Computers and Electrical Engineering 40 (2014) 1429–1438

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

A study on relationships between heuristics and optimal cuts in decision tree induction ☆

Hong-Yan Ji [a], Xi-Zhao Wang [b,*], Yu-Lin He [b,*], Wen-Liang Li [b]

[a] Dept. of Mathematics, Hebei University of Engineering, Handan, China
[b] Key Lab. of Machine Learning and Computational Intelligence, College of Mathematics and Computer Science, Hebei University, Baoding, China

ARTICLE INFO

Article history:
Available online 24 December 2013

ABSTRACT

Cut selection based on heuristic information is one of the most fundamental issues in the induction of decision trees with continuous valued attributes. This paper connects the selection of optimal cuts with a class of heuristic information functions together. It statistically shows that both training and testing accuracies in decision tree learning are dependent strongly on the selection of heuristics. A clear relationship between the second-order derivative of heuristic information function and locations of optimal cuts is mathematically derived and further is confirmed experimentally. Incorporating this relationship into a process of building decision trees, we can significantly reduce the number of detected cuts and furthermore improve the generalization of the decision tree.

1. Introduction

Decision tree learning [1–4] is one of the most widely used and practical methods for inductive inference, which is considered as an approach to approximating discrete-valued target functions. Classical decision tree learning algorithms including Concept Learning System (CLS), ID3 [5,6], C4.5 [7,8] and classification and regression trees (CART) [9,10], etc. are typically constructed top-down and their building process is recursive. How to choose extended attributes for growing the tree is most crucial issue in building decision trees.

The attributes used to describe samples can be continuous or discrete. Usually the discrete attributes are non-ordered nominal values [11]. In the process of building a decision tree, discrete-valued attributes partition the space of examples by their nominal values, while continuous-valued attributes usually partition the space of examples by cut points, in short, cuts. How to choose optimal cuts based on heuristic information functions is one of the most fundamental issues of constructing decision trees with continuous-valued attributes. It is well known that the number of detected cuts largely determines the efficiency of a decision tree generation algorithm. Usually for a decision tree generation algorithm, all candidate cuts need to be detected and then the optimal one is selected. Although most decision tree algorithms are able to handle continuous-valued attributes well, the large range of cut detection seriously limits the efficiency of the algorithms [12,13]. To overcome the defect of evaluating all cuts, Fayyad and Irani in [12] presented a method that generates decision trees using the information entropy minimization heuristic via discretizing continuous-valued attributes and proved that optimal cuts are always boundaries. To some extent, Fayyad's method narrows the detection range down to boundary points and furthermore improves the efficiency of cut selection. Wang and Hong in [14] extended Fayyad's method to the generation of decision trees with interval-valued attributes and proved that the points of minimum information entropy of partitioning are always unstable cuts. Furthermore, Yen and Chu in [15] proposed relaxation of instance boundaries

☆ Reviews processed and recommended for publication to Editor-in-Chief by Guest Editor Dr. Zhihong Man.
* Corresponding authors. Tel.: +86 15230420378 (Y.-L. He).
E-mail addresses: xizhaowang@ieee.org (X.-Z. Wang), csylhe@gmail.com (Y.-L. He).

0045-7906/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compeleceng.2013.11.030

(RIB) based on Fayyad's method in order to eliminate situations where a single isolated instance is located amidst samples of another class. RIB further narrowed the detection range of cuts.

For most algorithms of building decision trees with continuous-valued attributes, their heuristic information functions are based on information entropy. This paper makes an attempt to investigate whether other heuristic information functions have the similar properties. Several kinds of heuristic information functions are discussed and a generalized form of heuristic information functions is proposed. Whether the generalized heuristic information function has the property that the optimal cuts are always located in boundaries is studied. A clear relationship between the second-order derivative of heuristic information function and locations of optimal cuts is established. An investigation to this relationship leads to a significant reduction of number of detected cuts and further an improvement of generalization for a decision tree.

The rest of this paper is organized as follows. In Section 2 we present the process of selecting optimal cuts. Section 3 introduces several typical heuristic information functions. In Section 4 we establish a relationship between heuristic information functions and optimal cut selection. Section 5 concludes this paper.

2. Optimal cut selection

Cut selection is a key step of generating decision trees with continuous-valued attributes. Cut point is a threshold value T. For a continuous-valued attribute A, all instances with $A \leqslant T$ are assigned to a sub-interval while all instances with $A > T$ are assigned to the other one. In this way, a continuous-valued attribute is discretized by using a cut to split its range into two intervals. A typical process of optimal cut selection includes four steps: sorting, getting cuts, evaluating cuts and splitting node [16].

(1) Sorting: All individual values of a continuous attribute are sorted in either descending or ascending order.
(2) Getting cuts: Generally, the midpoint between two adjacent samples in the sorted sequence is evaluated as a potential cut point. Assuming that all samples have distinct values $v_1, v_2, \ldots, v_N$, there are $N-1$ candidate cuts to be evaluated.
(3) Evaluating cuts: Evaluating $N-1$ candidate cuts based on a certain evaluation function for determining which cut is the optimal one. One can find numerous evaluation functions in the literature such as information gain, Gini-index and classification error.
(4) Splitting node: Splitting the range of continuous values into two intervals according to the optimal cut.

Repeating above operations for each node that has the continuous-valued attributes until a stopping criterion is met. The optimal cut selection is the main workload of generating a decision tree. Usually detecting all candidate cuts for a continuous attribute is time-consuming (although the decision tree generation algorithm has the computational complexity much less than other types of learning algorithms). The number of candidate cuts (we must evaluate) directly determines the efficiency of a decision tree learning algorithm.

3. Heuristic information functions

At each node to be extended, the extended attribute we choose is most beneficial for classifying samples. The selected heuristic information function plays an essential role in the process of extending a node. From literatures one can find many heuristic information functions and most of them are impurity-based functions [6–10]. Here, we review some of the most representative ones.

3.1. Information entropy

Quinlan proposed the ID3 decision tree algorithm in 1986 using information gain [5] and later the C4.5 algorithm in 1993 using gain ratio [7]. Each of the two heuristic functions is based on the information entropy. The information entropy, which measures the impurity of instances in a node with respect to the classes, is defined as

$$Entropy(S) = -\sum_{i=1}^{k} p(C_i|S)\log_2 p(C_i|S), \tag{1}$$

where S is a data set (i.e., a node) to be extended, k is the number of classes and $p(C_i|S)$ is the probability of an example belonging to class C_i in the data set S. When all examples in S belong to the same class, the entropy is zero. When the probabilities of an example belonging to different classes are identical, the entropy reaches its maximum $\log_2 k$.

3.2. Gini-index

One alternative measure that has been used successfully in generating decision trees is the Gini-index which was proposed by Breiman in CART [9] and employed in the following function:

$$Gini(S) = 1 - \sum_{i=1}^{k} p^2(C_i|S) = \sum_{i=1}^{k} p(C_i|S)(1 - p(C_i|S)). \tag{2}$$

The formula of Gini-index is quite similar to entropy. That is, Gini-index is zero if all examples in S belong to the same class and Gini-index reaches its maximum $1 - 1/k$ if all probabilities of an example belonging to different classes are equivalent.

3.3. Classification error

Classification error is also a kind of impurity-based measure. It is defined as follows:

$$Classification\ Error(S) = 1 - \max\{p(C_i|S)\}. \tag{3}$$

This measure is similar to the two above-mentioned ones. It gets its minimum zero when all examples belong to the same class and gets its maximum $1 - 1/k$ when data set S contains the same number of examples for each class.

3.4. Ambiguity

Yuan and Shaw [17] used ambiguity as the attribute selection criteria for fuzzy decision tree. Let $\pi = (\pi(x)|x \in X)$ denote a normalized possibility distribution on $X = \{x_1, x_2, \ldots, x_n\}$, the ambiguity measure is defined as

$$Ambiguity(Y) = \sum_{i=1}^{n} (\pi_i^* - \pi_{i+1}^*) \ln i,$$

where Y is a fuzzy variable and $\pi^* = \{\pi_1^*, \pi_2^*, \ldots, \pi_n^*\}$ is the permutation of the possibility distribution $\pi = \{\pi(x_1), \pi(x_2), \ldots, \pi(x_n)\}$, sorted so that $\pi_i^* \geqslant \pi_{i+1}^*$ for all $i = 1, 2, \ldots, n$, and $\pi_{i+1}^* = 0$. The ambiguity denoting a type of impurity can be re-written as

$$Ambiguity(Y) = \sum_{i=1}^{n} (p_i^* - p_{i+1}^*) \ln i, \tag{4}$$

where $p^* = \{p_1^*, p_2^*, \ldots, p_n^*\}$ is a sorting of probability distribution $p = \{p(x_1), p(x_2), \ldots, p(x_n)\}$, $p_i = \dfrac{p(C_i|S)}{\max\{p(C_i|S)\}}$, $p_i^* \geqslant p_{i+1}^*$ for all $i = 1, 2, \ldots, n$, and $p_{i+1}^* = 0$.

3.5. Generalized heuristic information function

Given a data set S with positive and negative examples of a target concept which is represented as a Boolean function. As we show below, each of these impurity measures of S relative to this Boolean classification can be calculated as

$$\begin{cases} Entropy(S) = -p(+|S)\log_2 p(+|S) - (1 - p(+|S))\log_2(1 - p(+|S)) \\ Gini(S) = 1 - p^2(+|S) - (1 - p(+|S))^2 = 2p(+|S)(1 - p(+|S)) \\ Classification\ Error(S) = \begin{cases} p(+|S) & 0.5 \leqslant p(+|S) \leqslant 1 \\ 1 - p(+|S) & 0 \leqslant p(+|S) \leqslant 0.5 \end{cases} \\ Ambiguity(S) = \begin{cases} \dfrac{p(+|S)}{1 - p(+|S)} \ln 2 & 0 \leqslant p(+|S) \leqslant 0.5 \\ \dfrac{1 - p(+|S)}{p(+|S)} \ln 2 & 0.5 \leqslant p(+|S) \leqslant 1 \end{cases} \end{cases},$$

where $p(+|S)$ is the proportion of positive examples in S. Fig. 1 shows the forms of the four heuristic information functions to a Boolean classification, where $p(+|S)$ varies between 0 and 1.

Observing the specific forms of the four heuristic information functions, we can find their similarities and differences as follows.

Their differences mainly reflect in two aspects:

(1) These functions have different derivability at point $p(+|S) = 0.5$: ambiguity and classification errors are derivable, entropy and Gini-index are non-derivable.
(2) The second-order derivatives of these functions are different in the derivable intervals: The second-order derivatives of entropy and Gini-index are less than zero, classification error's second-order derivative is equal to zero and ambiguity's second-order derivative is greater than zero.

Their similarities mainly reflect in another two aspects:

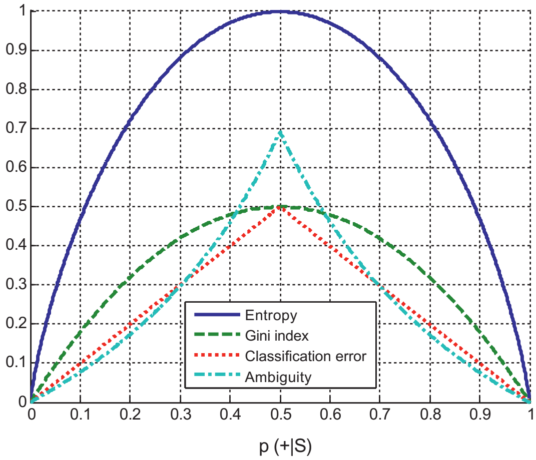

Fig. 1. Four heuristic information functions relative to a Boolean classification.

(1) These functions are symmetrical about point $p(+|S) = 0.5$, monotonically increasing in the left interval [0, 0.5], and monotonically decreasing in the right interval [0.5, 1]. They get their maximum at point $p(+|S) = 0.5$ and their minimum at points $p(+|S) = 0$ and $p(+|S) = 1$.
(2) These functions have the second-order derivatives in their derivable intervals (0, 0.5) and (0.5, 1).

Through summarizing these similarities and differences, we extract a general form of these functions as follows:

$$F(x) = \begin{cases} f(x) & 0 \leqslant x \leqslant \frac{1}{2} \\ f(1-x) & \frac{1}{2} \leqslant x \leqslant 1 \end{cases},$$

where x is the proportion of positive examples in S. $f(x)$ is such a function defined in [0, 1] with the properties: $f(0) = 0$, $f(x)$ is continuous and monotonically increasing in interval [0, 0.5], and its second-order derivative exists in interval (0, 0.5). It is worth noting that the function f defined above can be regarded as a general form of the 4 measures given in Section 3.

$F(x)$ is called the generalized heuristic information function which includes many existing heuristics such as the frequently used Entropy and Gini index as special cases. It is very practically useful for building decision trees to find a uniform (generalized) function summarizing diverse heuristics. The discussion about the generalized function is expected to make clear the impact of different heuristics on size of the generated decision tree. It is also to expected reveal some new key features of relations between the generalization capability and the size of a decision tree.

4. Relationship between generalized heuristic information function and optimal cut selection

For each node to be extended during the tree growing, we would like to select the attribute having the optimal cut as the extended attribute. Assume that S is the set of samples, representing the considered node, A is a candidate continuous-valued attribute, and T is a candidate cut of attribute A. When we use $F(x)$ as the heuristic function for decision tree generation, the cuts can be measured by using the following equation

$$G_S(T) = \frac{|S_1|}{|S|}F(p(+|S_1)) + \frac{|S_2|}{|S|}F(p(+|S_2)), \tag{5}$$

where set S is partitioned into the subsets S_1 and S_2 by cut T, $|\cdot|$ denotes the size of a data set, $p(+|S_1)$ and $p(+|S_2)$ are the proportions of positive examples in the two subsets (p_1 and p_2 for short).

For example, when $F(x)$ is the information entropy, $G_S(T)$ denotes the class information entropy of the partition induced by T:

$$G_S(T) = \frac{|S_1|}{|S|}Entropy(S_1) + \frac{|S_2|}{|S|}Entropy(S_2).$$

Most decision tree generation algorithms select the extended attribute with the highest information gain. Maximizing information gain is equivalent to minimizing the average class entropy. We will select the optimal cut with the smallest average class entropy for continuous-valued attributes based on the generalized information function. Fayyad and Irani [12,13] proved that the cuts with minimum average class entropy are always boundaries. This result can narrow the range of the optimal cuts detection from all candidate cuts to boundary cute only, and therefore, improve the computational efficiency of cut selection significantly. The key issue to be solved in this study is whether the optimal cuts based on generalized heuristic information function are also boundaries, that is, whether $G_S(T)$ gets its minimum at boundaries. We have the following proposition.

Proposition 4.1. *Suppose that $G_S(x)$ gets its minimum at $x = T$, then we have the following conclusions regarding the generalized heuristic information function and its optimal cuts*

(1) *If $f''(x) < 0$, then T must be a boundary point.*
(2) *If $f''(x) = 0$, then T can be a boundary point.*
(3) *If $f''(x) > 0$, then T must be a boundary or a point satisfying $p_1 = p_2$.*

Proof. Let S be a samples set belonging to node to be extended and A be a continuous-valued attribute regarding S. Sort the samples in S by increasing value of attribute A. Assume that cut T occurs within a sequence of n examples of the same class, where $n \geqslant 2$. Without loss of generality, we assume this class being positive. Let T_1 and T_2 be the boundary points of n examples, and S_1 and S_2 be the subsets of S divided by T. There are n_c examples that have values greater than T_1 and less than T, where $0 \leqslant n_c \leqslant n$. Fig. 2 describes this situation. □

The problem of judging the position of an optimal cut is converted to a problem of testing whether $G_S(T)$ gets its minimum at T_1 or T_2. If $G_S(T)$ gets its minimum at T_1 or T_2, then the optimal cut is a boundary; otherwise the optimal cut is not a boundary. Next, our main task is to obtain the position where $G_S(T)$ gets its minimum according to the values of $f''(x)$.

Let there be N^+ positive examples in S, nl examples in S with A-values less than T_1 where nl^+ examples are positive, and nr examples in S with A-values greater than T_1 where nr examples are positive. Noting that $0 \leqslant nl^+ \leqslant nl$, $0 \leqslant nr^+ \leqslant nr$, and $nl^+ + n + nr^+ = N^+$, p_1 and p_2 are the proportions of positive examples in S_1 and S_2 respectively, we have:

$$p_2 = \frac{N^+ - |S_1|p_1}{|S| - |S_1|}.$$

$G_S(T)$ can be written as a expression of p_1:

$$G_S(T) = \frac{|S_1|}{|S|}F(p_1) + \frac{|S_2|}{|S|}F(p_2) = \frac{|S_1|}{|S|}F(p_1) + \frac{|S_2|}{|S|}F\left(\frac{N^+ - |S_1|p_1}{|S| - |S_1|}\right). \quad (6)$$

While cut T moves from T_1 to T_2, p_1 increases and p_2 decreases monotonically. Because we cannot make sure the derivability on point 0.5, the change-intervals of p_1 can be divided into three kinds: the intervals are included within $(0, 0.5)$; the intervals include point 0.5 and the intervals are included within $(0.5, 1)$. The change-intervals of p_2 have the similar three cases.

Combining the change-intervals of p_1 and p_2, we can obtain the following 9 cases:

(1) The change-intervals of both p_1 and p_2 are contained within $(0, 0.5)$.
(2) p_1's change-interval is contained within $(0, 0.5)$, p_2's change-interval is contained within $(0.5, 1)$.
(3) p_1's change-interval is contained within $(0.5, 1)$, p_2's change-interval is contained within $(0, 0.5)$.
(4) The change-intervals of both p_1 and p_2 are contained within $(0.5, 1)$.
(5) The change-intervals of both p_1 and p_2 contain point 0.5.
(6) p_1's change-interval contains point 0.5, p_2's change-interval is contained within $(0, 0.5)$.
(7) p_1's change-interval contains point 0.5, p_2's change-interval is contained within $(0.5, 1)$.
(8) p_1's change-interval is contained within $(0, 0.5)$, p_2's change-interval contains point 0.5.
(9) p_1's change-interval is contained within $(0.5, 1)$, p_2's change-interval contains point 0.5.

Below we discuss each of the 9 cases respectively.

(1) The change-intervals of both p_1 and p_2 are included within $(0, 0.5)$.

In this case, we can rewrite the above expression as

$$G_S(T) = \frac{|S_1|}{|S|}f(p_1) + \frac{|S| - |S_1|}{|S|}f\left(\frac{N^+ - |S_1|p_1}{|S| - |S_1|}\right). \quad (7)$$

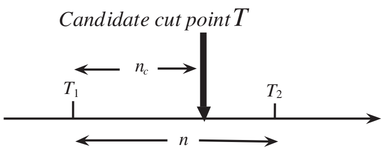

Fig. 2. A candidate cut *T*.

Taking the first derivative of $G_S(T)$ with respect to p_1, we have

$$\frac{d(G_S(T))}{dp_1} = \frac{|S_1|}{|S|}f'(p_1) + \frac{|S|-|S_1|}{|S|}f'\left(\frac{N^+ - |S_1|p_1}{|S|-|S_1|}\right)\frac{-|S_1|}{|S|-|S_1|} = \frac{|S_1|}{|S|}f'(p_1) - \frac{|S_1|}{|S|}f'\left(\frac{N^+ - |S_1|p_1}{|S|-|S_1|}\right) = \frac{|S_1|}{|S|}(f'(p_1) - f'(p_2)). \quad (8)$$

We now discuss the minimum value of $G_S(T)$ according to different values of the second-order derivative,

a. If $f''(x) < 0$, then $f(x)$ is monotonically decreasing with respect to x. If $p_1 < p_2$, we have $\frac{d(G_S(T))}{dp_1} = \frac{|S_1|}{|S|}(f'(p_1) - f'(p_2)) > 0$; if $p_1 > p_2$, we have $\frac{d(G_S(T))}{dp_1} = \frac{|S_1|}{|S|}(f'(p_1) - f'(p_2)) < 0$. It then results in the following assertions:
   (A1) If p_1 is permanently less than p_2, then $G_S(T)$ is monotonically increasing and gets its minimum at boundary point $n_c = 0$.
   (A2) If p_1 is permanently greater than p_2, then $G_S(T)$ is monotonically decreasing and gets its minimum at boundary point $n_c = n$.
   (A3) If p_1 is less than p_2 earlier and less than p_2 later, then $G_S(T)$ is monotonically increasing earlier and decreasing later, and then, gets its minimum at boundary points $n_c = 0$ or $n_c = n$.
   (A1)–(A3) imply that, no matter what relationship between p_1 and p_2, $G_S(T)$ gets its minimum value at boundaries.
b. If $f''(x) = 0$, then $f(x)$ is a constant, $\frac{d(G_S(T))}{dp_1} = 0$ for all $n_c = 0, 1, \ldots, n$. And therefore, no matter they are boundaries, $G_S(T)$ is a constant.
c. If $f''(x) > 0$, then $f(x)$ is monotonically increasing with respect to x. We have $\frac{d(G_S(T))}{dp_1} = \frac{|S_1|}{|S|}(f'(p_1) - f'(p_2)) < 0$ if $p_1 < p_2$ and $\frac{d(G_S(T))}{dp_1} = \frac{|S_1|}{|S|}(f'(p_1) - f'(p_2)) > 0$ if $p_1 > p_2$, which imply that $G_S(T)$ is monotonically decreasing and gets its minimum at boundary point $n_c = n$ when p_1 is permanently less than p_2; $G_S(T)$ is monotonically increasing and gets its minimum at boundary point $n_c = 0$ when p_1 is permanently greater than p_2; and $G_S(T)$ is monotonically decreasing earlier and increasing later, and gets its minimum at the point satisfying $p_1 = p_2$ when p_1 is grater than p_2 earlier and less than p_2 later. And therefore, we can see that $G_S(T)$ gets its minimum value at boundaries or the points satisfying $p_1 = p_2$.

(2) p_1's change-interval is included within (0,0.5) and p_2's change-interval is included within (0.5,1).

In this case, we can rewrite expression (6) as

$$G_S(T) = \frac{|S_1|}{|S|}f(p_1) + \frac{|S|-|S_1|}{|S|}f\left(1 - \frac{N^+ - |S_1|p_1}{|S|-|S_1|}\right). \quad (9)$$

Taking the first derivative of $G_S(T)$ with respect to p_1, we have

$$\begin{aligned}\frac{d(G_S(T))}{dp_1} &= \frac{|S_1|}{|S|}f'(p_1) + \frac{|S|-|S_1|}{|S|}f'\left(1 - \frac{N^+ - |S_1|p_1}{|S|-|S_1|}\right)\frac{|S_1|}{|S|-|S_1|} = \frac{|S_1|}{|S|}f'(p_1) + \frac{|S_1|}{|S|}f'\left(1 - \frac{N^+ - |S_1|p_1}{|S|-|S_1|}\right) \\ &= \frac{|S_1|}{|S|}(f'(p_1) + f'(1-p_2)) > 0, \end{aligned} \quad (10)$$

$\frac{d(G_S(T))}{dp_1} > 0$, for all $n_c = 0, 1, \ldots, n$, implies that $G_S(T)$ is monotonically decreasing and gets its maximum at boundary point $n_c = 0$.

(3) p_1's change-interval is included within (0.5,1) and p_2's change-interval is included within (0,0.5).
Similarly to the case (2), it is easy to verify case (3).

(4) The change-intervals of both p_1 and p_2 are included within (0.5,1).

In this case, we can rewrite expression (6) as

$$G_S(T) = \frac{|S_1|}{|S|}f(1-p_1) + \frac{|S|-|S_1|}{|S|}f\left(1 - \frac{N^+ - |S_1|p_1}{|S|-|S_1|}\right). \quad (11)$$

Taking the first derivative of $G_S(T)$ with respect to p_1, we have:

$$\begin{aligned}\frac{d(G_S(T))}{dp_1} &= -\frac{|S_1|}{|S|}f'(1-p_1) + \frac{|S|-|S_1|}{|S|}f'\left(1 - \frac{N^+ - |S_1|p_1}{|S|-|S_1|}\right)\frac{|S_1|}{|S|-|S_1|} = -\frac{|S_1|}{|S|}f'(1-p_1) + \frac{|S_1|}{|S|}f'\left(1 - \frac{N^+ - |S_1|p_1}{|S|-|S_1|}\right) \\ &= -\frac{|S_1|}{|S|}(f'(1-p_1) - f'(1-p_2)) = \frac{|S_1|}{|S|}(f'(p_1) - f'(p_2)).\end{aligned}$$

Noting that the expression above is the same to (8), we can obtain the same conclusion.

(5) The change-intervals of both p_1 and p_2 include point 0.5.

This combination is the most complex one. By dividing each of these change-intervals into two subintervals, the partition can be illustrated in Table 1. p_1 increases and p_2 decreases along with n_c changes from 0 to n, so p_1 changes from values less

Table 1
The change-intervals partition of p_1 and p_2.

Subintervals	p_1	p_2
State 1	(0,0.5)	(0,0.5)
State 2	(0,0.5)	(0.5,1)
State 3	(0.5,1)	(0,0.5)
State 4	(0.5,1)	(0.5,1)

than 0.5 to values greater than 0.5 and p_2 changes from values greater than 0.5 to values less than 0.5. The change varies from state2 to state3 through the middle process state (1) or state (4).

The monotonicity of p_1 and p_2 in these subintervals are listed in Table 2 which has been discussed in the first four combinations. Noting that $G_S(T)$ is continuous in its domain and the monotonicity of p_1 and p_2 in state 1 is same as in state 4, we can get the position of $G_S(T)$'s minimum:

(1) If $f''(x) < 0$, $G_S(x)$ gets its minimum value at one of the boundary points.
(2) If $f''(x) = 0$, $G_S(x)$ can gets its minimum value at more than one boundary point.
(3) If $f''(x) > 0$, $G_S(x)$ gets its minimum value at a boundary point or the point with $p_1 = p_2$.

The other cases can be verified similarly to cases (1)–(5). We now end the proof.[1]

The optimal cuts of information entropy and Gini-index are always on boundaries, while ambiguity gets its optimal cuts at boundaries or some special points. If the classification error function attains its minimum at non-boundaries, then it can attain the same minimum at boundaries. For a given heuristic information function, we can determine the positions of the optimal cuts according to the values of second-order derivatives. For example, if we use $\sin(x)$ as a heuristic information function, the optimal cuts are always boundaries because the second-order derivative of $\sin(x)$ in [0,1] is less than zero.

When the second-order derivative of the heuristic information function is greater than zero, optimal cut point may be a non-boundary point with $p_1 = p_2$.

$$p_{(+|S)} = \frac{p_1|S_1| + p_2|S_2|}{|S|} = p_1\frac{|S_1| + |S_2|}{|S|} = p_1. \tag{12}$$

Noting that

$$p_1 = \frac{nl^+ + n_c}{nl + n_c}, \tag{13}$$

we have

$$n_c = \frac{p_{(+|S)}nl - nl^+}{1 - p_{(+|S)}}. \tag{14}$$

If n_c (the number of the points between T_1 and T) satisfies $0 < n_c < n$, then there exits a non-boundary point such that $G_S(T)$ reaches its minimum.

5. Numerical experiments

5.1. Experimental setting

Nine data sets are selected from UCI machine learning repository [19], which has been extensively used in testing the performance of diversified kinds of classifiers, LIBSVM available at http://www.csie.ntu.edu.tw/cjlin/libsvm and ELENA dataset available via anonymous ftp: ftp.dice.ucl.ac.be in the directory pub/neural/ELENA/databases. The sizes of data sets are from 683 to 19,020. The information about 9 data sets is summarized in Table 3.

5.2. Experimental objectives

It is to verify whether a decision tree learning system depends on its selection of heuristic function. The selected heuristic functions are required to meet the conditions given in Section 4. We selected 4 generalized heuristic information functions for our comparison. They are

[1] *Note.* For particular engineering issue of building a binary decision tree in which the second-order derivative of the heuristic function is difficult to analytically evaluate, an approach to numerically estimating the second-order is necessary. For more details to numerical computation of derivative, one can see Ref. [18].

Table 2
The monotonicity of p_1 and p_2 in these subintervals.

Subintervals	$f''(x) < 0$			$f''(x) = 0$	$f''(x) > 0$		
State 1	Increase	Convex	Decrease	Constant	Increase	Convex	Decrease
State 2	Increasing			Increasing	Increasing		
State 3	Decreasing			Decreasing	Decreasing		
State 4	Increase	Convex	Decrease	Constant	Increase	Convex	Decrease

Table 3
Data sets used in our experiments.

Data set	# Of cases	# Of classes	# Of attributes
Pima	768	2	9
Breast cancer	683	2	10
Credit	690	2	10
Abalone	4177	29	8
Clouds	5000	2	2
SVMguide1	7089	2	4
Waveform	5000	3	21
Waveform + noise	5000	3	40
MAGIC04	19,020	2	10

(1) $f_1(x) = -x\log_2 x - (1-x)\log_2(1-x)$. This special case is corresponding to the classical entropy heuristic information function.
(2) $f_2(x) = x(1-x)$. This special case is corresponding to the Gini index, another classical heuristic information function.
(3) $f_3(x) = \begin{cases} x & 0.5 < x \leqslant 1 \\ 1-x & 0 \leqslant x \leqslant 0.5 \end{cases}$. This case corresponds to the classical min–max heuristic.
(4) $f_4(x) = \sin(\pi x)$. This is a case different from several classical heuristics.

Usually the used heuristic functions are smooth (such as entropy and Gini index) with expressions of elementary functions. Their second-order derivative is easy to evaluate. But for particular engineering problems in which the function is not smooth, a numerical method for computing the second-order derivative is necessary [18]. It is easy to directly evaluate the second-order derivatives of these 4 generalized heuristic information functions as follows: $f_1''(x) = \frac{1}{\ln 2} \times \frac{1}{x(1-x)}$, $f_2''(x) = -2$, $f_3''(x) = 0$ and $f_4''(x) = -\pi^2 \sin(\pi x)$.

A reason for selecting the 4 heuristics is to verify that (1) our proposed generalized heuristic information function can include many specific forms and (2) the frequently used classical heuristics, entropy and Gini index, can be considered as two special cases of our generalized function.

Sure, there are other forms of heuristics to be selected. Here we only would like to show such a statement that any function satisfying conditions of our generalized function (given in Section 3.5) can be selected as a heuristic for which the non-boundary cuts are not necessary to be evaluated during building the decision tree. (It is worth noting that there exit many heuristics for which all cuts (boundary and non-boundary) must be evaluated during building the decision tree.)

5.3. Experimental steps

The ten-fold cross-validation is performed for each data set. With the change of generalized heuristic information functions, we observe (1) the training accuracy, (2) the testing accuracy, and (3) the ratio of boundary-cuts to all cuts. The experiments repeat 5 times and the averaged values are recorded for each heuristic information function. The experimental records are summarized in Table 4 where the symbol NB represents Non-Boundary.

Table 4
Experimental results. The bold values indicates the best classification accuracy among 4 heuristics.

Data sets	Training accuracy ($f_1/f_2/f_3/f_4$)	Testing accuracy ($f_1/f_2/f_3/f_4$)	Ratio of NB cuts (%)
Pima	(0.7897/0.7991/0.7923/0.8163)	(0.7525/0.7636/0.7656/0.7893)	67.87
Breast cancer	(0.9542/0.9467/0.9502/0.9488)	(0.9213/0.9147/0.9322/0.9298)	78.23
Credit	(0.8636/0.8507/0.8644/0.8596)	(0.8029/0.8164/0.8211/0.8194)	75.56
Abalone	(0.6381/0.6504/0.6448/0.6391)	(0.6139/0.6014/0.6122/0.6087)	62.31
Clouds	(**0.8774**/0.8559/0.8354/0.8506)	(**0.8697**/0.8468/0.8432/0.8379)	70.81
SVMguide1	(0.7283/0.7455/0.7345/0.7301)	(0.7025/0.6765/0.6904/0.7117)	69.75
Waveform	(0.8485/**0.8662**/0.8421/0.8460)	(0.8348/**0.8579**/0.8334/0.8398)	72.66
Waveform + noise	(0.8406/0.8317/0.8385/0.8396)	(0.8259/0.8146/0.8164/0.8245)	73.59
MAGIC04	(0.7804/0.7826/0.7773/0.7801)	(0.7598/0.7461/0.7497/0.7588)	71.67

5.4. Experimental analysis

It is easy to view from Table 4 that the learning accuracy including training and testing is strongly dependent on the selection of generalized heuristic functions. For example, the heuristics 1 and 2 have the similar advantages regarding data sets Clouds and Waveform respectively. And the heuristic 4 has the advantages more than the other 3 heuristics with respect to the data set Pima. It is sure that the performance of learning is also dependent on the specific characteristic of the individual datasets, and it is interesting to give a specifically detailed analysis on relationship between individual datasets and their suitable heuristics.

Here, we select the Pima datasets for the analysis, on which the heuristic 4 obtains the better classification accuracy. Pima India diabetes data has 8 numerical attributes, and contains 768 cases related to the diagnosis of diabetes (268 positive and 500 negative). The local structure of Pima shows very nonlinear property. It is observed that there exist many cases that two samples are very near but their classes are different. It results in a boundary cut phenomenon. That is, a sample near boundary cuts usually has the statistical testing error more than a sample far from boundary cuts. Since the heuristic function with smooth second-order derivatives can bring more boundary cuts to some extent and then be adaptive to the highly nonlinear boundary, the bell shaped heuristics (such as heuristic 4) are suitable more than other type of heuristics for Pima dataset.

It is noted that robustness of a decision tree depends on he selection of heuristic information functions. Specifically we find that the decision trees generated based on heuristics 2, 3 and 4 respectively for Pima India diabetes data are more robust than the one based on heuristic 1. It can be seen that the testing accuracies obtained by decision trees with heuristics 2, 3 and 4 are all higher than the testing accuracy of decision tree with heuristic 1, which is the mostly-used heuristic in decision tree induction [5,7]. It is acknowledged that there are many noise data in Pima India diabetes dataset [20] but the heuristics 2, 3 and 4 enhance rather than degrade the classification performance of decision tree on Pima India diabetes dataset. This indicates that the heuristics 2, 3 and 4 are more insensitive to the noise data than heuristic 1.

We employ Wilcoxon signed-ranks test [21] to examine whether the difference among the 4 heuristics is significant. Wilcoxon signed-ranks test is safe and robust non-parametric test for statistical comparison of two classification methods [22]. In our experiment, 10-fold cross validation is repeated 5 times. There are 5×10 differences, and Wilcoxon signed-ranks test is distributed approximately normally. For a confidence level of 0.05 and regarding 6 pairs of heuristics, the tests give a result that the five differences are significant and one is not. It shows from the viewpoint of statistics that the learning accuracy is really dependent on the selection of heuristics.

From the last column of Table 4 one cane see that the ratio of non-boundary cuts to all cuts is 0.7138 in average. It implies that, during the decision tree generation, around 71.38% computational load (which refers to the times of detecting candidate cuts) can be saved. The analysis on the second-order derivatives of generalized heuristic information functions really can really help reduce the computational complexity of generating a decision tree.

Although the decision tree learning system depends on the selection of heuristics, generally it is hard to say which kind of heuristic functions can significantly outperform other heuristics in decision tree learning from data with numerical attributes. It strongly depends on the local features of data. Our experiments confirm this conclusion.

6. Conclusions and future works

In this paper, we reviewed the process of decision tree induction with continuous valued attributes and several classical heuristic information functions. The expanded attribute for splitting a node to two sub-nodes is associated with a best cut. For classification problems in which the decision tree learning is based on finding best cuts, we have presented a generalized heuristic information function covering those existing frequently used heuristic information functions. We mathematically obtained a relationship between the second-order derivative of heuristic information functions and locations of optimal cuts, and further confirmed it experimentally. The relationship clearly indicates that the non-boundary cuts are not necessary to be detected when the generalized heuristic function meets some conditions related to the second-order derivatives. We statistically showed that the learning accuracy (including training and testing) is dependent strongly on the selection of heuristics. Considering the impact of this relationship on building a decision tree, we can significantly reduce the number of detected cuts, which indicates a big reduction of computational complexity for using cuts to generate a binary tree with continuous attributes. Furthermore, we experimentally showed that the generalization capability of the decision tree can be improved by incorporating this relationship into the process of decision tree generation, and the magnitude of improvement is generally dependent on the local characteristics of a specific data set.

Our future works regarding this topic will include how to categorize the generalized heuristic information functions such that a sub-category of heuristics can have better performance than other sub-categories with respect to a specified group of classification problems with continuous valued attributes.

Acknowledgements

This research is partially supported by the National Natural Science Foundations of China (71371063, 61170040 and 60903089), by the Natural Science Foundations of Hebei Province (F2011201063, F2012201023, F2013201110, F2013201060 and F2013201220), and by the Key Scientific Research Foundation of Education Department of Hebei Province (ZD2010139).

References

[1] Mitchell T. Machine learning. New York: McGraw Hill; 1997.
[2] Wang XZ, Dong CR. Improving generalization of fuzzy if-then rules by maximizing fuzzy entropy. IEEE Trans Fuzzy Syst 2009;17(3):556–67.
[3] Yi WG, Lu MY, Liu Z. Multi-valued attribute and multi-labeled data decision tree algorithm. Int J Mach Learn Cybern 2011;2(2):67–74.
[4] Wang XZ, Dong LC, Yan JH. Maximum ambiguity based sample selection in fuzzy decision tree induction. IEEE Trans Knowl Data Eng 2012;24(8):1491–505.
[5] Quinlan JR. Induction of decision trees. Mach Learn 1986;1(1):81–106.
[6] López R, Mántaras D. A distance-based attribute selection measure for decision tree induction. Mach Learn 1991;6(1):81–92.
[7] Quinlan JR. C4.5: programs for machine learning. Morgan Kaufman; 1993.
[8] Todorovski L, Džeroski S. Combining classifiers with meta decision trees. Mach Learn 2003;50(3):223–49.
[9] Olshen L, Breiman JH, Friedman RA, Stone CJ. Classification and regression tree. Monterey, Calif, USA: Wadsworth International Group; 1984.
[10] Utgoff PE, Berkman NC, Clouse JA. Decision tree induction based on efficient tree restructuring. Mach Learn 1997;29(1):5–44.
[11] Quinlan JR. Improved use of continuous attributes in C4.5. J Artif Intell Res 1996;4:77–99.
[12] Fayyad UM, Irani KB. On the handing of continues-valued attributes in decision tree. Mach Learn 1992;8:87–102.
[13] Fayyad UM, Irani KB. Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the 13th international joint conference on artificial intelligence; 1993. p. 1022–7.
[14] Wang XZ, Hong JR. Learning algorithm of decision tree generation for interval-valued attributes. J Softw 1998;9(8):637–40.
[15] Yen E, Chu IWM. Relaxing instance boundaries for the search of splitting points of numerical attributes in classification trees. Inform Sci 2007;177(5):1276–89.
[16] Liu H, Hussain F, Tan CL, Dash M. Discretization: an enabling technique. Data Min Knowl Discov 2002;6:393–423.
[17] Yuan Y, Shaw MJ. Induction of fuzzy decision trees. Fuzzy Sets Syst 1995;69:125–39.
[18] Kiusalaas J. Numerical methods in engineering with MATLAB. Cambridge University Press; 2010.
[19] UCI Machine learning repository. <http://archive.ics.uci.edu/ml/>.
[20] Zhang L, Coenen F, Leng P. An attribute weight setting method for k-NN based binary classification using quadratic programming. In: Proceedings of the 15th European conference on artificial intelligence; 2002. p. 325–9.
[21] Wilcoxon F. Individual comparisons by ranking methods. Biometrics Bull 1945;1(6):80–3.
[22] Demsar J. Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 2006;7:1–30.

Hong-Yan Ji received her Master's degree in applied mathematics from Hebei University in 2006. She is currently working at department of mathematics in Hebei University of Engineering as a lecturer, and is an on-job Ph.D. candidate. Her research interests include fuzzy set theory, decision trees with uncertainty, and approximate reasoning.

Xi-Zhao Wang received the Ph.D. degree in computer science from Harbin Institute of Technology in 1998. He is presently the dean and professor at school of Mathematics and Computer Science in Hebei University. His main research interests include learning from examples with fuzzy representation, multi-classifier fusion, and the recent machine learning with big data. He is an IEEE Fellow.

Yu-Lin He received the Master's degree in computer science from Hebei University, China, in 2009, where he is currently working toward the Ph.D. degree in the College of Mathematics and Computer Science. His research interests include neural networks, decision trees and approximate reasoning.

Wen-Liang Li received the Master's degree in computer science from Hebei University, China, in 2011. His research interests include computational intelligence in game, neural networks and decision trees.