Mach Learn (2012) 89:183–211
DOI 10.1007/s10994-012-5318-3

Learning monotone nonlinear models using the Choquet integral

Ali Fallah Tehrani · Weiwei Cheng · Krzysztof Dembczyński · Eyke Hüllermeier

Received: 7 November 2011 / Revised: 17 July 2012 / Accepted: 19 July 2012 /
Published online: 10 August 2012

Abstract The learning of predictive models that guarantee monotonicity in the input variables has received increasing attention in machine learning in recent years. By trend, the difficulty of ensuring monotonicity increases with the flexibility or, say, nonlinearity of a model. In this paper, we advocate the so-called Choquet integral as a tool for learning monotone nonlinear models. While being widely used as a flexible aggregation operator in different fields, such as multiple criteria decision making, the Choquet integral is much less known in machine learning so far. Apart from combining monotonicity and flexibility in a mathematically sound and elegant manner, the Choquet integral has additional features making it attractive from a machine learning point of view. Notably, it offers measures for quantifying the importance of individual predictor variables and the interaction between groups of variables. Analyzing the Choquet integral from a classification perspective, we provide upper and lower bounds on its VC-dimension. Moreover, as a methodological contribution, we propose a generalization of logistic regression. The basic idea of our approach, referred to as choquistic regression, is to replace the linear function of predictor variables, which is commonly used in logistic regression to model the log odds of the positive class, by the Choquet integral. First experimental results are quite promising and suggest that the combination of monotonicity and flexibility offered by the Choquet integral facilitates strong performance in practical applications.

Editors: Dimitrios Gunopulos, Donato Malerba, and Michalis Vazirgiannis.

A. Fallah Tehrani · W. Cheng (✉) · E. Hüllermeier
Department of Mathematics and Computer Science, Marburg University, Marburg, Germany
e-mail: cheng@mathematik.uni-marburg.de

A. Fallah Tehrani
e-mail: fallah@mathematik.uni-marburg.de

E. Hüllermeier
e-mail: eyke@mathematik.uni-marburg.de

K. Dembczyński
Institute of Computing Science, Poznań University of Technology, Poznań, Poland
e-mail: krzysztof.dembczynski@cs.put.poznan.pl

Keywords Choquet integral · Monotone learning · Nonlinear models · Choquistic regression · Classification · VC dimension

1 Introduction

A proper specification of the type of dependency between a set of predictor (input) variables $X_1, \ldots, X_m$ and the target (output) variable Y is an important prerequisite for successful model induction. The specification of a corresponding hypothesis space imposes an inductive bias that, amongst others, allows for the incorporation of background knowledge in the learning process. An important type of background knowledge is *monotonicity*: Everything else being equal, the increase (decrease) of a certain input variable X_i can only produce an increase in the output variable Y (e.g., a real number in regression, a class in ordered classification, or the probability of the positive class in binary classification). Adherence to this kind of background knowledge may not only be beneficial for model induction, but is often even considered as a hard constraint. For example, no medical doctor will accept a model in which the probability of cancer is *not* monotonically increasing in tobacco consumption.

The simplest type of dependency is a linear relationship:

$$Y = \sum_{i=1}^{m} \alpha_i X_i + \epsilon, \tag{1}$$

where $\alpha_1, \ldots, \alpha_m$ are real coefficients and ϵ is an error term. Monotonicity can be guaranteed quite easily for (1), since monotonicity in X_i is equivalent to the constraint $\alpha_i \geq 0$. Another important advantage of (1) is its comprehensibility. In particular, the direction and strength of influence of each predictor X_i are directly reflected by the corresponding coefficient α_i.

Perhaps the sole disadvantage of a linear model is its inflexibility and, coming along with this, the supposed absence of any *interaction* between the variables: The effect of an increase of X_i is always the same, namely $\partial Y / \partial X_i = \alpha_i$, regardless of the values of all other attributes. In many real applications, this assumption is not tenable. Instead, more complex, nonlinear models are needed to properly capture the dependencies between the inputs X_i and the output Y.

An increased flexibility, however, typically comes at the price of a loss in terms of the two previous criteria: comprehensibility is hampered, and monotonicity is more difficult to assure. In fact, as soon as an interaction between attributes is allowed, the influence of an increase in X_i may depend on all other variables, too. As a simple example, consider the extension of (1) by the addition of *interaction terms*, a model which is often used in statistics:

$$Y = \sum_{i=1}^{m} \alpha_i X_i + \sum_{1 \leq i < j \leq m} \alpha_{ij} X_i X_j + \epsilon. \tag{2}$$

For this model, $\partial Y / \partial X_i$ is given by $\alpha_i + \sum_{j \neq i} \alpha_{ij} X_j$ and depends on the values of *all* other attributes, which means that, depending on the context as specified by these values, the monotonicity condition may change from one case to another. Consequently, it is difficult to find simple *global* constraints on the coefficients that assure monotonicity. For example, assuming that all attributes are non-negative, it is clear that $\alpha_i \geq 0$ and $\alpha_{ij} \geq 0$ for all $1 \leq i \leq j \leq m$ will imply monotonicity. While being sufficient, however, these constraints are non-necessary conditions, and may therefore impose restrictions on the model space that are

more far-ranging than desired; besides, negative interactions cannot be modeled in this way. Quite similar problems occur for commonly used nonlinear methods in machine learning, such as neural networks and kernel machines.

In this paper, we advocate the use of the (discrete) Choquet integral as a tool that is interesting in this regard. As will be argued in more detail later on, the Choquet integral combines the aforementioned properties in a quite convenient and mathematically elegant way: By its very nature as an integral, it is a monotone operator, while at the same time allowing for interactions between attributes. Moreover, it disposes of natural measures for quantifying the *importance* of individual and the *interaction* within groups of features, which provide important insights into the model and thereby support interpretability.

The rest of this paper, parts of which have already been presented in Fallah Tehrani et al. (2011), Hüllermeier and Fallah Tehrani (2012b), is organized as follows. In the next section, we give a brief overview of related work. In Sect. 3, we recall the basic definition of the Choquet integral and some related notions. In Sect. 4, we analyze the flexibility of binary classifiers based on the Choquet integral in terms of the notion of VC dimension. In Sect. 5, we propose a generalization of logistic regression for binary classification, in which the Choquet integral is used to model the log odds of the positive class. In Sect. 6, we elaborate on complexity issues and propose a method for finding a suitable level of (non-)additivity for the Choquet integral in a concrete learning task. Experimental results are presented in Sect. 7, prior to concluding the paper with a few remarks in Sect. 8.

2 Related work

As already mentioned, the problem of monotone classification has received increasing attention in the machine learning community in recent years,[1] despite having been introduced in the literature much earlier (Ben-David et al. 1989). Meanwhile, several machine learning algorithms have been modified so as to guarantee monotonicity in attributes, including nearest neighbor classification (Duivesteijn and Feelders 2008), neural networks (Sill 1998), decision tree learning (Ben-David 1995; Potharst and Feelders 2002), rule induction (Dembczyński et al. 2009), as well as methods based on isotonic separation (Chandrasekaran et al. 2005) and piecewise linear models (Dembczyński et al. 2006).

Instead of modifying learning algorithms so as to guarantee monotone models, another idea is to modify the training data. To this end, data pre-processing methods such as relabeling techniques have been developed. Such methods seek to repair inconsistencies in the training data, so that (standard) classifiers learned on that data will tend to be monotone (although, in general, they still do not guarantee this property) (Feelders 2010; Kotłowski et al. 2008).

Although the Choquet integral has been widely applied as an aggregation operator in multiple criteria decision making (Grabisch et al. 2000; Grabisch 1995a; Torra 2011), it has been used much less in the field of machine learning so far. There are, however, a few notable exceptions. First, the problem of extracting a Choquet integral (or, more precisely, the non-additive measure on which it is defined) in a data-driven way has been addressed in the literature (Beliakov 2008). Essentially, this is a parameter identification problem, which is commonly formalized as a constraint optimization problem, for example using the sum of squared errors as an objective function (Torra and Narukawa 2007; Grabisch 2003). To

[1] For example, a workshop on "Learning Monotone Models from Data" was organized at ECMLPKDD 2009 in Bled, Slovenia.

this end, Mori and Murofushi (1989) proposed an approach based on the use of quadratic forms, while an alternative heuristic, gradient-based method called HLMS (Heuristic Least Mean Squares) was introduced in Grabisch (1995b). In Angilella et al. (2009), Beliakov and James (2011), the Choquet integral is used in the context of ordinal classification. Besides, the Choquet integral has been used as an aggregation operator in the context of ensemble learning, i.e., for combining the predictions of different classifiers (Grabisch and Nicolas 1994).

3 The discrete Choquet integral

In this section, we give a brief introduction to the (discrete) Choquet integral, which, to the best of our knowledge, is not widely known in the field of machine learning so far. Since the Choquet integral can be seen as a generalization of the standard (Lebesque) integral to the case of non-additive measures, we start with a reminder of this type of measure.

3.1 Non-additive measures

Let $C = \{c_1, \ldots, c_m\}$ be a finite set and $\mu : 2^C \to [0, 1]$ a measure on this set. For each $A \subseteq C$, we interpret $\mu(A)$ as the *weight* or, say, the *importance* of the set of elements A. As an illustration, one may think of C as a set of criteria (binary features) relevant for a job, like "speaking French" and "programming Java", and of $\mu(A)$ as the evaluation of a candidate satisfying criteria A (and not satisfying $C \setminus A$). The term "criterion" is indeed often used in the decision making literature, where it suggests a monotone "the higher the better" influence. In the context of machine learning, to which we shall turn later on, criteria are playing the role of features (input attributes).

A standard assumption on a measure $\mu(\cdot)$, which is, for example, at the core of probability theory, is additivity: $\mu(A \cup B) = \mu(A) + \mu(B)$ for all $A, B \subseteq C$ such that $A \cap B = \emptyset$. Unfortunately, additive measures cannot model any kind of interaction between elements: Extending a set of elements A by a set of elements B always increases the weight $\mu(A)$ by the weight $\mu(B)$, regardless of A and B.

Suppose, for example, that the elements of two sets A and B are *complementary* in a certain sense. For instance, $A = \{\texttt{French}, \texttt{Spanish}\}$ and $B = \{\texttt{Java}\}$ could be seen as complementary, since both language skills and programming skills are important for the job. Formally, this can be expressed in terms of a positive interaction: $\mu(A \cup B) > \mu(A) + \mu(B)$. In the extreme case, when language skills and programming skills are indeed essential, $\mu(A \cup B)$ can be high although $\mu(A) = \mu(B) = 0$ (suggesting that a candidate lacking either language or programming skills is completely unacceptable). Likewise, elements can interact in a negative way: If two sets A and B are partly *redundant* or *competitive*, then $\mu(A \cup B) < \mu(A) + \mu(B)$. For example, $A = \{\texttt{C}, \texttt{C\#}\}$ and $B = \{\texttt{Java}\}$ might be seen as redundant, since one programming language does in principle suffice.

The above considerations motivate the use of non-additive measures, also called capacities or fuzzy measures, which are simply normalized and monotone (Sugeno 1974):

$$\begin{aligned} &\mu(\emptyset) = 0, \qquad \mu(C) = 1, \quad \text{and} \\ &\mu(A) \leq \mu(B) \quad \text{for all } A \subseteq B \subseteq C. \end{aligned} \tag{3}$$

A useful representation of non-additive measures, that we shall explore later on for learning Choquet integrals, is in terms of the *Möbius transform*:

$$\mu(B) = \sum_{A \subseteq B} \boldsymbol{m}_\mu(A) \tag{4}$$

for all $B \subseteq C$, where the Möbius transform $\boldsymbol{m}_\mu$ of the measure μ is defined as follows:

$$\boldsymbol{m}_\mu(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|} \mu(B). \tag{5}$$

The value $\boldsymbol{m}_\mu(A)$ can be interpreted as the weight that is *exclusively* allocated to A, instead of being indirectly connected with A through the interaction with other subsets.

A measure μ is said to be k-order additive, or simply k-additive, if k is the smallest integer such that $\boldsymbol{m}(A) = 0$ for all $A \subseteq C$ with $|A| > k$. This property is interesting for several reasons. First, as can be seen from (4), it means that a measure μ can formally be specified by significantly fewer than 2^m values, which are needed in the general case. Second, k-additivity is also interesting from a semantic point of view: As will become clear in the following, this property simply means that there are no interaction effects between subsets $A, B \subseteq C$ whose cardinality exceeds k.

3.2 Importance of criteria and interaction

An additive (i.e., k-additive with $k = 1$) measure μ can be written as follows:

$$\mu(A) = \sum_{c_i \in A} w_i,$$

with $w_i = \mu(\{c_i\})$ the weight of c_i. Due to (3), these weights are non-negative and such that $\sum_{i=1}^m w_i = 1$. In this case, there is obviously no interaction between the criteria c_i, i.e., the influence of a c_i on the value of μ is independent of the presence or absence of any other c_j. Besides, the weight w_i is a natural quantification of the *importance* of c_i.

Measuring the importance of a criterion c_i becomes obviously more involved when μ is non-additive. Besides, one may then also be interested in a measure of *interaction* between the criteria, either pairwise or even of a higher order. In the literature, measures of that kind have been proposed, both for the importance of single as well as the interaction between several criteria.

Suppose to be given a fuzzy measure μ on C. In order to quantify the weight of a single criterion c_i, it is natural to look at the increase in importance due to adding c_i to another subset $A \subset C$, which comes down to comparing $\mu(A \cup \{c_i\})$ and $\mu(A)$. While the difference between these two values is always equal to w_i in the additive case, it may depend on the subset A in the non-additive case. The *Shapley value*, also called importance index of c_i, therefore averages the difference $\mu(A \cup \{c_i\}) - \mu(A)$ over all $A \subset C$:

$$\varphi(c_i) = \sum_{A \subseteq C \setminus \{c_i\}} \frac{1}{m\binom{m-1}{|A|}} \big(\mu\big(A \cup \{c_i\}\big) - \mu(A)\big). \tag{6}$$

The Shapley value of μ is the vector $\boldsymbol{\varphi}(\mu) = (\varphi(c_1), \ldots, \varphi(c_m))$. One can show that $0 \leq \varphi(c_i) \leq 1$ and $\sum_{i=1}^m \varphi(c_i) = 1$. Thus, $\varphi(c_i)$ is a measure of the *relative* importance of c_i. Obviously, $\varphi(c_i) = \mu(\{c_i\})$ if μ is additive.

The *interaction index* between criteria c_i and c_j, as proposed by Murofushi and Soneda (1993), is defined as follows:

$$I_{i,j} = \sum_{A \subseteq C \setminus \{c_i, c_j\}} \frac{(\mu(A \cup \{c_i, c_j\}) - \mu(A \cup \{c_i\}) - \mu(A \cup \{c_j\}) + \mu(A))}{(m-1)\binom{m-2}{|A|}}.$$

This index ranges between -1 and 1 and indicates a positive (negative) interaction between criteria c_i and c_j if $I_{i,j} > 0$ ($I_{i,j} < 0$). The interaction index can also be expressed in terms of the Möbius transform:

$$I_{i,j} = \sum_{K \subseteq C \setminus \{c_i, c_j\}} \frac{1}{|K|+1} \boldsymbol{m}\big(\{c_i, c_j\} \cup K\big).$$

Furthermore, as proposed by Grabisch (1997), the definition of interaction can be extended to more than two criteria, i.e., to subsets $T \subseteq C$:

$$I_T = \sum_{k=0}^{m-|T|} \frac{1}{k+1} \sum_{K \subseteq C \setminus T, |K|=k} \boldsymbol{m}(T \cup K).$$

3.3 The Choquet integral

So far, the criteria c_i were simply considered as binary features, which are either present or absent. Mathematically, $\mu(A)$ can thus also be seen as an *integral* of the indicator function of A, namely the function f_A given by $f_A(c) = 1$ if $c \in A$ and $= 0$ otherwise. Now, suppose that $f : C \to \mathbb{R}_+$ is any non-negative function that assigns a *value* to each criterion c_i; for example, $f(c_i)$ might be the degree to which a candidate satisfies criterion c_i. An important question, then, is how to *aggregate* the evaluations of individual criteria, i.e., the values $f(c_i)$, into an overall evaluation, in which the criteria are properly weighted according to the measure μ. Mathematically, this overall evaluation can be considered as an integral $\mathcal{C}_\mu(f)$ of the function f with respect to the measure μ.

Indeed, if μ is an additive measure, the standard integral just corresponds to the *weighted mean*

$$\mathcal{C}_\mu(f) = \sum_{i=1}^{m} w_i \cdot f(c_i) = \sum_{i=1}^{m} \mu\big(\{c_i\}\big) \cdot f(c_i), \tag{7}$$

which is a natural aggregation operator in this case. A non-trivial question, however, is how to generalize (7) in the case where μ is non-additive.

This question, namely how to define the integral of a function with respect to a non-additive measure (not necessarily restricted to the discrete case), is answered in a satisfactory way by the Choquet integral, which has first been proposed for additive measures by Vitali (1925) and later on for non-additive measures by Choquet (1954). The point of departure of the Choquet integral is an alternative representation of the "area" under the function f, which, in the additive case, is a natural interpretation of the integral. Roughly speaking, this representation decomposes the area in a "horizontal" instead of a "vertical" manner, thereby making it amenable to a straightforward extension to the non-additive case. More specifically, note that the weighted mean can be expressed as follows:

$$\sum_{i=1}^{m} f(c_i) \cdot \mu\big(\{c_i\}\big) = \sum_{i=1}^{m} \big(f(c_{(i)}) - f(c_{(i-1)})\big)\big(\mu\big(\{c_{(i)}\}\big) + \cdots + \mu\big(\{c_{(m)}\}\big)\big)$$

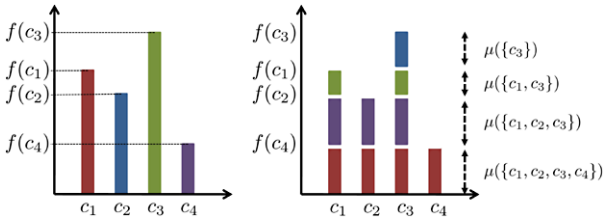

Fig. 1 Vertical (*left*) versus horizontal (*right*) integration. In the first case, the height of a single bar, $f(c_i)$, is multiplied with its "width" (the weight $\mu(\{c_i\})$), and these products are added. In the second case, the height of each horizontal section, $f(c_{(i)}) - f(c_{(i-1)})$, is multiplied with the corresponding "width" $\mu(A_{(i)})$

$$= \sum_{i=1}^{m} \big(f(c_{(i)}) - f(c_{(i-1)})\big) \cdot \mu(A_{(i)}),$$

where $(\cdot)$ is a permutation of $\{1, \ldots, m\}$ such that $0 \leq f(c_{(1)}) \leq f(c_{(2)}) \leq \cdots \leq f(c_{(m)})$ (and $f(c_{(0)}) = 0$ by definition), and $A_{(i)} = \{c_{(i)}, \ldots, c_{(m)}\}$; see Fig. 1 as an illustration.

Now, the key difference between the left and right-hand side of the above expression is that, whereas the measure μ is only evaluated on single elements c_i on the left, it is evaluated on *subsets* of elements on the right. Thus, the right-hand side suggests an immediate extension to the case of non-additive measures, namely the Choquet integral, which, in the discrete case, is formally defined as follows:

$$\mathcal{C}_\mu(f) = \sum_{i=1}^{m} \big(f(c_{(i)}) - f(c_{(i-1)})\big) \cdot \mu(A_{(i)}).$$

In terms of the Möbius transform of μ, the Choquet integral can also be expressed as follows:

$$\begin{aligned}
\mathcal{C}_\mu(f) &= \sum_{i=1}^{m} \big(f(c_{(i)}) - f(c_{(i-1)})\big) \cdot \mu(A_{(i)}) \\
&= \sum_{i=1}^{m} f(c_{(i)}) \cdot \big(\mu(A_{(i)}) - \mu(A_{(i+1)})\big) \\
&= \sum_{i=1}^{m} f(c_{(i)}) \sum_{R \subseteq T_{(i)}} \boldsymbol{m}(R) \\
&= \sum_{T \subseteq C} \boldsymbol{m}(T) \times \min_{i \in T} f(c_i) \qquad (8)
\end{aligned}$$

where $T_{(i)} = \{S \cup \{c_{(i)}\} \mid S \subset \{c_{(i+1)}, \ldots, c_{(m)}\}\}$.

4 The VC dimension of the Choquet integral

Advocating the Choquet integral as a novel tool for machine learning immediately begs an interesting theoretical question, namely the question regarding the capacity of the corresponding model class. In fact, since the Choquet integral in its general form (not restricted to k-additive measures) has a rather large number of parameters, one may expect it to be quite

flexible and, therefore, to have a high capacity. On the other hand, the parameters cannot be chosen freely. Instead, they are highly constrained due to the properties of the underlying fuzzy measure.

In any case, knowledge about the VC dimension of the Choquet integral (or, more specifically, a binary classifier based on the Choquet integral as an underlying aggregation function) is not only of theoretical but also of practical relevance. In particular, it may help finding the right level of flexibility for the data at hand. As mentioned earlier, because of its highly nonlinear nature, one may expect the Choquet integral in its most general form comes with a danger of overfitting the data. On the other hand, a restriction to k-additive measures may provide a reasonable means for regularization. Both conjectures will be confirmed in this section.

In what follows, we are going to analyze the capacity of the Choquet integral in terms of the VC dimension (Vapnik 1998). To this end, we consider a setting in which the Choquet integral is used to classify instances represented in the form of m-dimensional vectors $\boldsymbol{x} = (x_1, x_2, \ldots, x_m) \in \mathbb{R}_+^m$, where $x_i = f(c_i)$ can be thought of as the evaluation of the criterion c_i. More specifically, we consider the model class $\mathcal{H}$ consisting of all threshold classifiers of the form

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_m) \mapsto \mathbb{I}\big(\mathcal{C}_\mu(\boldsymbol{x}) > \beta\big), \tag{9}$$

where $\mathbb{I}$ maps truth degrees $\{\text{false}, \text{true}\}$ to $\{0, 1\}$ as usual, μ is a fuzzy measure, $\mathcal{C}_\mu(\boldsymbol{x})$ is the Choquet integral of the (normalized) attribute values $x_1, x_2, \ldots, x_m$, and $\beta \in [0, 1]$ is a threshold value (as will be seen below, (9) corresponds to the "decision making" part of the choquistic regression model to be introduced in the next section; since this part is responsible for the classification decision, results on the VC dimension of $\mathcal{H}$ directly apply to choquistic regression, too). Note that the class $\mathcal{H}$ is parametrized by μ and β.

Theorem 1 *For the model class $\mathcal{H}$ as defined above, $VC(\mathcal{H}) = \Omega(2^m/\sqrt{m})$. That is, the VC dimension of $\mathcal{H}$ grows asymptotically at least as fast as $2^m/\sqrt{m}$.*

Proof In order to prove this claim, we construct a sufficiently large data set $\mathcal{D}$ and show that, despite its size, it can be shattered by $\mathcal{H}$. In this construction, we restrict ourselves to binary attribute values, which means that $x_i \in \{0, 1\}$ for all $1 \leq i \leq m$. Consequently, each instance $\boldsymbol{x} = (x_1, \ldots, x_m) \in \{0, 1\}^m$ can be identified with a subset of indices $S_{\boldsymbol{x}} \subseteq X = \{1, 2, \ldots, m\}$, namely its *indicator set* $S_{\boldsymbol{x}} = \{i \mid x_i = 1\}$.

In combinatorics, an *antichain* of $X = \{1, 2, \ldots, m\}$ is a family of subsets $\mathcal{A} \subset 2^X$ such that, for all $A, B \in \mathcal{A}$, neither $A \subseteq B$ nor $B \subseteq A$. An interesting question related to the notion of an antichain concerns its potential size, that is, the number of subsets in $\mathcal{A}$. This number is obviously restricted due to the above non-inclusion constraint on pairs of subsets. An answer to this question is given by a well-known result of Sperner (1928), who showed that this number is

$$\binom{m}{\lfloor m/2 \rfloor}. \tag{10}$$

Moreover, Sperner has shown that the corresponding antichain $\mathcal{A}$ is given by the family of all q-subsets of X with $q = \lfloor m/2 \rfloor$, that is, all subsets $A \subset X$ such that $|A| = q$.

Now, we define the data set $\mathcal{D}$ in terms of the collection of all instances $\boldsymbol{x} = (x_1, \ldots, x_m) \in \{0, 1\}^m$ whose indicator set $S_{\boldsymbol{x}}$ is a q-subset of X. Recall that, from a decision making perspective, each attribute can be interpreted as a criterion. Thus, each instance in our data set satisfies exactly q of the m criteria, and there is not a single "dominance"

relation in the sense that the set of criteria satisfied by one instance is a superset of those satisfied by another instance. Intuitively, the instances in $\mathcal{D}$ are therefore maximally incomparable. This is precisely the property we are now going to exploit in order to show that $\mathcal{D}$ can be shattered by $\mathcal{H}$.

Recall that a set of instances $\mathcal{D}$ can be shattered by a model class $\mathcal{H}$ if, for each subset $\mathcal{P} \subseteq \mathcal{D}$, there is a model $H \in \mathcal{H}$ such that $H(\boldsymbol{x}) = 1$ for all $\boldsymbol{x} \in \mathcal{P}$ and $H(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathcal{D} \setminus \mathcal{P}$. Now, take any such subset $\mathcal{P}$ from our data set $\mathcal{D}$ as constructed above, and recall that the Choquet integral in (9) can be written as

$$\mathcal{C}_\mu(\boldsymbol{x}) = \sum_{T \subseteq C} \boldsymbol{m}(T) \times f_T(\boldsymbol{x}), \tag{11}$$

where $f_T(\boldsymbol{x}) = 1$ if $T \subseteq S_{\boldsymbol{x}}$ and $f_T(\boldsymbol{x}) = 0$ otherwise. We define the values $\boldsymbol{m}(T)$, $T \subseteq C$, of the Möbius transform as follows:

$$\boldsymbol{m}(T) = \begin{cases} |\mathcal{P}|^{-1} & \text{if } T = S_{\boldsymbol{x}} \text{ for some } \boldsymbol{x} \in \mathcal{P}, \\ 0 & \text{otherwise.} \end{cases}$$

Obviously, this definition of the Möbius transform is feasible and yields a proper fuzzy measure μ: The sum of masses is equal to 1, and since all masses are non-negative, monotonicity is guaranteed right away. Moreover, from the construction of $\boldsymbol{m}$ and the fact that, for each pair $\boldsymbol{x} \neq \boldsymbol{x}' \in \mathcal{D}$, neither $S_{\boldsymbol{x}} \subseteq S_{\boldsymbol{x}'}$ nor $S_{\boldsymbol{x}'} \subseteq S_{\boldsymbol{x}}$, the Choquet integral is obviously given as follows:

$$\mathcal{C}_\mu = \begin{cases} |\mathcal{P}|^{-1} & \text{if } \boldsymbol{x} \in \mathcal{P}, \\ 0 & \text{otherwise.} \end{cases}$$

Thus with $\beta = 1/(2|\mathcal{P}|)$, the classifier (9) behaves exactly as required, that is, it classifies all $\boldsymbol{x} \in \mathcal{P}$ as positive and all $\boldsymbol{x} \notin \mathcal{P}$ as negative.

Noting that the special case where $\mathcal{P} = \emptyset$ is handled correctly by the Möbius transform $\boldsymbol{m}$ such that $\boldsymbol{m}(C) = 1$ and $\boldsymbol{m}(T) = 0$ for all $T \subsetneq C$ (and any threshold $\beta > 0$), we can conclude that the data set $\mathcal{D}$ can be shattered by $\mathcal{H}$. Consequently, the VC dimension of $\mathcal{H}$ is at least the size of $\mathcal{D}$, whence (10) is a lower bound of $VC(\mathcal{H})$.

For the asymptotic analysis, we make use of Sterling's approximation of large factorials (and hence binomial coefficients). For the sequence $(b_1, b_2, \ldots)$ of the so-called central binomial coefficients b_n, it is known that

$$b_n = \binom{2n}{n} = \frac{(2n)!}{(n!)^2} \geq \frac{1}{2} \frac{4^n}{\sqrt{\pi \cdot n}}. \tag{12}$$

Thus, the fact that $VC(\mathcal{H})$ grows asymptotically at least as fast as $2^m/\sqrt{m}$ immediately follows by setting $n = m/2$ and ignoring constant terms. □

Remark 1 Recall the expression (8) of the Choquet integral in terms of its Möbius transform. This expression shows that the Choquet integral corresponds to a linear function, albeit a constrained one, in the *feature space* spanned by the set of features $\{f_T \mid T \subseteq \{1, 2, \ldots, m\}\}$ (already used in (11)), where each feature is a min-term

$$f_T = f_T(\boldsymbol{x}) = f_T(x_1, \ldots, x_m) = \min_{i \in T} x_i. \tag{13}$$

The dimensionality of this feature space is $2^m - 1$. Thus, it follows immediately that $VC(\mathcal{H}) \leq 2^m$ (the class of linear hyperplanes in $\mathbb{R}^n$ has VC-dimension $n + 1$). Together

with the lower bound $2^m/\sqrt{m}$, which is not much smaller (despite the restriction to binary attribute vectors), we thus dispose of a relatively tight approximation of $VC(\mathcal{H})$.

Remark 2 Interestingly, the proof of Theorem 1 does not exploit the full non-additivity of the Choquet integral. In fact, the measure we constructed there is $\lfloor m/2 \rfloor$-additive, since $\boldsymbol{m}(T) = 0$ for all $T \subseteq C$ with $|T| > \lfloor m/2 \rfloor$. Consequently, the estimation of the VC-dimension still applies to the restricted case of k-additive measures, provided $k \geq \lfloor m/2 \rfloor$. For smaller k, it is not difficult to adapt the proof so as to show that

$$VC(\mathcal{H}) \geq \binom{m}{k}. \tag{14}$$

5 Choquistic regression

Consider the standard setting of binary classification, where the goal is to predict the value of an output (response) variable $y \in \mathcal{Y} = \{0, 1\}$ for a given instance

$$\boldsymbol{x} = (x_1, \ldots, x_m) \in \mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m$$

represented in terms of a feature vector. More specifically, the goal is to learn a classifier $\mathcal{L} : \mathcal{X} \to \mathcal{Y}$ from a given set of (i.i.d.) training data

$$\mathcal{D} = \left\{ \left(\boldsymbol{x}^{(i)}, y^{(i)} \right) \right\}_{i=1}^{n} \subset (\mathcal{X} \times \mathcal{Y})^n$$

so as to minimize the risk

$$R(\mathcal{L}) = \int_{\mathcal{X} \times \mathcal{Y}} \ell\big(\mathcal{L}(\boldsymbol{x}), y\big)\, d\mathbf{P}_{XY}(\boldsymbol{x}, y),$$

where $\ell(\cdot)$ is a loss function (e.g., the simple 0/1 loss given by $\ell(\hat{y}, y) = 0$ if $\hat{y} = y$ and $= 1$ if $\hat{y} \neq y$).

Logistic regression is a well-established statistical method for (probabilistic) classification (Hosmer and Lemeshow 2000). Its popularity is due to a number of appealing properties, including monotonicity and comprehensibility: Since the model is essentially *linear* in the input attributes, the strength of influence of each predictor is directly reflected by the corresponding regression coefficient. Moreover, the influence of each attribute is *monotone* in the sense that an increase of the value of the attribute can either only increase or only decrease the probability of the positive class (depending on whether the associated regression coefficient is positive or negative).

Formally, the probability of the positive class (and hence of the negative class) is modeled as a generalized linear function of the input attributes, namely in terms of the logarithm of the probability ratio:

$$\log\left(\frac{\mathbf{P}(y = 1 \,|\, \boldsymbol{x})}{\mathbf{P}(y = 0 \,|\, \boldsymbol{x})}\right) = w_0 + \boldsymbol{w}^\top \boldsymbol{x}, \tag{15}$$

where $\boldsymbol{w} = (w_1, w_2, \ldots, w_m) \in \mathbb{R}^m$ is a vector of regression coefficients and $w_0 \in \mathbb{R}$ a constant bias (the intercept). A positive regression coefficient $w_i > 0$ means that an increase of the predictor variable x_i will increase the probability of a positive response, while a negative coefficient implies a decrease of this probability. Besides, the larger the absolute value $|w_i|$ of the regression coefficient, the stronger the influence of x_i.

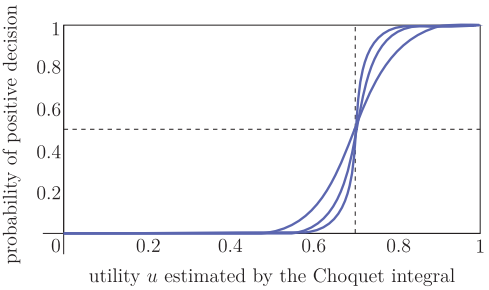

Fig. 2 Probability of a positive decision, $\mathbf{P}(y=1\,|\,\boldsymbol{x})$, as a function of the estimated degree of utility, $u=U(\boldsymbol{x})$, for a threshold $\beta=0.7$ and different values of γ

Since $\mathbf{P}(y=0\,|\,\boldsymbol{x})=1-\mathbf{P}(y=1\,|\,\boldsymbol{x})$, a simple calculation yields the posterior probability

$$\pi_l \stackrel{\text{df}}{=} \mathbf{P}(y=1\,|\,\boldsymbol{x}) = \big(1+\exp\big(-w_0-\boldsymbol{w}^\top\boldsymbol{x}\big)\big)^{-1}. \tag{16}$$

The logistic function $z\mapsto(1+\exp(-z))^{-1}$, which has a sigmoidal shape, is a specific type of *link function*.

Needless to say, the linearity of the above model is a strong restriction from a learning point of view, and the possibility of interactions between predictor variables has of course also been noticed in the statistical literature (Jaccard 2001). A standard way to handle such interaction effects is to add interaction terms to the linear function of predictor variables, like in (2). As explained earlier, however, the aforementioned advantages of logistic regression will then be lost.

In the following, we therefore propose an extension of logistic regression that allows for modeling nonlinear relationships between input and output variables while preserving the advantages of comprehensibility and monotonicity. As mentioned earlier, the monotonicity constraint is important if the direction of influence of an input attribute is known beforehand and needs to be reflected by the model, an assumption that we shall make in the following. As an aside, we note that one may also envision the case where an attribute is known to have a monotone influence, although the direction of influence is unknown. The learning problem then becomes slightly more difficult, since the learner has to figure out whether the influence is positive (increasing) or negative (decreasing). We shall not consider this problem any further, however, and instead assume the direction of influence to be given as prior knowledge.

5.1 The Choquistic model

In order to model nonlinear dependencies between predictor variables and the response, and to take interactions between predictors into account, we propose to extend the logistic regression model by replacing the linear function $\boldsymbol{x}\mapsto w_0+\boldsymbol{w}^\top\boldsymbol{x}$ in (15) with the Choquet integral. More specifically, we propose the following model

$$\pi_c \stackrel{\text{df}}{=} \mathbf{P}(y=1\,|\,\boldsymbol{x}) = \big(1+\exp\big(-\gamma\,\big(\mathcal{C}_\mu(f_{\boldsymbol{x}})-\beta\big)\big)\big)^{-1}, \tag{17}$$

where $\mathcal{C}_\mu(f_{\boldsymbol{x}})$ is the Choquet integral (with respect to the measure μ) of the function

$$f_{\boldsymbol{x}}:\{c_1,\ldots,c_m\}\to[0,1] \tag{18}$$

that maps each attribute c_i to a normalized value $x_i=f_{\boldsymbol{x}}(c_i)\in[0,1]$; $\beta,\gamma\in\mathbb{R}$ are constants.

Recalling the idea of "evaluating" an instance $\boldsymbol{x}$ in terms of a set of criteria, the model (17) can be seen as a two-step procedure: The first step consists of an assessment of $\boldsymbol{x}$ in

terms of a (latent) utility degree

$$u = U(\boldsymbol{x}) = \mathcal{C}_\mu(f_{\boldsymbol{x}}) \in [0, 1].$$

Then, in a second step, a discrete choice (yes/no decision) is made on the basis of this utility. Roughly speaking, this is done through a "probabilistic thresholding" at the utility threshold β. If $U(\boldsymbol{x}) > \beta$, then the decision tends to be positive, whereas if $U(\boldsymbol{x}) < \beta$, it tends to be negative. The precision of this decision is determined by the parameter γ (see Fig. 2): For large γ, the decision function converges toward the step function $u \mapsto \mathbb{I}(u > \beta)$, jumping from 0 to 1 at β. For small γ, this function is smooth, and there is a certain probability to violate the threshold rule $u \mapsto \mathbb{I}(u > \beta)$. This might be due to the fact that, despite being important for decision making, some properties of the instances to be classified are not captured by the utility function. In that case, the utility $U(\boldsymbol{x})$, estimated on the basis of the given attributes, is not a perfect predictor for the decision eventually made. Thus, the parameter γ can also be seen as an indicator of the quality of the classification model.

5.2 Normalization

The normalization (18) is meant to turn each predictor variable into a criterion, i.e., a "the higher the better" attribute, and to assure commensurability between the criteria (Modave and Grabisch 1998). A simple transformation is given by the mapping

$$z_i = \frac{x_i - m_i}{M_i - m_i}, \tag{19}$$

where m_i and M_i are lower and upper bounds for x_i, which are either known or estimated from the data; if the influence of x_i is actually negative (i.e., $w_i < 0$), then the mapping $z_i = (M_i - x_i)/(M_i - m_i)$ is used instead.

The transformation (19) is problematic in the presence of outliers, in which case the distribution of its image can become extremely skewed. As an alternative, which is less sensitive in this regard and, moreover, produces a more uniform distribution of normalized values, we therefore propose the mapping

$$z_i = F^{-1}(x_i), \tag{20}$$

where F is the cumulative distribution function $x \mapsto \mathbf{P}(X_i \leq x)$. Of course, since this function is in general not known, it has to be replaced by an estimate $\hat{F}$; to this end, we simply adopt the empirical distribution of the training data (i.e., $\hat{F}(x)$ is the relative frequency of instances $\boldsymbol{x} = (x_1, \ldots, x_m)$ in the training data for which $x_i \leq x$).

5.3 Logistic regression as a special case

In order to verify that our model (17) is a proper generalization of standard logistic regression, recall that the Choquet integral reduces to a weighted mean (7) in the special case of an additive measure μ. Moreover, recall the transformation (19) and consider any linear function $\boldsymbol{x} \mapsto g(\boldsymbol{x}) = w_0 + \boldsymbol{w}^\top \boldsymbol{x}$ with $\boldsymbol{w} = (w_1, \ldots, w_m)$. This function can also be written in the form

$$\begin{aligned} g(\boldsymbol{x}) &= w_0 + \sum_{i=1}^{m} \big(w_i p_i + |w_i|(M_i - m_i) z_i \big) \\ &= w_0 + \sum_{i=1}^{m} w_i p_i + \sum_{i=1}^{m} |w_i|(M_i - m_i) z_i \end{aligned}$$

$$= w_0' + \left(\sum_{i=1}^{m} u_i\right)^{-1} \sum_{i=1}^{m} u_i' z_i$$
$$= \gamma\left(\sum_{i=1}^{m} u_i' z_i - \beta\right),$$

where $p_i = m_i$ if $w_i \geq 0$ and $p_i = M_i$ if $w_i < 0$, $u_i = |w_i|(M_i - m_i)$, $\gamma = (\sum_{i=1}^{m} u_i)^{-1}$, $u_i' = u_i/\gamma$, $w_0' = w_0 + \sum_{i=1}^{m} w_i p_i$, $\beta = -w_0'/\gamma$. By definition, the u_i' are non-negative and sum up to 1, which means that $\sum_{i=1}^{m} u_i' z_i$ is a weighted mean of the z_i that can be represented by a Choquet integral.

5.4 Parameter estimation

The model (17) has several degrees of freedom: The fuzzy measure μ (Möbius transform $\boldsymbol{m} = \boldsymbol{m}_\mu$) determines the (latent) utility function, while the utility threshold β and the scaling parameter γ determine the discrete choice model. The goal of learning is to identify these degrees of freedom on the basis of the training data $\mathcal{D}$. Like in the case of standard logistic regression, it is possible to harness the maximum likelihood (ML) principle for this purpose. The log-likelihood of the parameters can be written as

$$\begin{aligned} l(\boldsymbol{m}, \gamma, \beta) &= \log \mathbf{P}(\mathcal{D} \mid \boldsymbol{m}, \beta, \gamma) \\ &= \log\left(\prod_{i=1}^{n} \mathbf{P}\big(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{m}, \beta, \gamma\big)\right) \\ &= \sum_{i=1}^{n} y^{(i)} \log \pi_c^{(i)} + \big(1 - y^{(i)}\big) \log\big(1 - \pi_c^{(i)}\big). \end{aligned} \tag{21}$$

One easily verifies that (21) is convex with respect to $\boldsymbol{m}$, γ, and β. In principle, maximization of the log-likelihood can be accomplished by means of standard gradient-based optimization methods. However, since we have to assure that μ is a proper fuzzy measure and, hence, that $\boldsymbol{m}$ guarantees the corresponding monotonicity and boundary conditions, we actually need to solve a *constrained* optimization problem:

$$\begin{aligned} \max_{\boldsymbol{m},\gamma,\beta} \quad & \left\{ -\gamma \sum_{i=1}^{n} \big(1 - y^{(i)}\big)\big(\mathcal{C}_{\boldsymbol{m}}\big(\boldsymbol{x}^{(i)}\big) - \beta\big) \right. \\ & \left. - \sum_{i=1}^{n} \log\big(1 + \exp\big(-\gamma\,\big(\mathcal{C}_{\boldsymbol{m}}\big(\boldsymbol{x}^{(i)}\big) - \beta\big)\big)\big) - \eta \sum_{T \subseteq C} \big|\boldsymbol{m}(T)\big| \right\} \\ \text{s.t.} \quad & \eta, \gamma > 0, \qquad 0 \leq \beta \leq 1, \qquad \sum_{T \subseteq C} \boldsymbol{m}(T) = 1, \quad \text{and} \\ & \sum_{B \subseteq A \setminus \{c_i\}} \boldsymbol{m}\big(B \cup \{c_i\}\big) \geq 0 \quad \forall A \subseteq C,\ \forall c_i \in C. \end{aligned} \tag{22}$$

The last part of the objective function (22) is a standard L_1-regularizer on the Möbius transform, which is added as a means to prevent over-fitting; moreover, since many weights are typically set to 0 under L_1-regularization, it also serves as a feature selection mechanism (Lee et al. 2006).

A solution to the above problem can be produced by standard solvers. Concretely, we used the `fmincon` function implemented in the optimization toolbox of Matlab. This method is based on a sequential quadratic programming approach.

Recall that, once the model has been identified, the importance of each attribute and the degree of interaction between groups of attributes can be derived from the Möbius transform $\boldsymbol{m}$; these are given, respectively, by the Shapley value and the interaction indexes as introduced in Sect. 3.2.

6 Complexity reduction

Obviously, choquistic regression can be interpreted as fitting a (constrained) linear function in the feature space spanned by the set of features f_T defined by (13), with one feature for each subset of criteria $T \subseteq \{1, 2, \ldots, m\}$. Since the dimensionality of this feature space is $2^m - 1$, the method is clearly critical from a complexity point of view. It was already mentioned that an L_1-regularization in (22) may shrink some coefficients to 0 and, therefore, some of the features f_T may disappear. Although this may help to simplify the choquistic model, that is, the result produced by the learning algorithm, it does not simplify the optimization problem itself.

Thus, one may wonder whether some of the features (13) could not even be eliminated *prior* to solving the actual optimization problem. Specifically interesting in this regard is a possible restriction of the choquistic model to k-additive measures, for a suitable value of $k < m$. Since this means that significantly less parameters (namely $2^k - 1$) need to be identified, the computational complexity might be reduced drastically. Besides, a restriction to k-additive measures may also have advantages from a learning point of view, as it reduces the capacity of the underlying model class (cf. Sect. 4) and thus may prevent from overfitting the data in cases where the full flexibility of the Choquet integral is actually not needed. Of course, the key problem to be addressed concerns the question of how to choose k in the most favorable way.

6.1 Exploiting equivalence of features for dimensionality reduction

In the following, we shall elaborate on the following question: Is it possible to find an upper bound on the required level of complexity of the model, namely the level of additivity k, prior to fitting the Choquet integral to the data? Or, more specifically, can we determine the value k in such a way that fitting a k-additive measure is definitely enough, in the sense that each labeling of the training data produced by the full Choquet integral ($k = m$) can also be produced by a Choquet integral based on a k-additive measure?

In this regard, it is noticeable that, for a given instance $\boldsymbol{x} = (x_1, \ldots, x_m)$, many of the min-terms (13) will assume the same value (in fact, there are $2^m - 1$ such terms but only m possible values). Consequently, in the expression

$$\mathcal{C}_\mu(\boldsymbol{x}) = \sum_{T \subseteq C} \boldsymbol{m}(T) \times f_T(\boldsymbol{x}) \tag{23}$$

of the Choquet integral, many coefficients $\boldsymbol{m}(T)$ can be grouped and, in principle, be replaced by a single one. The groups thus defined solely depend on the order of the values $x_1, \ldots, x_m$ of the original attributes. The number of terms in (23) will thus reduce from $2^m - 1$ to at most m. However, since the order may change from instance to instance, different groupings may be obtained for different instances.

Now, imagine that a subset of features $\mathcal{F} = \{f_{T_1}, \ldots, f_{T_r}\}$ assumes the same value, not only for a single instance, but for all instances in the training data. Then, this set can be said to form an equivalence class. Thus, one of the features could in principle be selected as a representative, absorbing all the weights of the others; more specifically, the weight of this feature would be set to $\boldsymbol{m}(T_1) + \boldsymbol{m}(T_2) + \cdots + \boldsymbol{m}(T_r)$, while the weights of the other features in $\mathcal{F}$ would be set to 0.

Note, however, that this "transfer of Möbius mass" will in general not be feasible, as it may cause a violation of the monotonicity constraint on the fuzzy measure μ. As a side remark, we also note that, from a learning point of view, the equivalence of features may obviously cause problems with regard to the identifiability of coefficients; due to the monotonicity constraints just mentioned, however, this is not necessarily the case.

More generally, for two features f_A and f_B $(A, B \subseteq C)$, denote by $v(A, B) \in [0, 1]$ the fraction of training examples on which they assume the same value. We say that f_A covers f_B (and, vice versa, f_B covers f_A) if $v(A, B) = 1$. Moreover, for a feature f_A, we denote by $C(f_A) \subseteq 2^C$ the set of features it covers. A straightforward way to find a sufficiently large k then consists of finding the smallest k such that

$$\bigcup_{T \subseteq C, |T| \leq k} C(f_T) = 2^C. \tag{24}$$

From the above construction, it follows that working with the corresponding k-additive measure, for k thus defined, is theoretically sound and guarantees that there is no loss in terms of expressivity of the model on the training data. We summarize this finding in terms of the following proposition.

Proposition 1 *Consider a set of training instances $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}$ and let k^* be the smallest value in $\{1, \ldots, m\}$ satisfying (24). Moreover, let μ be any measure on the set of criteria $\{c_1, \ldots, c_m\}$, and $\mathcal{C}_\mu$ the Choquet integral with respect to this measure. Then, there exists a k-additive measure μ^* such that*

$$\mathcal{C}_{\mu^*}\big(\boldsymbol{x}^{(i)}\big) = \mathcal{C}_\mu\big(\boldsymbol{x}^{(i)}\big) \tag{25}$$

for all $i \in \{1, \ldots, n\}$.

We like to emphasize that k^* is only an upper bound on the complexity needed to fit the training data. Thus, it is not necessarily the optimal k from the point of view of model induction (which might be figured out by the regularizer in (22)). In particular, note that the computation of k^* does not refer to the output values $y^{(i)}$. Instead, it should be considered as a measure of the complexity of the training instances. As such, it is obviously connected to the notion of VC dimension.

Since the exact reproducibility (25) may appear overly stringent or, stated differently, a small loss may actually be acceptable, we finally propose a relaxation somewhat in line with idea of *probably approximately correct* (PAC) learning (Valiant 1984). First, noting that the Choquet integral may change by at most ϵ when combining features f_A and f_B such that $|f_A - f_B| < \epsilon$, one may think of relaxing the definition of equivalence as follows: f_A and f_B are ϵ-equivalent (on a given training instance $\boldsymbol{x}$) if $|f_A(\boldsymbol{x}) - f_B(\boldsymbol{x})| < \epsilon$. Second, we relax the condition of coverage. Denoting by $v(A, B) \in [0, 1]$ the fraction of training examples on which f_A and f_B are ϵ-equivalent, we say that f_A ϵ-δ-covers f_B if $v(A, B) \geq 1 - \delta$. Roughly speaking, for a small ϵ and δ close to 0, this means that, with only a few exceptions, the values of f_A and f_B are almost the same on the training data (we used $\epsilon = \delta = 0.1$ is our

Table 1 Data sets and their properties

data set	#instances	#attributes	source
Den Bosch (DBS)	120	8	Daniels and Kamp (1999)
CPU	209	6	UCI
Breast Cancer (BCC)	286	7	UCI
Auto MPG	392	7	UCI
Employee Selection (ESL)	488	4	WEKA
Mammographic (MMG)	961	5	UCI
Employee Rejection/Acceptance (ERA)	1000	4	WEKA
Lecturers Evaluation (LEV)	1000	4	WEKA
Car Evaluation (CEV)	1728	6	UCI

experiments below). In order to find a proper upper bound k^*, the principle (24) can be used as before, just replacing coverage with ϵ-δ-coverage.

7 Experiments

In this section, we present the results of an experimental study that was conducted in order to validate the practical performance of our choquistic regression (CR) method. The goal of this study is twofold. First, we would like to show that CR is competitive in terms of predictive accuracy. To this end, we compare it with several alternative methods on a number of (monotone) benchmark data sets. Second, we would like to corroborate our claim that the CR model provides interesting information about attribute importance and interaction. To this end, we discuss some examples showing that the corresponding Shapley and interaction values produced by CR are indeed meaningful and plausible.

7.1 Data sets

Although the topic is receiving increasing interest in the machine learning community, benchmark data for monotone classification is by far not as abundant as for conventional classification. In total, we managed to collect 9 data sets from different sources, notably the UCI repository[2] and the WEKA machine learning framework (Hall et al. 2009), for which monotonicity in the input variables is a reasonable assumption; see Table 1 for a summary. All the data sets can be downloaded from our website.[3] Many of them have also been used in previous studies on monotone learning. Some of them have a numerical or ordered categorical output, however, which was hence binarized. Moreover, all input attributes were normalized using (20).

Den Bosch (DBS) This data set contains 8 attributes describing houses in the city of Den Bosch: area, number of bedrooms, type of house, volume, storeys, type of garden, garage, and price. The output is a binary variable indicating whether the price of the house is low or high (depending on whether or not it exceeds a threshold).

[2] http://archive.ics.uci.edu/ml/.

[3] http://www.uni-marburg.de/fb12/kebi/.

CPU This is a standard benchmark data set from the UCI repository. It contains eight input attributes, two of which were removed since they are obviously of no predictive value (vendor name, model name). The problem is to predict the (estimated) relative performance of a CPU (binarized by thresholding at the median) based on its machine cycle time in nanoseconds, minimum main memory in kilobytes, maximum main memory in kilobytes, cache memory in kilobytes, minimum channels in units, maximum channels in units.

Breast Cancer (BCC) This data set was obtained from the University Medical Center, Institute of Oncology, Ljubljana, Yugoslavia. There are 7 attributes, namely menopause gain, tumor-size, inv-nodes, node-caps, deg-malig, breast cost, irradiat gain. The output is a binary variable, namely no-recurrence-events and recurrence-events.

Auto MPG This data set was used in the 1983 American Statistical Association Exposition. The problem is to predict the city-cycle fuel consumption in miles per gallon (binarized by thresholding at the median) based on the following attributes of a car: cylinders, displacement, horsepower, weight, acceleration, model year, origin. We removed incomplete instances.

Employee Selection (ESL) This data set contains profiles of applicants for certain industrial jobs. The values of the four input attributes were determined by expert psychologists based upon psychometric test results and interviews with the candidates. The output is an overall score on an ordinal scale between 1 and 9, corresponding to the degree of suitability of each candidate to this type of job. We binarized the output value by distinguishing between suitable (score 6–9) and unsuitable (score 1–5) candidates.

Mammographic (MMG) This data set is about breast cancer screening by mammography. The goal is to predict the severity (benign or malignant) of a mammographic mass lesion from BI-RADS attributes (mass shape, mass margin, density) and the patient's age.

Employee Rejection/Acceptance (ERA) This data set originates from an academic decision-making experiment. The input attributes are features of a candidate such as past experience, verbal skills, etc., and the output is the subjective judgment of a decision-maker, measured on an ordinal scale from 1 to 9, to which degree he or she tends to accept the applicant for the job. We binarized the output value by distinguishing between acceptance (score 5–9) and rejection (score 1–4).

Lecturers Evaluation (LEV) This data set contains examples of anonymous lecturer evaluations, taken at the end of MBA courses. Students were asked to score their lecturers according to four attributes such as oral skills and contribution to their professional/general knowledge. The output was a total evaluation of each lecturer's performance, measured on an ordinal scale from 0 to 4. We binarized the output value by distinguishing between good (score 3–4) and bad evaluation (score 0–2).

Car Evaluation (CEV) This data set contains 6 attributes describing a car, namely, buying price, price of the maintenance, number of doors, capacity in terms of persons to carry, the size of luggage boot, estimated safety of the car. The output is the overall evaluation of the car: unacceptable, acceptable, good, very good. We binarized this evaluation into unacceptable versus not unacceptable (acceptable, good or very good).

7.2 Methods

Since choquistic regression (CR) can be seen as an extension of standard logistic regression (LR), it is natural to compare these two methods. Essentially, this comparison should give an idea of the usefulness of an increased flexibility. On the other side, one may also ask for the usefulness of assuring monotonicity. Therefore, we additionally included two other extensions of LR, which are flexible but not necessarily monotone, namely kernel logistic regression (KLR) with polynomial and Gaussian kernels. The degree of the polynomial kernel was set to 2, so that it models low-level interactions of the features. The Gaussian kernel, on the other hand, is able to capture interactions of higher order. For each data set, the width parameter of the Gaussian kernel was selected from $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$ in the most favorable way. Likewise, the regularization parameter η in choquistic regression was selected from $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$.

Finally, we also included two methods that are both monotone and flexible, namely the MORE algorithm for learning rule ensembles under monotonicity constraints (Dembczyński et al. 2009) and the LMT algorithm for logistic model tree induction (Landwehr et al. 2003). Following the idea of forward stagewise additive modeling (Tibshirani et al. 2001), the MORE algorithm treats a single rule as a subsidiary base classifier in the ensemble. The rules are added to the ensemble one by one. Each rule is fitted by concentrating on the examples that are most difficult to classify correctly by rules that have already been generated. The LMT algorithm builds tree-structured models that contain logistic regression functions at the leaves. It is based on a stagewise fitting process to construct the logistic regression models that can select relevant attributes from the data. This process is used to build the logistic regression models at the leaves by incrementally refining those constructed at higher levels in the tree structure.

7.3 Results

7.3.1 Performance in terms of predictive accuracy

As performance measures, we determined the standard misclassification rate (0/1 loss) as well as the AUC. Estimates of both measures were obtained by randomly splitting the data into two parts, one part for training and one part for testing. This procedure was repeated 100 times, and the results were averaged. In order to analyze the influence of the amount of training data, we varied the proportion between training and test data from 20:80 over 50:50 to 80:20. In these experiments, we used a variant of CR in which the underlying fuzzy measure is restricted to be k-additive, with k determined by means of an internal cross-validation. Compared with other variants (cf. Sect. 7.3.2), this one performed best in terms of accuracy.

A possible improvement of CR over its competitors, in terms of predictive accuracy, may be due to two reasons: First, in comparison to standard LR, it is more flexible and has the ability to capture nonlinear dependencies between input attributes. Second, in comparison to non-monotone learners, it takes background knowledge about the dependency between input and output variables into consideration.

An overview of the results of the experiments is given in Tables 2 and 3. Moreover, a summary in terms of pairwise win statistics is provided in Tables 4 and 5. As can be seen, CR compares quite favorably with the other approaches, especially with the non-monotone KLR methods, both in terms of 0/1 loss and AUC. It also outperforms LR, at least for sufficiently extensive training data; if the amount of training data is small, however, LR is

Table 2 Classification performance in terms of the mean and standard deviation of 0/1 loss. From top to bottom: 20 %, 50 %, and 80 % training data. (Average ranks comparing significantly worse with CR at the 90 % confidence level are put in bold font)

data set	CR	LR	KLR-ply	KLR-rbf	MORE	LMT
DBS	.1713 ± .0424(2)	.2124 ± .0650(6)	.1695 ± .0437(1)	.1883 ± .0536(4)	.1932 ± .0511(5)	.1779 ± .0420(3)
CPU	.0811 ± .0103(3)	.0711 ± .0312(1)	.0996 ± .0231(6)	.0802 ± .0292(2)	.0829 ± .0379(4)	.0850 ± .0256(5)
BCC	.2775 ± .0335(2)	.2893 ± .0240(6)	.2760 ± .0243(1)	.2787 ± .0237(3)	.2827 ± .0255(4)	.2884 ± .0306(5)
MPG	.0709 ± .0193(1)	.0832 ± .0151(6)	.0788 ± .0097(4)	.0772 ± .0107(2)	.0811 ± .0119(5)	.0773 ± .0148(3)
ESL	.0682 ± .0129(1)	.0733 ± .0107(2)	.1488 ± .0278(6)	.0756 ± .0167(3)	.0838 ± .0241(5)	.0771 ± .0148(4)
MMG	.1725 ± .0120(1)	.1729 ± .0122(2)	.1960 ± .0160(6)	.1791 ± .0133(4)	.1764 ± .0137(3)	.1803 ± .0171(5)
ERA	.2889 ± .0273(1)	.2902 ± .0317(2)	.3001 ± .0130(5)	.2934 ± .0112(3)	.3155 ± .0150(6)	.2963 ± .0126(4)
LEV	.1499 ± .0122(1)	.1655 ± .0082(3)	.1627 ± .0119(2)	.1691 ± .0125(5)	.1707 ± .0186(6)	.1672 ± .0140(4)
CEV	.0448 ± .0089(3)	.1410 ± .0079(6)	.0663 ± .0130(5)	.0618 ± .0151(4)	.0339 ± .0076(1)	.0432 ± .0116(2)
avg. rank	1.67	3.78	**4**	3.33	**4.33**	3.89
DBS	.1572 ± .0416(4)	.1708 ± .0380(6)	.1333 ± .0333(1)	.1692 ± .0382(5)	.1457 ± .0413(3)	.1473 ± .0406(2)
CPU	.0464 ± .0281(1)	.0626 ± .0247(4)	.0835 ± .0264(6)	.0547 ± .0233(3)	.0489 ± .0226(2)	.0674 ± .0243(5)
BCC	.2687 ± .0282(4)	.2799 ± .0245(6)	.2591 ± .0287(1)	.2599 ± .0301(2)	.2640 ± .0288(3)	.2717 ± .0295(5)
MPG	.0577 ± .0251(1)	.0654 ± .0150(2)	.0728 ± .0159(4)	.0744 ± .0151(5)	.0751 ± .0178(6)	.0672 ± .0164(3)
ESL	.0601 ± .0126(1)	.0704 ± .0113(4)	.1023 ± .0225(6)	.0682 ± .0121(2)	.0695 ± .0139(3)	.0709 ± .0135(5)
MMG	.1667 ± .0144(1)	.1701 ± .0158(5)	.1721 ± .0164(6)	.1693 ± .0130(4)	.1691 ± .0140(3)	.1671 ± .0167(2)
ERA	.2844 ± .0306(1)	.2851 ± .0303(2)	.2926 ± .0151(4)	.2882 ± .0142(3)	.3037 ± .0180(6)	.2956 ± .0148(5)
LEV	.1372 ± .0125(1)	.1651 ± .0133(6)	.1520 ± .0160(4)	.1493 ± .0165(3)	.1486 ± .0157(2)	.1545 ± .0142(5)
CEV	.0376 ± .0059(4)	.1360 ± .0101(6)	.0328 ± .0057(3)	.0463 ± .0086(5)	.0215 ± .0053(2)	.0174 ± .0069(1)
avg. rank	2	**4.56**	3.89	3.56	3.33	3.67

Table 2 (*Continued*)

data set	CR	LR	KLR-ply	KLR-rbf	MORE	LMT
DBS	.1416 ± .0681(4)	.1616 ± .0743(6)	.1265 ± .0663(2)	.1343 ± .0672(3)	.1242 ± .0609(1)	.1433 ± .0667(5)
CPU	.0212 ± .0301(1)	.0640 ± .0335(5)	.0754 ± .0372(6)	.0405 ± .0284(3)	.0412 ± .0299(4)	.0338 ± .0352(2)
BCC	.2496 ± .0485(1)	.2773 ± .0548(6)	.2569 ± .0506(2)	.2598 ± .0529(4)	.2570 ± .0463(3)	.2707 ± .0554(5)
MPG	.0551 ± .0160(1)	.0611 ± .0263(2)	.0727 ± .0268(4)	.0740 ± .0284(6)	.0737 ± .0269(5)	.0614 ± .0251(3)
ESL	.0542 ± .0218(1)	.0660 ± .0203(3)	.0922 ± .0279(6)	.0657 ± .0229(2)	.0661 ± .0219(4)	.0691 ± .0228(5)
MMG	.1584 ± .0251(1)	.1657 ± .0232(4)	.1741 ± .0246(6)	.1696 ± .0271(5)	.1645 ± .0235(3)	.1595 ± .0283(2)
ERA	.2813 ± .0280(1)	.2843 ± .0302(2)	.2918 ± .0290(5)	.2905 ± .0312(3)	.2988 ± .0276(6)	.2910 ± .0290(4)
LEV	.1314 ± .0176(1)	.1627 ± .0249(6)	.1472 ± .0231(3)	.1496 ± .0233(5)	.1397 ± .0214(2)	.1474 ± .0232(4)
CEV	.0273 ± .0089(4)	.1328 ± .0173(6)	.0286 ± .0075(5)	.0239 ± .0066(3)	.0190 ± .0070(2)	.0089 ± .0047(1)
avg. rank	1.67	**4.44**	**4.33**	3.78	3.33	3.44

Table 3 Performance in terms the average AUC ± standard deviation. From top to bottom: 20 %, 50 %, and 80 % training data. (Average ranks comparing significantly worse with CR at the 90 % confidence level are put in bold font)

data set	CR	LR	KLR-ply	KLR-rbf	MORE	LMT
DBS	.9290 ± .0322(2)	.8866 ± .0511(5)	.9359 ± .0218(1)	.9053 ± .0433(4)	.8731 ± .0481(6)	.9151 ± .0228(3)
CPU	.9822 ± .0121(2)	.9806 ± .0124(4)	.9716 ± .0072(6)	.9843 ± .0116(1)	.9749 ± .0235(5)	.9816 ± .0113(3)
BCC	.6400 ± .0641(6)	.6970 ± .0411(3)	.6509 ± .0568(5)	.7124 ± .0290(2)	.6639 ± .0567(4)	.7310 ± .0675(1)
MPG	.9788 ± .0160(1)	.9675 ± .0068(5)	.9704 ± .0075(4)	.9741 ± .0055(3)	.9501 ± .0263(6)	.9753 ± .0092(2)
ESL	.9670 ± .0074(4)	.9721 ± .0060(1)	.9638 ± .0106(5)	.9705 ± .0099(2)	.9466 ± .0484(6)	.9696 ± .0086(3)
MMG	.8867 ± .0123(4)	.8962 ± .0080(1)	.8552 ± .0203(6)	.8938 ± .0121(2)	.8754 ± .0274(5)	.8890 ± .0259(3)
ERA	.7669 ± .0334(1)	.7602 ± .0331(4)	.7555 ± .0139(5)	.7662 ± .0098(2)	.7198 ± .0329(6)	.7619 ± .0160(3)
LEV	.8971 ± .0098(1)	.8905 ± .0081(2)	.8870 ± .0094(3)	.8860 ± .0128(4)	.8137 ± .0621(6)	.8797 ± .0182(5)
CEV	.9825 ± .0080(3)	.9332 ± .0033(6)	.9818 ± .0058(5)	.9821 ± .0076(4)	.9888 ± .0063(2)	.9902 ± .0042(1)
avg. rank	2.67	3.44	4.44	2.67	**5.11**	2.67
DBS	.9341 ± .0228(2)	.9191 ± .0293(4)	.9492 ± .0198(1)	.9174 ± .0316(6)	.9179 ± .0403(5)	.9259 ± .0289(3)
CPU	.9920 ± .0073(2)	.9914 ± .0056(3)	.9771 ± .0109(6)	.9925 ± .0056(1)	.9873 ± .0149(5)	.9883 ± .0077(4)
BCC	.6912 ± .0469(6)	.7184 ± .0367(3)	.7001 ± .0396(4)	.7294 ± .0344(2)	.6980 ± .0586(5)	.7387 ± .0656(1)
MPG	.9818 ± .0075(1)	.9803 ± .0084(3)	.9776 ± .0083(4)	.9752 ± .0068(5)	.9563 ± .0313(6)	.9814 ± .0074(2)
ESL	.9720 ± .0084(4)	.9764 ± .0062(1)	.9726 ± .0080(3)	.9754 ± .0070(2)	.9557 ± .0301(6)	.9707 ± .0120(5)
MMG	.9003 ± .0132(1)	.8972 ± .0125(4)	.8962 ± .0140(5)	.8995 ± .0091(2)	.8839 ± .0305(6)	.8976 ± .0153(3)
ERA	.7705 ± .0310(4)	.7633 ± .0241(5)	.7740 ± .0148(2)	.7745 ± .0141(1)	.7215 ± .0381(6)	.7719 ± .0144(3)
LEV	.9098 ± .0103(1)	.8935 ± .0113(4)	.8999 ± .0120(3)	.9012 ± .0128(2)	.8185 ± .0580(6)	.8920 ± .0164(5)
CEV	.9912 ± .0024(4)	.9362 ± .0071(6)	.9950 ± .0019(2)	.9907 ± .0031(5)	.9921 ± .0042(3)	.9977 ± .0017(1)
avg. rank	2.78	3.67	3.33	2.89	**5.33**	3

Table 3 (*Continued*)

data set	CR	LR	KLR-ply	KLR-rbf	MORE	LMT
DBS	.9427 ± .0443(3)	.9224 ± .0514(6)	.9608 ± .0347(1)	.9495 ± .0459(2)	.9409 ± .0539(4)	.9343 ± .0479(5)
CPU	.9971 ± .0063(2)	.9907 ± .0085(5)	.9827 ± .0167(6)	.9984 ± .0052(1)	.9909 ± .0167(4)	.9959 ± .0078(3)
BCC	.7349 ± .0692(1)	.7253 ± .0715(4)	.7071 ± .0720(5)	.7335 ± .0690(3)	.7042 ± .0853(6)	.7342 ± .0791(2)
MPG	.9855 ± .0108(1)	.9843 ± .0138(2)	.9797 ± .0121(4)	.9771 ± .0142(5)	.9551 ± .0372(6)	.9841 ± .0106(3)
ESL	.9766 ± .0150(2)	.9722 ± .0167(4)	.9746 ± .0141(3)	.9782 ± .0126(1)	.9507 ± .0508(6)	.9713 ± .0176(5)
MMG	.9135 ± .0233(1)	.9048 ± .0237(3)	.9011 ± .0199(4)	.8991 ± .0255(5)	.8889 ± .0363(6)	.9063 ± .0215(2)
ERA	.7670 ± .0290(4)	.7630 ± .0281(5)	.7731 ± .0293(3)	.7759 ± .0315(1)	.7228 ± .0475(6)	.7735 ± .0296(2)
LEV	.9122 ± .0202(1)	.8928 ± .0234(5)	.9048 ± .0201(2)	.9031 ± .0172(3)	.8078 ± .0661(6)	.8996 ± .0222(4)
CEV	.9959 ± .0027(3)	.9352 ± .0095(6)	.9942 ± .0018(4)	.9970 ± .0013(2)	.9936 ± .0046(5)	.9993 ± .0017(1)
avg. rank	2	**4.44**	3.56	2.56	**5.44**	3

Table 4 Win statistics (number of data sets on which the first method was better than the second one) for 20 %, 50 %, and 80 % training data for 0/1 loss case

	CR	LR	KLR-ply	KLR-rbf	MORE	LMT
CR	–	8 | 9 | 9	7 | 6 | 8	8 | 8 | 7	8 | 6 | 7	8 | 7 | 8
LR	1 | 0 | 0	–	4 | 5 | 5	5 | 2 | 3	5 | 2 | 3	5 | 4 | 3
KLR-ply	2 | 3 | 1	5 | 4 | 4	–	3 | 4 | 4	5 | 4 | 3	3 | 4 | 3
KLR-rbf	1 | 1 | 2	4 | 7 | 6	6 | 5 | 5	–	7 | 4 | 3	6 | 5 | 4
MORE	1 | 3 | 2	4 | 7 | 6	4 | 5 | 6	2 | 5 | 6	–	4 | 4 | 4
LMT	1 | 2 | 1	4 | 5 | 6	6 | 5 | 6	3 | 4 | 5	5 | 5 | 5	–

Table 5 Win statistics (number of data sets on which the first method was better than the second one) for 20 %, 50 %, and 80 % training data for AUC case

	CR	LR	KLR-ply	KLR-rbf	MORE	LMT
CR	–	6 | 7 | 9	7 | 4 | 7	5 | 5 | 4	7 | 7 | 9	5 | 6 | 7
LR	3 | 2 | 0	–	6 | 5 | 4	3 | 3 | 2	8 | 8 | 6	3 | 3 | 2
KLR-ply	2 | 5 | 2	3 | 4 | 5	–	2 | 3 | 4	5 | 8 | 8	2 | 4 | 3
KLR-rbf	4 | 4 | 5	6 | 6 | 7	7 | 6 | 5	–	8 | 7 | 9	5 | 5 | 5
MORE	2 | 2 | 0	1 | 1 | 3	4 | 1 | 1	1 | 2 | 0	–	0 | 0 | 1
LMT	4 | 3 | 2	6 | 6 | 7	7 | 5 | 6	4 | 4 | 4	9 | 9 | 8	–

even better, probably because CR will then tend to overfit the data. This is indeed a general trend that can be observed both for performance in terms of average ranks and the number of wins in pairwise comparison with another method: The more training data is available, the better CR becomes, arguably because its flexibility is then becoming more and more advantageous.

Needless to say, statistical significance is difficult to achieve due to the limited number of data sets. In terms of pairwise comparison, for example, a standard sign test will not report a significant difference (at the 10 % significance level) unless one of the method wins at least 7 of the 9 data sets. For the 0/1 loss, this is indeed accomplished by CR in all cases except two (comparison with KLR-ply and MORE on 50 % training data); see Table 4. For AUC, CR is superior, too, but significance is reached less often; see Table 5.

We also applied the two-step procedure recommended by Demsar (2006), consisting of a Friedman test and (provided this one rejects the null-hypothesis of overall equal performance of all methods) the subsequent use of a Nemenyi test in order to compare methods in a pairwise manner; both tests are based on average ranks. For both 0/1 loss and AUC, the Friedman test finds significant differences among the six classifiers (at the 10 % significance level) when all three different proportions of data are used for training. The critical distance of ranks in the Nemenyi test is 2.28 for both measures. In Tables 2 and 3, the average ranks for which this difference is exceeded are highlighted in bold font.

7.3.2 Variants of choquistic regression

In the above experiments, we used CR with a fuzzy measure of optimal order, namely a k-additive measure with k determined through internal cross-validation. In addition, we also learned with standard CR, i.e., CR using the full fuzzy measure with $k = m$ (the number of attributes). As can be seen in Table 6, adapting k does obviously pay off and leads to

Table 6 Performance in terms the average 0/1 loss and AUC ± standard deviation for CR using the full fuzzy measure compared with using a k-additive measure with cross-validated k. From top to bottom: 20 %, 50 %, and 80 % training data

data set	0/1 loss full	0/1 loss k-additive	AUC full	AUC k-additive
DBS	.2329 ± .0518	**.1713 ± .0424**	.8981 ± .0135	**.9290 ± .0322**
CPU	.1341 ± .0802	**.0811 ± .0103**	.9505 ± .0377	**.9822 ± .0121**
BCC	.3342 ± .0252	**.2775 ± .0335**	.6112 ± .0678	**.6400 ± .0641**
MPG	.0709 ± .0193	.0709 ± .0193	.9788 ± .0182	.9788 ± .0182
ESL	.0730 ± .0168	**.0682 ± .0129**	.9667 ± .0085	**.9670 ± .0074**
MMG	.1776 ± .0101	**.1725 ± .0120**	**.8899 ± .0145**	.8867 ± .0123
ERA	.2981 ± .0158	**.2889 ± .0273**	.7579 ± .0103	**.7669 ± .0334**
LEV	.1526 ± .0146	**.1499 ± .0122**	**.8984 ± .0103**	.8971 ± .0098
CEV	.0448 ± .0089	.0488 ± .0089	.9825 ± .0080	.9825 ± .0080
DBS	.2261 ± .0685	**.1572 ± .0416**	.8995 ± .0486	**.9341 ± .0228**
CPU	.0702 ± .0912	**.0464 ± .0281**	.9834 ± .0256	**.9920 ± .0073**
BCC	.3122 ± .0324	**.2687 ± .0282**	.6596 ± .0309	**.6912 ± .0469**
MPG	.0577 ± .0251	.0577 ± .0251	.9818 ± .0075	.9818 ± .0075
ESL	.0711 ± .0133	**.0601 ± .0126**	.9695 ± .0102	**.9720 ± .0084**
MMG	.1671 ± .0139	**.1667 ± .0144**	.8940 ± .0110	**.9003 ± .0132**
ERA	.2930 ± .0162	**.2844 ± .0306**	.7641 ± .0146	**.7705 ± .0310**
LEV	.1421 ± .0142	**.1372 ± .0125**	.9088 ± .0132	**.9098 ± .0103**
CEV	.0376 ± .0059	.0376 ± .0059	.9912 ± .0024	.9912 ± .0024
DBS	.2192 ± .0466	**.1416 ± .0681**	.9052 ± .0210	**.9427 ± .0443**
CPU	.0241 ± .0413	**.0212 ± .0301**	.9866 ± .0187	**.9971 ± .0063**
BCC	.2853 ± .0592	**.2496 ± .0485**	.6945 ± .0455	**.7349 ± .0692**
MPG	.0551 ± .0160	.0551 ± .0160	. 9855 ± .0108	.9855 ± .0108
ESL	.0658 ± .0221	**.0542 ± .0218**	.9755 ± .0160	**.9766 ± .0150**
MMG	.1628 ± .0187	**.1584 ± .0251**	.8966 ± .0162	**.9135 ± .0233**
ERA	.2899 ± .0191	**.2813 ± .0280**	**.7687 ± .0261**	.7670 ± .0290
LEV	.1370 ± .0162	**.1314 ± .0176**	**.9140 ± .0124**	.9122 ± .0202
CEV	.0273 ± .0089	.0273 ± .0089	.9959 ± .0027	.9959 ± .0027

improved performance most of the time. For the "full" CR, which is the most flexible variant, there is obviously a risk to overfit the training data and hence generalize worse.

Moreover, we also combined CR with the complexity reduction method proposed in Sect. 6. In addition to the average performance, the results in Table 7 also show the typical values of k as determined by this method (namely the most frequently chosen one). As can be seen, the method is indeed effective in the sense that the order of the fuzzy measure is often significantly reduced without compromising performance. On the other hand, in terms of performance, this method is still not competitive with using an optimal (cross-validated) k. This is not surprising, since the k determined by our complexity reduction method is only an upper bound (and learned in an unsupervised instead of a supervised manner).

Table 7 Performance in terms the average 0/1 loss and AUC ± standard deviation for CR using complexity reduction ($\epsilon = \delta = 0.1$). From top to bottom: 20 %, 50 %, and 80 % training data

data set	k	0/1 loss	AUC
DBS	4	.2286 ± .0549	.9235 ± .0489
CPU	4	.0998 ± .0347	.9664 ± .0227
BCC	3	.2888 ± .0578	.6193 ± .0406
MPG	4	.0719 ± .0108	.9787 ± .0067
ESL	3	.0737 ± .0103	.9663 ± .0049
MMG	3	.1761 ± .0107	.8857 ± .0174
ERA	4	.2981 ± .0158	.7579 ± .0103
LEV	4	.1526 ± .0146	.8984 ± .0103
CEV	6	.0448 ± .0089	.9825 ± .0080
DBS	4	.1944 ± .0631	.9338 ± .0368
CPU	4	.0361 ± .0432	.9902 ± .0139
BCC	3	.2838 ± .0448	.6232 ± .0374
MPG	4	.0570 ± .0080	.9812 ± .0044
ESL	3	.0727 ± .0148	.9740 ± .0077
MMG	3	.1667 ± .0130	.8976 ± .0087
ERA	4	.2930 ± .0162	.7641 ± .0146
LEV	4	.1421 ± .0142	.9088 ± .0132
CEV	6	.0376 ± .0059	.9912 ± .0024
DBS	4	.1939 ± .0615	.9381 ± .0471
CPU	4	.0244 ± .0531	.9962 ± .0090
BCC	3	.2755 ± .0404	.7142 ± .0507
MPG	4	.0597 ± .0126	.9832 ± .0057
ESL	3	.0603 ± .0236	.9769 ± .0146
MMG	3	.1620 ± .0250	.9001 ± .0202
ERA	4	.2899 ± .0191	.7687 ± .0261
LEV	4	.1370 ± .0162	.9140 ± .0124
CEV	6	.0273 ± .0089	.9959 ± .0027

Table 8 Average values of the scaling parameter γ in the choquistic regression model

DBS	CPU	BCC	MPG	ESL	MMG	ERA	LEV	CEV
36.69	691.81	15.30	23.87	45.12	19.05	8.07	15.13	69.23

7.3.3 Model interpretation

As mentioned earlier, one may expect a close connection between the scaling parameter γ in the choquistic model and the prediction accuracy of the model. More specifically, the better the model performs on a particular data set, the higher γ is expected to be. It is worth mentioning that our experimental results are in perfect agreement with this expectation. Indeed, comparing the ranking of the nine data sets in terms of accuracy and in terms of the average values of γ (shown in Table 8), we obtain a (Kendall tau) correlation of more than 0.8 throughout.

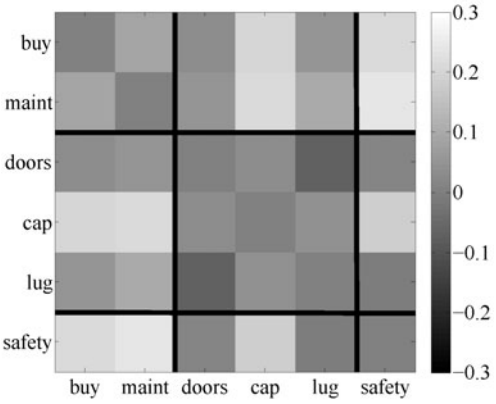

Fig. 3 Visualization of the interaction index for the car evaluation data (numerical values are shown in terms of level of *gray*, values on the diagonal are set to 0). Groups of related criteria are indicated by the *black lines*

As one of its key features, the Choquet integral offers interesting information about the importance of individual attributes as well as the interaction between them; this aspect was highlighted in Sect. 3.2. In fact, in many practical applications, this type of information is at least as important as the prediction accuracy of the model. A detailed analysis of this type of information is difficult and beyond the scope of this paper. Instead, we just give a few examples showing the plausibility of the results.

Regarding the Shapley index that measures the importance of individual attributes, the (average) values on the Car MPG data are as follows: cylinders ≈ 0.13, displacement ≈ 0, horsepower ≈ 0.25, weight ≈ 0.46, acceleration ≈ 0.03, model year ≈ 0.13, origin ≈ 0. In terms of attribute importance, this conveys the following picture:

$$\text{weight} > \text{horsepower} > \text{cylinders} \mid \text{model year} > \text{acceleration} > \text{displacement} \mid \text{origin}$$

Recalling the meaning of the data set, these weights should reflect the influence on the fuel consumption, and seen from this point of view, they appear to be fully plausible.

For the CPU data, the following Shapley values are obtained: machine cycle time in nanoseconds ≈ 0.07, minimum main memory in kilobytes ≈ 0.24, maximum main memory in kilobytes ≈ 0.30, cache memory in kilobytes ≈ 0.20, minimum channels in units ≈ 0.10, maximum channels in units ≈ 0.09. Thus, the most important properties are those concerning the memory (main and cache). The influence of the other properties (channels, cycle time) is not as strong, although they are not completely unimportant either.

Apart from the importance of individual attributes, it is interesting to look at the interaction between different attributes. As an example, Fig. 3 provides a visualization of the pairwise interaction between attributes for the car evaluation data, for which CR performs significantly better than LR. Recall that, in this data set, the evaluation of a car (output attribute) depends on a number of criteria, namely (a) buying price, (b) price of the maintenance, (c) number of doors, (d) capacity in terms of persons to carry, (e) size of luggage boot, (f) safety of the car. These criteria form a natural hierarchy, according to which the data was produced (Bohanec and Rajkovic 1990): (a) and (b) form a subgroup *price*, whereas the other properties are of a *technical* nature and can be further decomposed into *comfort* (c–e) and *safety* (f). Interestingly, the interaction in our model nicely agrees with this hierarchy or, stated differently, allows for recovering this hierarchy from the pairwise interactions between attributes: Interaction within each subgroup tends to be smaller (as can be seen from the darker colors) than interaction between criteria from different sub-

groups, suggesting a kind of redundancy in the former and complementarity in the latter case.

8 Concluding remarks

In this paper, we have advocated the use of the discrete Choquet integral as an aggregation operator in machine learning, especially in the context of learning monotone models. Apart from combining monotonicity and flexibility in a mathematically sound and elegant manner, the Choquet integral offers measures for quantifying the importance of individual predictor variables and the interaction between groups of variables, thereby providing important information about the relationship between independent and dependent variables.

We have analyzed several properties of the Choquet integral that appear to be interesting from a machine learning point of view, notably its capacity in terms of the VC-dimension. Moreover, we have addressed the issue of complexity reduction or, more specifically, the restriction of the Choquet integral to k-additive measures. In this regard, we have proposed a method for finding a suitable value of k.

As a concrete machine learning application of the Choquet integral, we have proposed a generalization of logistic regression, in which the Choquet integral is used for modeling the log odds of the positive class. First experimental studies have shown that this method, called choquistic regression, compares quite favorably with other methods. We like to mention again, however, that an improvement in prediction accuracy should not be seen as the only goal of monotone learning. Instead, the adherence to monotonicity constraints is often an important prerequisite for the acceptance of a model by domain experts.

Compared to standard logistic regression, the benefits of choquistic regression are coming at the expense of an increased computational complexity of the underlying learning algorithm, which solves a maximum likelihood estimation problem. This is mainly caused by the large number of parameters of the fuzzy measure on which the Choquet integral is based, and the complicated dependency between these parameters. In Hüllermeier and Fallah Tehrani (2012a), first steps aiming at a reduction of this complexity are made. Nevertheless, speeding up choquistic regression and making it scalable toward data sets with many attributes is an important topic of ongoing and future work.

Needless to say, the Choquet integral can be combined with machine learning methods other than logistic regression. Moreover, its use is not restricted to (binary) classification. In fact, we are quite convinced of its high potential in machine learning in general, and we are looking forward to exploring this potential in greater detail.

Acknowledgements This work was supported by the German Research Foundation (DFG). Ali Fallah Tehrani, Weiwei Cheng, and Eyke Hüllermeier were supported by the German Research Foundation. Krzysztof Dembczyński was supported by the German Research Foundation and the Polish Ministry of Science and Higher Education.

References

Angilella, S., Greco, S., & Matarazzo, B. (2009). Non-additive robust ordinal regression with Choquet integral, bipolar and level dependent Choquet integrals. In *Proceedings of the joint 2009 international fuzzy systems association world congress and 2009 European society of fuzzy logic and technology conference*. IFSA/EUSFLAT (pp. 1194–1199).

Beliakov, G. (2008). Fitting fuzzy measures by linear programming. Programming library fmtools. In *Proc. FUZZ-IEEE 2008, IEEE international conference on fuzzy systems*, Piscataway, NJ (pp. 862–867).

Beliakov, G., & James, S. (2011). Citation-based journal ranks: the use of fuzzy measures. *Fuzzy Sets and Systems*, *167*(1), 101–119.

Ben-David, A. (1995). Monotonicity maintenance in information-theoretic machine learning algorithms. *Machine Learning*, *19*, 29–43.

Ben-David, A., Sterling, L., & Pao, Y. H. (1989). Learning and classification of monotonic ordinal concepts. *Computational Intelligence*, *5*(1), 45–49.

Bohanec, M., & Rajkovic, V. (1990). Expert system for decision making. *Sistemica*, *1*(1), 145–157.

Chandrasekaran, R., Ryu, Y., Jacob, V., & Hong, S. (2005). Isotonic separation. *INFORMS Journal on Computing*, *17*, 462–474.

Choquet, G. (1954). Theory of capacities. *Annales de L'Institut Fourier*, *5*, 131–295.

Daniels, H., & Kamp, B. (1999). Applications of mlp networks to bond rating and house pricing. *Neural Computation and Applications*, *8*, 226–234.

Dembczyński, K., Kotłowski, W., & Słowiński, R. (2006). Additive preference model with piecewise linear components resulting from dominance-based rough set approximations. In *Lecture notes in computer science: Vol. 4029. International conference on artificial intelligence and soft computing 2006* (pp. 499–508).

Dembczyński, K., Kotlowski, W., & Slowinski, R. (2009). Learning rule ensembles for ordinal classification with monotonicity constraints. *Fundamenta Informaticae*, *94*(2), 163–178.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Duivesteijn, W., & Feelders, A. (2008). Nearest neighbour classification with monotonicity constraints. In *Lecture notes in computer science: Vol. 5211. Machine learning and knowledge discovery in databases* (pp. 301–316). Berlin: Springer.

Fallah Tehrani, A., Cheng, W., Dembczynski, K., & Hüllermeier, E. (2011). Learning monotone nonlinear models using the Choquet integral. In *Proceedings ECML/PKDD–2011, European conference on machine learning and principles and practice of knowledge discovery in databases*, Athens, Greece.

Feelders, A. (2010). Monotone relabeling in ordinal classification. In *Proceedings of the 10th IEEE international conference on data mining* (pp. 803–808). Washington: IEEE Computer Society.

Grabisch, M. (1995a). Fuzzy integral in multicriteria decision making. *Fuzzy Sets and Systems*, *69*(3), 279–298.

Grabisch, M. (1995b). A new algorithm for identifying fuzzy measures and its application to pattern recognition. In *Proceedings of IEEE international conference on fuzzy systems* (Vol. 1, pp. 145–150). New York: IEEE.

Grabisch, M. (1997). k-order additive discrete fuzzy measures and their representation. *Fuzzy Sets and Systems*, *92*(2), 167–189.

Grabisch, M. (2003). Modelling data by the Choquet integral. In *Information fusion in data mining* (pp. 135–148). Berlin: Springer.

Grabisch, M., & Nicolas, J. M. (1994). Classification by fuzzy integral: performance and tests. *Fuzzy Sets and Systems*, *65*(2–3), 255–271.

Grabisch, M., Murofushi, T., & Sugeno, M. (Eds.) (2000). *Fuzzy measures and integrals: theory and applications*. Heidelberg: Physica.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10–18.

Hosmer, D., & Lemeshow, S. (2000). *Applied logistic regression* (2nd ed.). New York: Wiley.

Hüllermeier, E., & Fallah Tehrani, A. (2012a). Efficient learning of classifiers based on the 2-additive Choquet integral. In *Computational intelligence in intelligent data analysis. Studies in computational intelligence*. Springer, forthcoming.

Hüllermeier, E., & Fallah Tehrani, A. (2012b). On the VC dimension of the Choquet integral. In *IPMU–2012, 14th international conference on information processing and management of uncertainty in knowledge-based systems*, Catania, Italy.

Jaccard, J. (2001). *Interaction effects in logistic regression*. Newbury Park: Sage Publications.

Kotłowski, W., Dembczyński, K., Greco, S., & Słowiński, R. (2008). Stochastic dominance-based rough set model for ordinal classification. *Information Sciences*, *178*(21), 3989–4204.

Landwehr, N., Hall, M., & Frank, E. (2003). Logistic model trees. In *Proceedings of the 14th European conference on machine learning* (pp. 241–252). Berlin: Springer.

Lee, S., Lee, H., Abbeel, P., & Ng, A. (2006). Efficient L1 regularized logistic regression. In *Proceedings of the 21st national conference on artificial intelligence* (pp. 401–408). Menlo Park: AAAI.

Modave, F., & Grabisch, M. (1998). Preference representation by a Choquet integral: commensurability hypothesis. In *Proceedings of the 7th international conference on information processing and management of uncertainty in knowledge-based systems* (pp. 164–171). Paris: Editions EDK.

Mori, T., & Murofushi, T. (1989). An analysis of evaluation model using fuzzy measure and the Choquet integral. In *Proceedings of the 5th fuzzy system symposium* (pp. 207–212). Japan Society for Fuzzy Sets and Systems.

Murofushi, T., & Soneda, S. (1993). Techniques for reading fuzzy measures (III): interaction index. In *Proceedings of the 9th fuzzy systems symposium* (pp. 693–696).

Potharst, R., & Feelders, A. (2002). Classification trees for problems with monotonicity constraints. *ACM SIGKDD Explorations Newsletter*, *4*(1), 1–10.

Sill, J. (1998). Monotonic networks. In *Advances in neural information processing systems* (pp. 661–667). Denver: MIT Press.

Sperner, E. (1928). Ein Satz über Untermengen einer endlichen Menge. *Mathematische Zeitschrift*, *27*(1), 544–548.

Sugeno, M. (1974). *Theory of fuzzy integrals and its application*. Ph.D. thesis, Tokyo Institute of Technology.

Tibshirani, R. J., Hastie, T. J., & Friedman, J. (2001). *The elements of statistical learning: data mining, inference, and prediction*. Berlin: Springer.

Torra, V. (2011). Learning aggregation operators for preference modeling. In *Preference learning* (pp. 317–333). Berlin: Springer.

Torra, V., & Narukawa, Y. (2007). *Modeling decisions: information fusion and aggregation operators*. Berlin: Springer.

Valiant, L. (1984). A theory of the learnable. *Communications of the ACM*, *27*(11), 1134–1142.

Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley.

Vitali, G. (1925). Sulla definizione di integrale delle funzioni di una variabile. *Annali Di Matematica Pura Ed Applicata*, *2*(1), 111–121.