \mathcal{H} :

$$\Delta \preceq \Delta' \Leftrightarrow_{\text{def}} \Delta \preceq_Q \Delta' \text{ and } \Delta \preceq_E \Delta'.$$

Needless to say, \preceq thus defined will usually not identify an optimal decision model in a unique way. Thus, one has to refer to heuristic principles in order to choose a particular model among those that are *non-dominated* (a model Δ is dominated by a model Δ' if $\Delta \prec \Delta'$).

The above setting can be seen as an interesting extension of the problem of *classification*, which is one of the standard problems in machine learning. In this setting, examples are of the form (x, y) , where $y \in \mathcal{Y}$ is the class associated with the instance x . An information of this type can also be represented as follows: $y' \prec_x y$ for all $y' \in \mathcal{Y} \setminus \{y\}$, that is, the correct class for x is preferred to all other classes. Consequently, $h \preceq_Q h'$ for two classification functions $\mathcal{X} \rightarrow \mathcal{Y}$ if all examples that are classified correctly by h are also classified correctly by h' . In this connection, it is interesting to note that in machine learning, classification functions are usually evaluated by their classification accuracy (relative frequency of correctly classified examples). The ordering relation over \mathcal{H} induced by this evaluation (scoring) is of course much more discriminative than the relation \preceq_Q . Moreover, most approaches used for finding a tradeoff between the classification accuracy and the complexity of a model are of a heuristic nature.

3 Learning from Positive and Negative Preferences

The framework presented in the previous section is extremely general. In this section, we consider a special type of experience-based decision making, along with a corresponding learning method for inducing a decision model.

Suppose that, for some decision problems x , the agent is able to divide the set of actions \mathcal{A} into *acceptable* and *non-acceptable* ones.

This distinction complements rather than replaces a preference relation \preceq_x : For two actions a, b it is quite possible that both are acceptable even though b is preferred to a . Still, the following consistency condition should of course be satisfied: $a \prec_x b$ for all actions $b \in A_x, a \in \mathcal{A} \setminus A_x$, where A_x denotes the actions acceptable for x .

The above assumption is related to the idea of a *denoted* utility scale as introduced by YAGER in [9]. A denoted scale is a bipolar scale that consists of a positive part and a negative part. Denoting a scale resp. distinguishing between acceptable and non-acceptable actions is obviously in line with the idea of satisficing: Suppose that a satisficing agent has determined, for each action $a \in \mathcal{A}$, the degree of utility resulting from applying that action to a problem x . Then, referring to a denoted utility scale, all actions with a utility above a certain (satisfying) threshold are declared acceptable. Note, however, that a distinction between acceptable and non-acceptable actions may also descend from other sources. Especially, we do not assume the existence of a utility function in general.

The learning problem can now be stated as follows: Induce an as efficient as possible decision model (2) which is consistent with a memory \mathcal{M} of examples (x_i, A_{x_i}) or, say, acceptable in the sense that it prescribes acceptable actions $a_i \in A_{x_i}$ for the x_i .

Again, this problem can be seen as an interesting generalization of classification in machine learning: For each instance x_i , there is now a set of acceptable decisions (classifications) rather than only one optimal decision. If we define the *acceptance level* by $(e - 1)/(k - 1)$, with $k = |\mathcal{A}|$ and e the *expected* number of acceptable actions for a (randomly selected) problem $x \in \mathcal{X}$, then standard classification is characterized by an acceptance level of 0. As opposed to this, the acceptance level is 1 in the extreme case where all actions are always acceptable.

One important aspect is that acceptable models might be more efficient (less complex) than optimal ones, or, more precisely: The larger

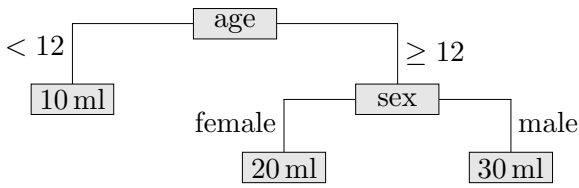


Figure 1: Decision tree implementing a strategy for choosing the dose of a drug.

the acceptance level, the less complex an acceptable model will be. To illustrate, consider the problem of choosing the dose of a drug for different patients. The decision model represented by the simple decision tree in Fig. 1 might lead to acceptable results (the utility of a decision depends on the patient’s state of health after the treatment), whereas optimizing the results might require a more complex model that differentiates more precisely between patients, e.g. by taking other attributes in addition to sex and age into account.

In the remainder of this section, we introduce a method for inducing acceptable decision models in the form of decision trees. Such models are obviously interesting from an efficiency point of view, since a decision tree can be seen as a compiled/reactive decision model consisting of a set of condition–action rules.

3.1 Decision Tree Induction

Here, we briefly recall the basic principles underlying decision tree induction, by now a well-known and widely applied technique in machine learning [5, 1].

The principle idea of decision tree learners is to partition a set of examples, \mathcal{S} , in a recursive manner. Each inner node η of the decision tree defines a partition of a subset $\mathcal{S}_\eta \subseteq \mathcal{S}$ of examples assigned to that node. This is done by classifying elements $x \in \mathcal{S}_\eta$ according to the value of a specific attribute X_i . The attribute is selected according to a measure of effectiveness in classifying the examples, thereby supporting the overall objective of constructing a small tree.

A widely applied “goodness of split” measure is the *information gain*, $G(\mathcal{S}, X_i)$, which is defined as the expected reduction of “impurity” in classifying examples which results from partitioning \mathcal{S} according to X_i :

$$G(\mathcal{S}, X_i) =_{\text{def}} I(\mathcal{S}) - \sum_{x_i} \frac{|\mathcal{S}_{x_i}|}{|\mathcal{S}|} I(\mathcal{S}_{x_i}), \quad (3)$$

where I is a measure of impurity and \mathcal{S}_{x_i} denotes the set of elements $x \in \mathcal{S}$ whose value for attribute X_i is x_i . A standard impurity measure is the entropy

$$E(S) = - \sum_{\alpha \in \mathcal{A}} q_\alpha \cdot \log_2(q_\alpha), \quad (4)$$

where q_α is the proportion of elements $x \in \mathcal{S}$ with associated classification (decision) α ($0 \log_2(0) = 0$ by definition).

The basic learning algorithm derives a decision tree as follows: The complete set of training samples is assigned to the root of the decision tree. A node η becomes a leaf (answer node) of the tree if all examples belonging to that node do have the same class (or if all attributes have already been used along the path from the root of the tree to η). Otherwise, η becomes a decision node: It is split by partitioning the associated set of examples. This is done by selecting an attribute as described above and by partitioning the examples according to the value for that attribute. Each element of the resulting partition defines one successor node.

Once the decision tree has been constructed, each path can be considered as a rule, and new examples can be classified on the basis of these rules. Quite often, the induced tree undergoes further (post-)processing [4].

3.2 Acceptable Decision Trees

As in standard decision tree induction, let decision problems be represented by a fixed set of attributes (features). Thus, $\mathcal{X} = X_1 \times X_2 \times \dots \times X_m$, where X_i denotes the (finite) domain of the i -th attribute; each problem $x \in \mathcal{X}$ is represented as a vector $x = (x_1 \dots x_m)$ of attribute values. Moreover, let $\mathcal{A} = \{\alpha_1 \dots \alpha_k\}$ be a set of potential actions.

The training data now consists of a set \mathcal{M} of examples of the form

$$(x_i, A_{x_i}) \in \mathcal{X} \times 2^{\mathcal{A}},$$

$i = 1 \dots n$, where A_{x_i} is the set of actions acceptable for the problem x_i . The learning task is to induce, on the basis of \mathcal{M} , a decision tree that prescribes acceptable actions, i.e. that implements a decision function $\Delta : \mathcal{X} \rightarrow \mathcal{A}$ such that $\forall x \in \mathcal{M} : \Delta(x) \in A_x$. As can be seen, further splitting of a set of examples \mathcal{M}_η is not necessary if

$$A(\mathcal{M}_\eta) =_{\text{def}} \bigcap_{x_i \in \mathcal{M}_\eta} A_{x_i} \neq \emptyset, \quad (5)$$

hence, (5) defines a natural stopping condition for the recursive partitioning. The corresponding node η in the decision tree then becomes a leaf, and any action $a \in A(\mathcal{M}_\eta)$ can be chosen as the prescribed decision associated with that node.

The main modification of the standard algorithm concerns the “goodness of split” measure. What is the purity, or rather the decision capability of a set of examples \mathcal{M}_η ? If we define q_α as the proportion of elements $x \in \mathcal{M}_\eta$ such that $\alpha \in A_x$, then $q_{\alpha_1} + \dots + q_{\alpha_k} = 1$ does no longer hold. Formally, the entropy measure (4) could still be applied. However, entropy has a meaningful interpretation only for probability distributions. Apart from that, this measure does not appear reasonable in our context: The stopping condition (5) obviously suggests a measure that is monotone increasing in the q_{α_i} , since the larger the q_{α_i} , the higher the chance to satisfy the condition (after few further splits). As opposed to this, the entropy measure does not only reward high values q_{α_i} (close to 1), but also small values close to 0.

Entropy-like measures, just as other measures of uncertainty originally established within the framework of probability, have been generalized to alternative frameworks such as evidence theory and possibility theory [3]. These extensions do not appear to be useful in our context, however, since we do neither need a measure of non-specificity nor a measure of conflict.

The perhaps simplest monotone measure¹ is

$$\max(q_{\alpha_1} \dots q_{\alpha_k}), \quad (6)$$

and this measure is indeed interesting since satisfaction of condition (5) is equivalent to $\max(q_{\alpha_1} \dots q_{\alpha_k}) = 1$. A drawback of the measure is of course that it is completely determined by only one of the q_{α_i} . As a generalization, we therefore considered the following OWA operator [8]:

$$O_\gamma =_{\text{def}} \frac{\sum_{i=1}^k \gamma^{k-1} r_i}{(1-\gamma)^k / (1-\gamma)}, \quad (7)$$

where $0 < \gamma \leq 1$ and r_i is the i -th largest value in the sequence $q_{\alpha_1} \dots q_{\alpha_k}$; (6) is obviously recovered for $\gamma = 0$.

As an alternative to (7) we considered a measure of “potential entropy” which is defined as follows: Let the class of *selections*, $\mathcal{F}(\mathcal{M})$, of a set of examples \mathcal{M} be given by the class of (standard) samples

$$\{(x_1, a_1), (x_2, a_2), \dots, (x_n, a_n)\} \subseteq \mathcal{X} \times \mathcal{A}$$

such that $a_i \in A_{x_i}$ for all $1 \leq i \leq n$. We define the potential entropy by

$$E^*(\mathcal{M}) =_{\text{def}} \min_{\mathcal{S} \in \mathcal{F}(\mathcal{M})} E(\mathcal{S}). \quad (8)$$

As can be seen, (8) is the standard entropy obtained for the most favorable instantiation of the (generalized) examples (x_i, A_{x_i}) . It corresponds to the “true” entropy that would have been derived if this instantiation was compatible with the ultimate decision tree. Taking this optimistic attitude is clearly justified since the tree is indeed constructed in a hopefully optimal manner.

Computing (8) comes down to solving a combinatorial optimization problem and becomes intractable for large samples. Therefore, we devised two heuristic approximations of (8). The first approximation is

$$E_1^*(\mathcal{M}) =_{\text{def}} E(\mathcal{S}^*), \quad (9)$$

where the selection \mathcal{S}^* is defined as follows: The actions α_i are “preferentially ordered”

¹This is of course a kind of purity rather than impurity measure, hence signs have to be reversed in (3).

according to their relative frequency q_{α_i} (ties are broken by coin flipping), starting with the most frequent one. Then, the most preferred action $a_i \in A_{x_i}$ is chosen for each example x_i .

The second approximation is defined as follows: Without loss of generality, suppose $q_{\alpha_1} \geq q_{\alpha_2} \geq \dots \geq q_{\alpha_k}$. Then,

$$E_2^*(\mathcal{M}) =_{\text{def}} - \sum_{i=1}^k q'_{\alpha_i} \log_2(q'_{\alpha_i}), \quad (10)$$

where $q'_{\alpha_i} = q_{\alpha_i}$ if $q_{\alpha_1} + \dots + q_{\alpha_i} \leq 1$ and $= \max(1 - (q_{\alpha_1} + \dots + q_{\alpha_{i-1}}), 0)$ otherwise. Note that (10) may underestimate the potential entropy (there might be no selection that agrees with the modified frequencies q'_{α_i}), whereas (9) is provably an upper bound.

3.3 Neutral Preferences

So far, a distinction has been made only between acceptable and non-acceptable actions resp. positive and negative preferences. In practice, however, a decision maker will often know the effect (utility) of applying an action in a situation x only for some of the alternatives $\alpha \in \mathcal{A}$. At least for risk-averse agents it is then natural to prefer actions that are known to be acceptable to actions whose effect is *unknown* and, moreover, the latter to actions known to be non-acceptable. Thus, the agent has *neutral* preferences that fall in-between the positive and negative ones.

In order to take neutral preferences into consideration, the above approach can be extended or, say, embedded into an optimization (search) process on a meta-level. The basic idea is to assign the neutral actions, for each example x_i , either to the acceptable or the non-acceptable actions. Thus, each A_{x_i} is either defined by the set of acceptable actions (option $o_i = 0$) or by those that are at least neutral ($o_i = 1$). Of course, for $o_i = 0$ the quality of the induced model will be better, while for $o_i = 1$ the model will become simpler. Again, a compromise between decision quality and complexity has to be found.

In principle, a model that achieves an optimal tradeoff between quality and complexity could be found by inducing a model for

each of the 2^n possible options $o = (o_1 \dots o_n)$. Needless to say, this becomes infeasible for large n . Thus, heuristic search methods have to be used instead of complete enumeration. For example, starting with the model for $o = (0 \dots 0)$, a greedy strategy could turn one o_i from 0 to 1 in each step. The corresponding i is chosen by testing all potential candidates and selecting the best one (one-step look-ahead search). The process stops if the new model does not improve the current one or if $o_i = 1$ for all $1 \leq i \leq n$.

4 Experimental Results

This section presents some experimental results. Since benchmark data sets are not as yet available for the type of learning problem introduced above, our experiments are based on synthetic data.

The main purpose of our first study was to evaluate and compare the performance of the different splitting measures introduced in the previous section. The purpose of a second study was to investigate the relationship between the acceptance level and the complexity of induced models.

4.1 Experimental Study I

Synthetic Data. In our first study, synthetic data is derived on the basis of randomly generated (acceptable) decision trees that serve as reference models. For each experiment, a decision tree of this kind is generated as follows: In a first step, a standard tree is grown in a recursive manner, starting with the root of the tree and flipping a (biased) coin to decide whether the current node becomes an inner node or a leaf. The probability of a node to become an inner node is given by $0.8 - t/10$, where t is the depth of the node. Once a leaf node has been generated, a decision $\alpha \in \{\alpha_1 \dots \alpha_k\}$ is assigned to that node at random. Likewise, each inner node is assigned one among m possible attributes at random. We let each attribute X_j assume the v values $x_j \in \{1 \dots v\}$, $v \in \mathbb{N}$, so that an inner node does always have v successors.

In a second step, further (acceptable) decisions are assigned to the leaf nodes of the tree. More precisely, at every leaf node, the actions are added independently of each other with a fixed probability p (hence the acceptance level is p). Finally, it is checked whether the expanded tree thus obtained can be simplified, i.e. whether one of the inner nodes satisfies the stopping condition (5) and should hence become a leaf.

A set of training examples is then generated by choosing 1000 instances x at random, using a uniform distribution over \mathcal{X} . The labels A_x of these examples are derived from the above reference tree. For the training examples, a decision tree is induced using one of the splitting measures discussed in the previous section. The performance of a measure is quantified in terms of the ratio between the complexity of the induced tree (number of leaf nodes) and the complexity of the reference tree. (Note that this ratio can thoroughly assume values < 1 , since the training examples might be explained by a model which is simpler than the model that has generated the data.) The expected performance of each splitting measure is approximated by averaging over 2000 experiments.

Results. Recall that an experimental study of the above type is characterized by the following parameters: m = number of attributes, k = number of decisions, v = number of values per attribute, p = probability of adding a decision.

The tables 1–2 show the experimental results for different settings of the parameters m, k, v (qualitatively similar results were obtained for various other settings, not presented here due to reasons of space). More precisely, each table shows the results for the different splitting measures discussed above and different acceptance levels p . For comparison purpose, and despite of the aforementioned semantic problems, we also included the standard entropy formula (4) applied to the values q_{α_i} .

The results clearly show that the measure E_1^* consistently outperforms the other measures. Moreover, this measure yields models that are

p	0.00	0.20	0.40	0.60	0.80
E_1^*	0.94	0.97	0.99	0.89	0.64
E_2^*	0.94	1.01	1.05	0.91	0.63
E	0.94	1.24	1.40	1.27	0.93
O_0	1.09	1.10	1.07	0.94	0.66
$O_{0.2}$	1.04	1.09	1.08	0.96	0.66
$O_{0.4}$	1.01	1.06	1.08	0.97	0.67
$O_{0.6}$	0.98	1.07	1.11	1.01	0.71
$O_{0.8}$	0.97	1.12	1.20	1.11	0.80

Table 1: Setting $m = 12, k = 10, v = 4$.

p	0.00	0.20	0.40	0.60	0.80
E_1^*	1.01	1.03	1.12	1.22	1.12
E_2^*	1.01	1.12	1.43	1.32	1.13
E	1.01	1.30	1.78	1.91	1.62
O_0	1.49	1.45	1.45	1.38	1.19
$O_{0.2}$	1.31	1.30	1.35	1.32	1.16
$O_{0.4}$	1.20	1.21	1.29	1.30	1.17
$O_{0.6}$	1.12	1.15	1.27	1.35	1.23
$O_{0.8}$	1.08	1.18	1.43	1.59	1.43

Table 2: Setting $m = 12, k = 15, v = 3$.

only slightly more or even less complex than the reference models. This suggests that E_1^* is indeed a very good measure, in fact not only in comparison to the alternatives but also in view of “absolute” standards.

As it was to be expected, the standard entropy E yields good results for p close to 0 but quickly breaks down as p increases. The results of the best OWA-measures are somewhat comparable to those of E_2^* . As can be seen, however, the performance strongly depends on the parameter γ which has to be adjusted to the parameter p : The larger p , the smaller γ should be chosen.

4.2 Experimental Study II

Synthetic Data. In the second study, data was generated as follows: For each experiment, 100 instances x_i were selected at random from $\mathcal{X} = \{1 \dots v\}^m$ (the meaning of parameters is the same as before). Moreover, a random utility degree, uniformly distributed in $[0, 1]$, was assigned to each problem–action pair (x_i, α_j) , $i = 1 \dots n, j = 1 \dots k$. Finally, an action α_j was declared acceptable

(for x_i) if it is not much worse than the optimal action or, more precisely, if $u(x_i, \alpha_j) \geq c \cdot \max_{\ell=1\dots k} u(x_i, \alpha_\ell)$. Here, c is a constant that can be considered as a measure of the “aspiration” of a satisficing agent: The smaller c , the larger the acceptance level and the less ambitious the agent.

Results. Decision trees were induced from the data using the measure E_1^* . For $m = 10, v = 3, k = 4$, the following table shows the complexity K (number of leaf nodes) of the trees as a function of the parameter c , averaged over 100 experiments:

c	1.0	0.8	0.6	0.4	0.2
K	72.59	55.67	39.55	24.92	11.67
	± 0.36	± 0.40	± 0.42	± 0.34	± 0.21

Interestingly, the relationship between c and K is almost perfectly linear. In any case, the aforementioned effect – a gain of simplicity at the cost of decision quality – becomes very obvious. Again, similar results were obtained for alternative settings.

5 Concluding Remarks

We have introduced a framework of experience-based decision making in which a close connection is established between decision making and machine learning. In particular, we have considered a setting in which a (satisficing) decision maker can express positive and negative (any maybe neutral) preferences. Learning a decision model from experience was then formalized as a kind of generalized classification problem. We have extended standard decision tree induction for this purpose. The basic modification of the algorithm concerns the measure used for selecting splitting attributes at the inner nodes of the tree.

Different splitting measures have been proposed and evaluated by means of experimental studies. These studies have shown that a heuristic approximation of the “potential entropy measure” yields very good results and consistently outperforms the alternative measures.

As concerns the learning method for inducing acceptable decision trees, our main concern in this paper was the construction of simple models on the basis of the given data. A further point that needs to be investigated is of course the *generalization performance* of the learning method, i.e., the quality of predicted preferences for new situations $x \notin \mathcal{M}$. Apart from that, other machine learning methods might be useful for the problem considered in this paper. An obvious alternative to (divide-and-conquer) decision tree induction, for instance, is the use of general (separate-and-conquer) rule learning methods.

References

- [1] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA, 1984.
- [2] I. Gilboa and D. Schmeidler. Case-based decision theory. *Quarterly Journal of Economics*, 110(4):605–639, 1995.
- [3] G.J. Klir and M.J. Wierman. *Uncertainty-Based Information*. Physica-Verlag, Heidelberg, 1998.
- [4] J. Mingers. An empirical comparison of pruning methods for decision tree induction. *Machine Learning*, 4:227–243, 1989.
- [5] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.
- [6] L.J. Savage. *The Foundations of Statistics*. John Wiley and Sons, Inc., New York, 1954.
- [7] H.A. Simon. *Models of Man*. John Wiley and Sons, New York, 1957.
- [8] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):183–190, 1988.
- [9] R.R. Yager. Using a notion of acceptable in uncertain ordinal decision making. *Int. J. Uncertainty, Fuzziness, and Knowledge-Based Systems*, 10(3), 2002.