Knowledge-Based Systems 112 (2016) 14–25

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Multivariate decision trees with monotonicity constraints

Shenglei Pei [a,b], Qinghua Hu [a,*], Chao Chen [b]

[a] School of Computer Science and Technology, Tianjin University, Tianjin, China
[b] School of Physics and Electronic Information Engineering, Qinghai University for Nationalities, Xining, China

ARTICLE INFO

Article history:
Received 6 April 2016
Revised 24 July 2016
Accepted 27 August 2016
Available online 29 August 2016

Keywords:
Monotone classification
Multivariate decision trees
Monotonicity constraints
Incomparable sample pairs

ABSTRACT

Classification with monotonicity constraint is a fundamental task in social analysis, management and decision making, where a monotonic function guarantees that objects with better feature values are not assigned with worse decisions. In practice, an object may be better than another on some attributes, while worse on others. These objects are considered to be incomparable. Incomparable object pairs limit the performance of a monotone classifier. In this work, we design an algorithm to combat this issue by constructing multivariate decision trees with monotonicity constraints (MMT). The classification model is naturally deemed as an oblique tree as it discovers partitions via oblique hyperplane in the input space. Our algorithm generates the projections of the objects which are used to split the data by improved splitting criteria with rank mutual information (RMI) or rank Gini impurity (RGI). Moreover, an improved algorithm with L_1-regularization is also proposed to compute the optimal subsets of features in the process of constructing the trees, which leads to a more compact tree. Experimental results show that the proposed algorithm improves the classification performance in monotone classification tasks. The proposed algorithm is also effective even if data is contaminated by non-monotonic noisy samples.

1. Introduction

In recent years, monotone classification has attracted more and more attention in the area of machine learning and decision analysis. As a matter of fact, monotone classification tasks widely exist in various fields, such as social science [1,2], medical treatment [3,4], fault diagnosis [5,6] and credit rating [7,8] etc. Fig. 1 shows an example, the loan applicants $\mathbf{x}_3$ and $\mathbf{x}_2$ have the same credit level, but $\mathbf{x}_3$ has a higher income than $\mathbf{x}_2$. Thus $\mathbf{x}_3$ should get more loan than $\mathbf{x}_2$. In these learning tasks, monotone classification functions guarantee monotonicity between attributes and class labels, which is a kind of ordinal structure. That is, the increasing values of attributes should not lead to decreasing class labels. Monotone classification is a kind of common tasks, which can increase the expressive power of decision models allowing the presentation of semantic concepts for preference, rank and importance, etc.

Generally speaking, there are two groups of algorithms for monotone classification: nonlinear function models and rule based monotone models. Some nonlinear models have been proposed to solve the tasks of monotone classification. Monotone neural networks were proposed by Daniels et al. [9,10]. The monotonicity of these neural networks is kept by the constraints of non-negative weights during the training phase. Pelckmans et al. [11] studied the monotonic kernel regression and then developed a monotonic LS-SVM regression model. However, these methods have some drawbacks. Daniels'method assumes that the tasks are strictly monotonic. Pelckmans'method is designed for ordinal regression problems. These algorithms generate complex boundaries to extract knowledge, which may be hardly comprehensible for domain experts.

The rule-based methods have been widely applied to the monotone classification tasks these years as it can generate comprehensible rules to aid decision making, including the kNN algorithm with monotonicity constraints, the dominance-based rough set approach (DRSA) and the monotone decision tree algorithms. The kNN algorithm with monotonicity constraints was proposed by Duivesteijn and Feelders [12]. Their algorithm improves the prediction performance over the standard k-nearest neighbor classifier and predicts the class labels by using a modified nearest neighbor rule. However, this algorithm applies a relabeling technique to the data, which may refute the nature of monotonicity. DRSA, proposed by Greco et al. [13,14] is applied to the monotonic tasks After that a collection of extended models based on DRSA was proposed [15,16]. These methods extract a set of comprehensible decision rules from the comparable samples.

Decision tree is different from DRSA. Its structure is simple and interpretable. And it has also been generalized to monotone classification [7,17,18]. Monotone decision tree algorithm was put for-

* Corresponding author. Fax: +862227401839.
E-mail address: huqinghua@tju.edu.cn (Q. Hu).

http://dx.doi.org/10.1016/j.knosys.2016.08.023
0950-7051/

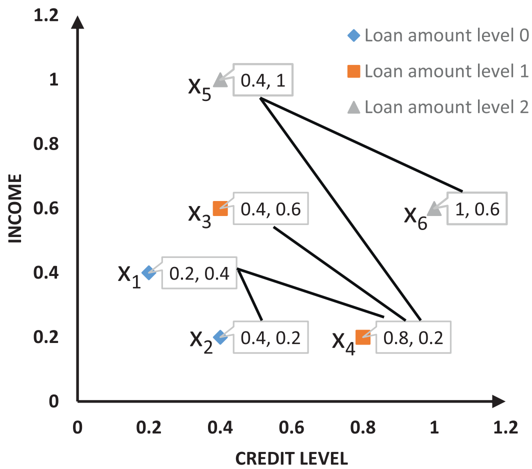

Fig. 1. Monotonic and incomparable samples pairs, where the endpoints of the solid line are the incomparable sample pair.

ward by Makino et al. [19] and later by Feelders and Pardoel [20] . However, their models are sensitive to noisy data, and the performance will reduce when applied to non-monotone data. In 2010, Hu et al. proposed the definition of rank mutual information (RMI) to evaluate monotone classification [21], and designed a monotone decision tree algorithm based on RMI [22]. In 2015, inspired by the idea in [21], Marsala and Petturiti proposed rank Gini impurity (RGI) [23]. Monotone decision trees with RMI or RGI improve the classification performance of noisy tasks. It has been shown that the extracted decision rules are reasonable and comprehensible in the comparable samples. Recently, some ensemble algorithms were also proposed to adapt the problem with monotonicity constraints [24,25]. They improve the predictive performance by some simple ensemble mechanisms and pruning technique, but they do not solve the issue of incomparable sample pairs.

The statistics show that more than 80% sample pairs are incomparable [26] in practical monotonic classification tasks. These incomparable sample pairs limit the application of monotone decision trees. In Fig. 1, one can see that the loan applicant $\mathbf{x}_5$ has a higher income and a lower credit level than the loan applicant $\mathbf{x}_4$. Thus, the applicants $\mathbf{x}_4$ and $\mathbf{x}_5$ are incomparable in terms of these two attributes. That is, $\mathbf{x}_4$ and $\mathbf{x}_5$ cannot be inferred who should assign more loans. Nevertheless, the aforementioned algorithms basically deal with monotonicity constraints in the comparable samples, except that the kNN algorithm with monotonicity constraint may impair natural property of the original dataset as a form of relabeling in classification. In these approaches, the potential benefit of the incomparable pairs has not been further studied. In fact, we know that we sometimes do not compare objects on each feature, but compare them on the locally weighted sum of the feature values. This kind of monotonic classification can be implemented by a set of linear monotone functions. Moreover, decision trees are easily comprehensible for domain experts. Therefore, we intend to propose a new monotone decision tree model to solve the incomparable issues of the samples. In this study, linear combinations are used to represent the relationship among the multiple attributes of the incomparable samples. Thus, we transform incomparable sample pairs into comparable sample pairs by using linear functions. Furthermore, the linear discriminant method is integrated into the multivariate decision tree, which can extract comprehensible decision rules from the data. Li and Sweigart et al. introduced decision trees based on linear discriminants and tabu search [27]. Fisher's decision tree was proposed by López–Chau et al. [28], which is easily comprehensible. The decision trees mentioned above and the related literature have shown that the linear discriminant method splits data by using several attributes of different weights. And the splitting hyperplane can have an arbitrary direction. However, these linear discriminant methods do not guarantee monotonicity.

Based on the points discussed above, we propose an algorithm to construct multivariate decision trees with monotonicity constraints (MMT), whose splitting hyperplane is oblique. In Fig. 1, one can see that the incomparable samples ($\mathbf{x}_4$, $\mathbf{x}_5$) may be split by an oblique line. In our case, we introduce the linear discriminant function to solve incomparable issues in monotone classification tasks. Our algorithms generate linear discrimination function by using the nonnegative least-squares method [29–32] and also generate the best split by using RMI and RGI.

We present the design, implementation and evaluation of multivariate decision trees with monotonicity constraints. The main differences of our algorithms compared with other monotone classification algorithms are summarized as follows: (1) This algorithm solves the incomparable issue in monotone classification tasks. (2) Our algorithm is robust to the non-monotone noisy data. (3) We apply L_1-regularization to improve the MMT algorithm for training compact decision tree.

The rest of this paper is organized as follows. In Section 2, we review some related studies. In Section 3, the proposed MMT algorithm and MMT-L_1 algorithms are presented. In Section 4, we describe some experiments on both artificial and real-world datasets to demonstrate the effectiveness of the proposed algorithms. Finally, we give the conclusions and future work in Section 5.

2. Related work

In this section, we review the related definitions and notions of monotonic classification and decision trees with monotonicity constraints.

2.1. Incomparable sample pairs of monotonic classification

Let $U = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a set of objects and $A = \{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$ be a set of attributes to describe the objects. Y is a set of class labels and y_i is the class label of $\mathbf{x}_i$ where $y_i \in \{\tilde{y}_1, \ldots, \tilde{y}_c\}$. We assume all attributes and class labels are ordinal values.

Given a set of objects U, $\forall \mathbf{x} \in U$ and $B \subseteq A$, where $B = \{\mathbf{a}_1, \ldots, \mathbf{a}_{m'}\}$. Let $\mathbf{a}_k(\mathbf{x})$ be the attribute value of sample $\mathbf{x}$ on $\mathbf{a}_k$. The ordinal relations between samples in terms of attribute $\mathbf{a}_k$ or Y is denoted by $\leqslant$. Thus, the partial ordering $\preccurlyeq$ on U is defined as

$$\mathbf{x}_i \preccurlyeq_{\mathbf{B}} \mathbf{x}_j \iff \mathbf{a}_k(\mathbf{x}_i) \leqslant \mathbf{a}_k(\mathbf{x}_j), \ \text{for } k = 1, \ldots, m'. \tag{1}$$

Then, the samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are *comparable in term of B if we have* either $\mathbf{x}_i \preccurlyeq_{\mathbf{B}} \mathbf{x}_j$ or $\mathbf{x}_j \preccurlyeq_B \mathbf{x}_i$; otherwise, they are *incomparable.* In Fig. 1, one can see that the loan applicant $\mathbf{x}_5$ has a higher income and a lower credit level than the loan applicant $\mathbf{x}_4$. Thus, the applicants $\mathbf{x}_4$ and $\mathbf{x}_5$ are incomparable in terms of income and credit.

In most real-world classification tasks, the attributes may be monotonic to class labels [33]. For $\forall \mathbf{a}_k \in B$, we have $\mathbf{a}_k(\mathbf{x}_i) \geqslant \mathbf{a}_k(\mathbf{x}_j)$, then we say $\mathbf{x}_i$ is not worse than $\mathbf{x}_j$ regarding B and denote this by $\mathbf{x}_i \succeq_B \mathbf{x}_j$.

Given that $\mathbf{x}_i \in U$, $B \subseteq A$, we associate $\mathbf{x}_i$ with the following sets

$$[\mathbf{x}_i]_B^{\leqslant} = \{\mathbf{x}_j \in U \mid \mathbf{x}_j \preccurlyeq_B \mathbf{x}_i\}, \tag{2}$$
$$[\mathbf{x}_i]_Y^{\leqslant} = \{\mathbf{x}_j \in U \mid \mathbf{x}_j \leqslant_Y \mathbf{x}_i\}.$$

Definition 1. Given a set of objects U, A is the set of attributes, $B \subseteq A$. We say the decisions of U is monotonically consistent in terms of B, if $\forall \mathbf{x}_i, \mathbf{x}_j \in U$,

$$\mathbf{x}_i \preccurlyeq_B \mathbf{x}_j \Rightarrow Y(\mathbf{x}_i) \leqslant Y(\mathbf{x}_j) \ \text{or} \ \mathbf{x}_i \succcurlyeq_B \mathbf{x}_j \Rightarrow Y(\mathbf{x}_i) \geqslant Y(\mathbf{x}_j); \tag{3}$$