TDK-dolgozat

Puskás Levente

Extension of decision trees in case of uncertain inputs and outputs

Author:
Puskás Levente
Msc. Programming Informatics 1.semester

Supervisor:
Dombi József
Professor

Szegedi Tudományegyetet
Természettudományi és Informatikai Kar
Számítógépes Algoritmusok és Mesterséges Intelligencia Tanszék

Abstract

A gépi tanulás adatbázison alapuló algoritmusok fejlesztése. Ma az egyik legfontosabb kérdés az eljárások interpretálhatósága. A döntési fák intuitíav magyarázattal szolgálnak a döntésekre, így ezen a területen fontos szerepet játszanak. Az egyik legelterjettebb fa építő algoritmus az ID3. Ez az eljárás a fa felépítéséhez az entrópiából számol információ nyereséget. Ami ugyan egy hatékony heurisztika, de lehetőség van más függvények alkalmazására is. A dolgozatban bevezetünk egy új függvényt a fa felépítéséhe, amit a a fuzzyság mértéke alapján konstruáljuk fuzzy operátor segítségével. Az új függvény hasonló eredményeket ad mint az ID3 -ban a Shannon entrópia. Az eljárás egyszerűbb és jól interpretálható. Bevezettünk egy új módszert a döntési fa felépítésére amivel gyorsíthatjuk az eljárást. Kiterjesztjük az eljárást arra az esetre is ha a bemenetek valószínűségi értékekkel rendelkeznek.

Abstract

Nowadays one of the most important field of machine learning is interpretability. The decision trees provide an intuitive explanation to decisions, hence they play an important role on this field. One of the most widely used tree building algorithm is the ID3. This method calculates the information gain from the entropy. It is and effective heuristic, however we have the option to choose other functions too. In the paper we introduced a new function to build the tree. We constructed it based on the measure of fuzzyness. The new function produces similar results in the ID3 as the Shannon entropy. Our method is easier to handle and it is easily interpretable. We also introduced a new method to accelerate the construction of trees, and extended it so it can handle inputs if they have probabilistic nature.

1 Introduction

In learning tasks, the decision process has two major aims. First, it is to explain decisions and second, it is to make recommendations on how to make a decision in certain circumstances. These are very similar to the goals of the Inductive Learning approach [RN95; Mit97] in the field of Artificial Intelligence (AI), where the first step is to establish a model based on previous experience which is later applied to predict future situations. This parallel computation suggests that AI methods can be applied effectively in Decision Support Systems.

In Multicriteria Decision Analysis (MCDA), the inputs are usually described by numerical (continuous) criteria, such as value functions or orderings on a real interval. However, most learning methods in the field of artificial intelligence can detect relationships among elements of an input dataset described by a set of categorical (discrete) criteria (like hierarchical classifiers, decision trees). In order to apply these methods in MCDA, they should be extended so that they work on numerical criteria/attributes. In this paper, we propose such an extension. Our novel Continuous Decision (CD) and Continuous Decision Tree (CDT) methods help elucidate the structure of the input dataset described by a set of numerical criteria in the form of a discriminant function or a decision tree, respectively, which could be transformed into decision rules.

Here, some shortcomings of the well-known ID3 decision tree building method is discussed and solutions are proposed. An alternative measure is described that differs from the entropy function, which builds on the measure of fuzziness using a monotone fuzzy operator. In the proposed methods, continuous criteria are handled without discretization of their values. If the problem is viewed from a geometric perspective, our method allows us to separate the decision space with arbitrary figures, such as hyperplanes and spheres, which can also be interpreted in a straightforward way using the decision maker.

We shall consider classifiers from a Machine Learning perspective, and define the concept of the decision tree. We will also discuss the properties of ID 3. Our CDT method, introduced in the following section, is based on these approaches.

2 Decision tree classifiers

Classification models can be grouped by the way they are constructed. Namely, they are either made by human experts or are obtained inductively from a set of examples. The induced model is either

non-hierarchical (e.g. instance-based classifiers, or models obtained from a neural network, genetic algorithm, statistical method) or hierarchical, such as decision trees (for a short overview and further references, see [Qui93; Bre+84]). We will derive a hierarchical classifier that is constructed inductively.

In the following, T denotes the given set of elements with their class information (training set) from which a decision tree is induced:

$$T = \{(\mathbf{x}, c(\mathbf{x}))|\mathbf{x} \in X\},$$

where $\mathbf{x} \in X$ is described by a sequence of attribute values: $\mathbf{x} = (x_1, \ldots, x_m)$, x_i is the value of the ith attribute, m is the number of attributes, and $c(\mathbf{x})$ denotes the class of element $\mathbf{x}$. Let $C_1, \ldots, C_k$ denote the possible classes of elements in T.

From a machine learning point of view, two different types of attributes are distinguished: an attribute is either discrete (categorical), i.e. its value comes from a predefined finite set, or continuous (numerical), i.e. it is an element of a real interval.

Table 1: The training set that specifies when to go out and play

outlook	Temp(°F, °C)	Humidity(%)	Windy?	Class
sunny	75, 23.9	70	true	Play
sunny	80. 26.7	90	true	Don't Play
sunny	85, 29.4	85	false	Don't Play
sunny	72, 22.2	95	false	Don't Play
sunny	69, 20.6	70	false	Play
overcast	72, 22.2	90	true	Play
overcast	83, 28.3	78	false	Play
overcast	64, 17.8	65	true	Play
overcast	81, 27.2	75	false	Play
rain	71, 21.7	80	true	Don't Play
rain	65, 18.3	70	true	Don't Play
rain	75, 23.9	80	false	Play
rain	68, 20	80	false	Play
rain	70, 21.1	96	false	Play

A classifier is a model built from the training dataset, and it is applied later to predict class values of unknown elements. The model is based on the attribute values of the elements. A typical classifier is the decision tree (see Figure 1):

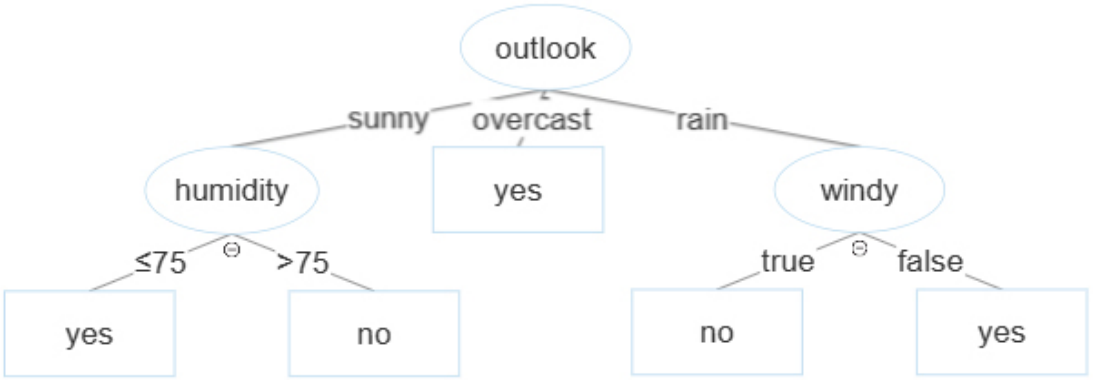

Figure 1: A decision tree built up from the training set in Table 1 using the C4.5 method

Definition 2.1. A decision tree is a special rooted tree, in which a class identifier is associated with each leaf node (it determines the class of elements which reached the node), and each internal (or

```
function BuildTree(examples, C_def)
  return a decision tree
  input: examples: set of examples with
       classes C_1, C_2, ..., C_k
       C_def: default class
  if all examples have class C_j then
    return leaf with title C_j
  else if examples is empty then
    return leaf with title C_def
  else
    test := select a test to separate examples
    make an inner node with test
    for each outcome O_i of test
      examples_i := elements of examples which outcome of test is O_i
      determinate value of C_def
      subtree_i := BuildTree(examples_i, C_def)
    return tree
```

Figure 2: The greedy algorithm used to build a decision tree from a training set of examples

decision) node specifies a test, with one branch and subtree for each outcome of the test.

Methods for constructing decision trees may be grouped by the variety of tests used in their inner nodes. In some approaches, only a single attribute selection is allowed as a test. For example, in the CART method[Bre+84], in Quinlan's ID3 method and in its extension to handle continuous attributes, the C4.5 method[Qui93]. In other methods, a mixture of the attributes is also allowed as tests, as in Cios's CID3[CL], in Oblique Decision Trees[MKS], and in our CDT method.

From a geometric perspective, the application of tests based on single attribute selection can be interpreted as a partition of the decision space – which is described by a set of continuous attributes – with hyperplanes whose edges are parallel to the axes. A mixture of attributes leads to hyperplanes in arbitrary positions. In our CDT method, arbitrary figures can be applied for the separation of the space, while in C4.5 only axis-parallel hyperplanes are used (Figure **??**).

In order to construct a decision tree, a training set is given, where elements are described by some (discrete or continuous) attributes, and a tree is searched for in the hypothesis space which best fits the elements of the training set. It is easy to find one that is consistent with the training set[RN95]. For example, a tree that contains only one element of the training set in all leaf nodes. This tree, however, does not tell us anything about the structure of the original problem and it has very poor predictive power. According to the principle of Occam's razor, it is worth looking for one of the smallest decision trees. The motivation is that a simple decision tree (less complex model) might perform better on unseen elements than a more complicated one[KV94; Mit97]. However, the problem of finding the smallest decision tree consistent with a training set is NP-Complete[HR].

To cope with this computational problem, usually a greedy divide-and-conquer algorithm is used to build a decision tree consistent with the training set (which, of course, does not lead to the smallest one in the general case)[Ste; Qui93]. The generic algorithm is presented in Figure 2. Initially, the one-node decision tree is considered (containing all the elements of the training set). As the algorithm proceeds, for in each step a test is chosen and it is applied to the examples that have reached the examined node. This test is then assigned to the node and branches are created according to the outcomes of the selected test. Then, the same method is recursively applied to the newly created branches. The crucial point of this algorithm is the test selection criterion. The methods discussed in the present paper (ID3, C4.5, CDT) differ at this point. In all three approaches, the test selection criterion is based on the homogeneity of the training set according to the class values. In ID3 and C4.5 the measure of homogeneity is based on the entropy function[Vet], while in CDT a different measure is applied, derived from fuzzy conjunction operators.

In the following, the well-known ID3 and C4.5 decision tree building algorithms are discussed.

2.1 ID3 algorithm based on Shannon entropy

Now, we will examine the original ID3 algorithm using new notations. Let the data table have the following form:

	C_1	C_2	...	C_m	R
a_1					
a_2					
$\vdots$					
a_l					r_l
$\vdots$					
a_N					

where a_i are the examples, C_k are the properties and $r_l \epsilon \{+, -\}$ i.e. a_i is a positive or negative example.

The attributes of the C_k's are $S_{k1}, S_{k2} \dots S_{kn_k}$, i.e.

$$C_k = \{S_{k1} \dots S_{kn_k}\}$$

We shall define the following:

$|S|$ the total number of examples (N)
$|S^+|$ the number of positive examples
$|S^-|$ the number of negative examples
$|S^+_{ki}|$ the number of positive examples for taken S_{ki}
$|S^-_{ki}|$ the number of negative examples for taken S_{ki}

The following identities are valid.

1. $|S| = |S^+| + |S^-|$
2. $|S^+| = \sum_{i=1}^{n_k} |S^+_{ki}|$ and $|S^-| = \sum_{i=1}^{n_k} |S^-_{ki}|$

For the ID3 algorithm, the Shannon entropy function plays a crucial rule:

$$\mathcal{E}(x) = -k \sum_{i=1}^{n} x_i ln(x_i)$$

In our case we have only positive and negative examples, so the entropy is

$$\mathcal{E}(x) = -\frac{1}{ln(2)}(xln(x) + (1-x)ln(1-x)) \tag{1}$$

When we constructed ID3, using entropy was just a heuristic idea. In our case we define

$$J(S) = -\frac{1}{ln(2)} \left(\frac{|S^+|}{|S|} ln \frac{|S^+|}{|S|} + \frac{|S^-|}{|S|} ln \frac{|S^-|}{|S|} \right). \tag{2}$$

We have to calculate the expected value of C_k.

$$E_S(C_k) = \frac{|S_{k1}|}{|S|} J(S_{k1}) + \frac{|S_{k2}|}{|S|} J(S_{k2}) + \dots \frac{|S_{kn_k}|}{|S|} J(S_{kn_k})$$

We have to choose the attributes where $J(S) - E_S(C_k)$ is the maximum. Because $J(S)$ is constant, we will look for a C_k where $E_S(C_k)$ is a minimum.

3 Vagueness measue instead of Shannon entropy

The basic idea of fuzzy sets is the introduction of the membership function, which replaces the classical characteristic function. It is an interesting question of knowing how close the membership function is to the characteristic function, when we use a certain class of membership functions. This measure is called the fuzziness measure.

Below we shall introduce an operator-dependent fuzziness measure called the vagueness measure. We will show that this measure satisfies the usual classical assumptions for the fuzziness measure. In addition, we will show that there is a connection between this measure and the entropy function.

In the Pliant concept we have the distending function instead of the membership function based on the Pliant operator, and now we will define a vagueness measure by using the generator function of the Pliant operator. On the basis of this consistent concept, we can derive a convergence theorem.

First, we will take a closer look at the fuzziness measure: Let $\mu(x)$ be the membership function and $d(\mu)$ be the fuzziness measure.

3.1 Vagueness measure

In the Pliant system, the logical values essentially arise from inequalities. If we are on the border of the inequality, i.e. just the equalities are fulfilled we are not sure whether we are inside or outside a region. If we move away from the border we are more likely to be inside or outside the region. Why are we so vague on the border? Because small changes can radically change the logical value. If we demand stable statements, we should avoid being just on the border. Hence it is important to measure the vagueness and also to know how it depends on the vagueness on the input values.

As we mentioned above, the idea and construction of a vagueness measure can be derived from the fuzziness measure. In 1972 DeLuca and Termini [DT] introduced a fuzziness measure that is used for the membership function. In the Pliant concept we will use the vagueness measure as the operand of the distending function.

In our concept the vagueness measure depends on the negation function. Let us denote the fixed point of the negation η by ν. We will suppose that the maximum of uncertainty is at ν. In terms of the fuzziness measure, the negation is $1-x$ and the fixed point is $\frac{1}{2}$ ((P2) property). We generalized it by replacing $\frac{1}{2}$ by ν.
The vagueness measure in based on a logical operator, namely on the conjunctive operator. In our starting point, we use the function

$$F(x) = K\bar{c}(x, \eta_\nu(x)),$$

where $\bar{c}$ is the mean conjunctive operator and η is the negation operator. It is obvious that $F(0) = 0$ and $F(1) = 0$ ((P1) property). We have to multiply F by a constant K, such that $KF(\nu) = 1$.

Proposition 3.1. *In Pliant system the constant K is $\frac{1}{\nu}$.*

Proof. In the Pliant system

$$\bar{c}(x, y) = f^{-1}\left(\frac{1}{2}(f(x) + f(y))\right) \tag{3}$$

and

$$\eta(x) = f^{-1}\left(\frac{f^2(\nu)}{f(x)}\right) \tag{4}$$

Here, f is the generator function of the conjunctive operator. [**Ins001**]
So we have the following representation of vagueness:

$$F(x) = K\bar{c}(x, \eta(x)) = Kf^{-1}\left(\frac{1}{2}\ f(\nu)\left(\frac{f(x)}{f(\nu)} + \frac{f(\nu)}{f(x)}\right)\right). \tag{5}$$

If we demand that the maximum value should be at ν, we have to find the minimum of F. We look for the minimum of

$$Y = \frac{X}{A} + \frac{A}{X}, \qquad \text{where } A = f(\nu),\ X = f(x). \tag{6}$$

It is obvious that at $X = f(\nu)$, i.e. $f(x) = f(\nu)$, the minimum is at ν, and we get

$$K = \frac{1}{\nu}$$

□

Definition 3.2. The vagueness measure is denoted by $\mathcal{V}$ and

$$\mathcal{V}(x) = \frac{1}{\nu}\ \bar{c}(x, \eta(x)) = \frac{1}{\nu}\ f^{-1}\left(\frac{1}{2}\ f(\nu)\left(\frac{f(x)}{f(\nu)} + \frac{f(\nu)}{f(x)}\right)\right). \tag{7}$$

Proposition 3.3. *The vagueness measure has the following properties.*

1. *Sharpness (No vagueness)* *(P1)*

   $\mathcal{V}(x) = 0 \qquad \Longleftrightarrow \qquad x \in \{0, 1\}$

2. *Maximality (maximal vagueness)* *(P2)*

   $\frac{1}{\nu}\ \mathcal{V}(x) = 1 \qquad \Longleftrightarrow \qquad x = \nu$

3. *Monotonicity* *(P3)*

   $\mathcal{V}(x_1) < \mathcal{V}(x_2)$ *if*

   $x_1 < x_2$ *and* $x_1 \leq \nu$

   or

   $x_1 > x_2$ *and* $x_1 \geq \nu$

4. *Symmetry* *(P4)*

   $\mathcal{V}(x) = \mathcal{V}(\eta(x))$

Proof. The proof is trivial. □

3.2 Vagueness measure in the Dombi operator case

Let $f(x) = \left(\frac{1-x}{x}\right)^{\alpha}$ $(\alpha > 0)$, namely the Dombi operator. Then

$$\mathcal{V}(x) = \frac{1}{\nu}\frac{1}{1 + \frac{1-\nu}{\nu}\left(\frac{1}{2}\left(\left(\frac{\nu}{1-\nu}\ \frac{1-x}{x}\right)^{\alpha} + \left(\frac{1-\nu}{\nu}\ \frac{x}{1-x}\right)^{\alpha}\right)\right)^{\frac{1}{\alpha}}} \tag{8}$$

If $\alpha = 1$ and $\nu = \frac{1}{2}$, then

$$\mathcal{V}(x) = 4x(1-x). \tag{9}$$

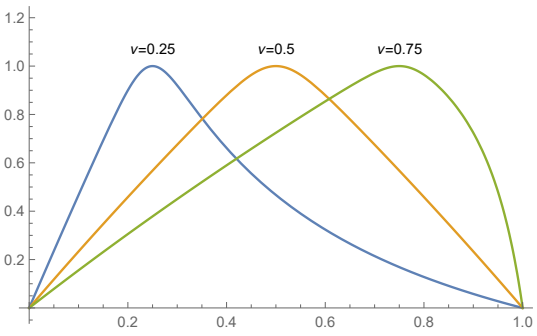

Figure 3: The effect of ν

4 ID3 algorithm based on vagueness measure

Next, we will introduce the following notations:

$$\begin{array}{lll} |S_{k1}| = |S_{k1}^+| + |S_{k1}^-| & x_{k1}^+ = \dfrac{|S_{k1}^+|}{|S^+|} & x_{k1}^- = \dfrac{|S_{k1}^-|}{|S^-|} \\ |S_{k2}| = |S_{k2}^+| + |S_{k2}^-| & x_{k2}^+ = \dfrac{|S_{k2}^+|}{|S^+|} & x_{k2}^- = \dfrac{|S_{k2}^-|}{|S^-|} \\ \vdots & \vdots & \\ |S_{kn_k}| = |S_{kn_k}^+| + |S_{kn_k}^-| & x_{kn_k}^+ = \frac{|S_{kn_k}^+|}{|S^+|} & x_{kn_k}^- = \frac{|S_{kn_k}^-|}{|S^-|} \end{array} \tag{10}$$

$$|S| = |S^+| + |S^-| \qquad w^+ = \frac{|S^+|}{|S|} \qquad w^- = \frac{|S^-|}{|S|},$$

where

$$|S^+| = \sum_{i=1}^{n_k} |S_{ki}^+| \qquad |S^-| = \sum_{i=1}^{n_k} |S_{ki}^-|$$

and, of course,

$$w^+ + w^- = 1 \qquad w\epsilon[0,1]$$

$$\sum_{i=1}^{n_k} x_{ki}^+ = 1 \qquad \sum_{i=1}^{n_k} x_{ki}^- = 1 \qquad x_{ki}^+\epsilon[0,1],\ x_{ki}^-\epsilon[0,1]$$

The following equations are valid.

1. $w^+ = \dfrac{|S^+|}{|S|}$ and $w^- = \dfrac{|S^-|}{|S|}$
2. $w^+ + w^- = 1$, where $w^+ \in [0,1], w^- \in [0,1]$
3. $\sum_{i=1}^{n_k} x_{ki}^+ = 1$ and $\sum_{i=1}^{n_k} x_{ki}^- = 1$, where $x_{ki}^+ \in [0,1], x_{ki}^- \in [0,1]$

Replacing (1) by (9) the vagueness measure might be as good as the entropy. It's worth mentioning that using (9), it can be calculated much more easily. If $x = 1$ or $x = 0$, then (1) has no meaning and we calculate the limes value. So instead of calculating the entropy, we will use the vagueness measure in Dombi operator case:

$$J(S) = 4\frac{|S^+|}{|S|}\left(1 - \frac{|S^+|}{|S|}\right) = 4\frac{|S^+||S^-|}{|S|^2} \tag{11}$$

Let us calculate the $J(C_k)$ values:

$$J(S_{k1}) = 4\frac{|S_{k1}^+||S_{k1}^-|}{|S_{k1}|^2}$$

$$J(S_{k2}) = 4\frac{|S_{k2}^+||S_{k2}^-|}{|S_{k2}|^2}$$

$$\vdots$$

$$J(S_{kn_k}) = 4\frac{|S_{kn_k}^+||S_{kn_k}^-|}{|S_{kn_k}|^2}$$

In a similar way to that for ID3, we have to calculate the expected value of C_k.

$$E_D(C_k) = 4\sum_{i=1}^{n_k}\frac{|S_{ki}|}{|S|}\frac{|S_{ki}^+||S_{ki}^-|}{|S_{ki}|^2} = \frac{4}{|S^+|+|S^-|}\sum_{i=1}^{n_k}\frac{|S_{ki}^+||S_{ki}^-|}{|S_{ki}^+|+|S_{ki}^-|}, \tag{12}$$

where we use

$$|S| = |S^+| + |S^-|$$

$$|S_{ki}| = |S_{ki}^+| + |S_{ki}^-|$$

Let us use x_{ki}^+, x_{ki}^-, w^+ and w^-, as defined in (10)

$$E_D(C_k) = \frac{4}{|S|w^+ + |S|w^-}\sum_{i=1}^{n_k}\frac{|S^+|x_{ki}^+\ |S^-|x_{ki}^-}{|S^+|x_{ki}^+ + |S^-|x_{ki}^-}.$$

Because $|S|w^+ + |S|w^- = |S| = |S^+| + |S^-|$

$$\begin{aligned}
E_D(C_k) &= \frac{4|S^+||S^-|}{|S^+|+|S^-|}\ \sum_{i=1}^{n_k}\frac{x_{ki}^+x_{ki}^-}{|S^+|x_{ki}^+ + |S^-|x_{ki}^-} \\
&= 4\frac{|S^+|}{|S^+|+|S^-|}\frac{|S^-|}{|S^+|+|S^-|}\sum_{i=1}^{n_k}\frac{x_{ki}^+x_{ki}^-}{\frac{|S^+|}{|S^+|+|S^-|}x_{ki}^+ + \frac{|S^-|}{|S^+|+|S^-|}x_{ki}^-} \\
&= 4w^+w^-\sum_{i=1}^{n_k}\frac{x_{ki}^+x_{ki}^-}{w^+x_{ki}^+ + w^-x_{ki}^-}
\end{aligned} \tag{13}$$

Since $4w^+w^-$ is a constant, we can minimize

$$\sum_{i=1}^{n_k}\frac{x_{ki}^+x_{ki}^-}{w^+x_{ki}^+ + w^-x_{ki}^-} = \sum_{i=1}^{n_k}\frac{1}{1 + w^+\frac{1-x_{ki}^-}{x_{ki}^-} + w^-\frac{1-x_{ki}^+}{x_{ki}^+}}. \tag{14}$$

$E_D(C_k)$ can also be defined in the following way:

$$\begin{aligned}
E_D(C_k) &= 4w^+w^-\sum_{i=1}^{n_k}\frac{1}{1 + \frac{w^+}{x_{ki}^-} - w^+ + \frac{w^-}{x_{ki}^+} - w^-} \\
&= 4w^+w^-\sum_{i=1}^{n_k}\frac{1}{1 + w^+\frac{1-x_{ki}^-}{x_{ki}^-} + w^-\frac{1-x_{ki}^+}{x_{ki}^+}}
\end{aligned} \tag{15}$$

As the weighted Dombi operator is

$$c_D(u, v; x, y) = \frac{1}{1 + u\frac{1-x}{x} + v\frac{1-y}{y}},$$

we can express $E_D(C_k)$ as

$$E_D(C_k) = 4w^+w^- \sum_{i=1}^{n_k} c_D(w^+, w^-; x_{ki}^-, x_{ki}^+). \tag{16}$$

We get the following result.

We have only positive and negative examples and we have to find the *minimum* of $E(C_k)$. Because $4w^+w^-$ does not depend on k, we can ignore it and choose

$$K = arg \min_k \sum_{i=1}^{n_k} \frac{1}{1 + w^+ \frac{1-x_{ki}^-}{x_{ki}^-} + w^- \frac{1-x_{ki}^+}{x_{ki}^+}} = arg \min_k \sum_{i=1}^{n_k} c_D(w^+, w^-; x_{ki}^-, x_{ki}^+) \tag{17}$$

We should mention that if

$$x_{ki}^- = 0 \qquad \text{or} \qquad x_{ki}^+ = 0,$$

then the value of the operator is 0.

Instead of (17), we can use (12) as well:

$$K = arg \min_k \sum_{i=1}^{n_k} \frac{|S_{ki}^+||S_{ki}^-|}{|S_{ki}^+| + |S_{ki}^-|} \tag{18}$$

Based on (17), we can find the optimum for k. Based on k we can build two sets of the items denoted by P for the positive response and N for the negative response. We will do it for all i. Now we will divide the original set of items into $i_1, \ldots, i_{n_k}$ subsets. The algorithm works on these subsets using the divide and conquer procedure. We halt the algorithm if N or P are empty.

Example 1.

Database:

	C_1	C_2	C_3	R
1	B	3	b	+
2	A	3	a	-
3	A	2	b	+
4	B	1	b	-
5	A	1	b	-
6	A	3	b	+
7	A	1	a	-
8	B	3	a	-

Property 1.	(A, B)	(C_1)	$S_{11} = A$	$S_{12} = B$	
Property 2.	$(1, 2, 3)$	(C_2)	$S_{21} = 1$	$S_{22} = 2$	$S_{23} = 3$
Property 3.	(a, b)	(C_3)	$S_{31} = a$	$S_{32} = b$	

We have 8 examples, 3 of them being positive and 5 of them being negative. Hence

$$w^+ = \frac{3}{8} \qquad w^- = \frac{5}{8}$$

Let us choose Property 1 and look for A. A is positive in (3) and (6), and the positive examples are 3, so

$$x_{11}^+ = \frac{2}{3}$$

A is negative in cases (2), (5) and (7) hence all together we have 5 negative examples, and

$$x_{11}^{-} = \frac{3}{5}$$

We have to calculate it for B too:

$$x_{12}^{+} = \frac{1}{3} \qquad x_{12}^{-} = \frac{2}{5}$$

Therefore

$$E_D(C_1) = \frac{1}{1 + \frac{3}{8}\frac{1-\frac{3}{5}}{\frac{3}{5}} + \frac{5}{8}\frac{1-\frac{2}{3}}{\frac{2}{3}}} + \frac{1}{1 + \frac{3}{8}\frac{1-\frac{2}{5}}{\frac{2}{5}} + \frac{5}{8}\frac{1-\frac{1}{3}}{\frac{1}{3}}} = \frac{225}{224} = 0.9955$$

Using the Shannon entropy we can calculate $E_S(C_1)$ too. In this case $|S_{11}| = 5, |S_{12}| = 3, |S| = 8, |S_{11}^{+}| = 2, |S_{11}^{-}| = 3, |S_{11}| = 5, |S_{12}^{+}| = 1, |S_{12}^{-}| = 2, |S_{12}| = 3$, so we have

$$J(S_{11}) = -\frac{1}{ln(2)}\left(\frac{2}{5}ln\frac{2}{5} + \frac{3}{5}ln\frac{3}{5}\right) = 0.971$$
$$J(S_{12}) = -\frac{1}{ln(2)}\left(\frac{1}{3}ln\frac{1}{3} + \frac{2}{3}ln\frac{2}{3}\right) = 0.918$$
$$E_S(C_1) = \frac{5}{8}J(S_{11}) + \frac{3}{8}J(S_{12}) = 0.951$$

For property (C_2),

$$x_{21}^{+} = \frac{0}{3} \qquad x_{21}^{-} = \frac{3}{5}$$
$$x_{22}^{+} = \frac{1}{3} \qquad x_{22}^{-} = \frac{0}{5}$$
$$x_{23}^{+} = \frac{2}{3} \qquad x_{23}^{-} = \frac{2}{5}$$

$$E_D(C_2) = \frac{1}{1 + \frac{3}{8}\frac{1-\frac{3}{5}}{\frac{3}{5}} + \frac{5}{8}\frac{1-\frac{0}{3}}{\frac{0}{3}}} + \frac{1}{1 + \frac{3}{8}\frac{1-\frac{0}{5}}{\frac{0}{5}} + \frac{5}{8}\frac{1-\frac{1}{3}}{\frac{1}{3}}} + \frac{1}{1 + \frac{3}{8}\frac{1-\frac{2}{5}}{\frac{2}{5}} + \frac{5}{8}\frac{1-\frac{2}{3}}{\frac{2}{3}}} = \frac{8}{15} = 0.53$$

$E_S(C_2) = 0.5$

For property (C_3),

$$x_{31}^{+} = \frac{0}{3} \qquad x_{31}^{-} = \frac{3}{5}$$
$$x_{32}^{+} = \frac{3}{3} \qquad x_{32}^{-} = \frac{2}{5}$$

$$E_D(C_3) = \frac{1}{1 + \frac{3}{8}\frac{1-\frac{3}{5}}{\frac{3}{5}} + \frac{5}{8}\frac{1-\frac{0}{3}}{\frac{0}{3}}} + \frac{1}{1 + \frac{3}{8}\frac{1-\frac{2}{5}}{\frac{2}{5}} + \frac{5}{8}\frac{1-\frac{3}{3}}{\frac{3}{3}}} = \frac{16}{25} = 0.64$$

$E_S(C_3) = 0.607$

In Table 2 we show that $E_D(C_k)$ is equivalent to $E_S(C_k)$. Because $E_D(C_2)$ is the minimum, the decision tree is:

Table 2: Equivalence of E_D and E_S

	E_D	E_S
C_1	0.996	0.951
C_2	0.530	0.500
C_3	0.640	0.607

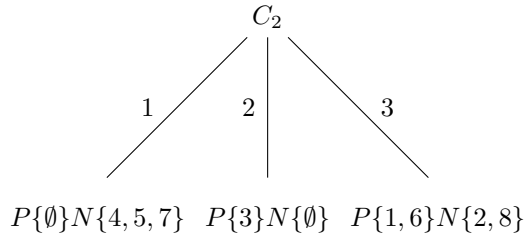

Figure 4: The decision tree after the first step of the calculation

The data table is effectively reduced to:

	C_1	C_3	R
1	B	b	+
2	A	a	-
6	B	b	+
8	A	b	-

With a similar calculation, we can get the minimum for S_3.

So the decision tree looks like this:

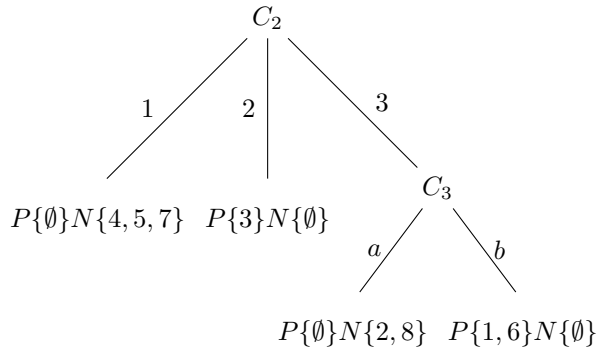

Figure 5: The final decision tree

5 A fast calculation of the PDT

Let C_k be a set of properties in the classical case.

We can represent it by a binary-valued vector

$$C_k = \{S_{k_1}, S_{k_2} \dots S_{k_l} \dots S_{k_{n_k}}\}$$

The response vector can also be divided into $\mathbf{r}^+$ and $\mathbf{r}^-$ values, i.e.

$$\begin{aligned} r_l &= 1 \qquad \text{if positive in the response} \quad (+) \\ r_l &= 0 \qquad \text{if negative in the response} \quad (-) \end{aligned}$$

So the database could be written in the following form:

Table 3: The original database

	$\overbrace{\qquad\qquad}^{C_1}$				$\overbrace{\qquad\qquad}^{C_k}$			$\overbrace{\qquad}^{R}$	
	$S_{1,1}$	$S_{1,2}$	$\dots$ S_{1,n_1}	$\cdots$	$S_{k,1}$	$S_{k,2}$	$\dots$ S_{k,n_k}	R^+	R^-
1	$x_{1,1}^{(1)}$	$x_{1,2}^{(1)}$	$\dots$ $x_{1,n_1}^{(1)}$	$\cdots$	$x_{k,1}^{(1)}$	$x_{k,2}^{(1)}$	$\dots$ $x_{k,n_k}^{(1)}$	$r_+^{(1)}$	$r_-^{(1)}$
2	$x_{1,1}^{(2)}$	$x_{1,2}^{(2)}$	$\dots$ $x_{1,n_1}^{(2)}$	$\cdots$	$x_{k,1}^{(2)}$	$x_{k,2}^{(2)}$	$\dots$ $x_{k,n_k}^{(2)}$	$r_+^{(2)}$	$r_-^{(2)}$
3	$x_{1,1}^{(3)}$	$x_{1,2}^{(3)}$	$\dots$ $x_{1,n_1}^{(3)}$	$\cdots$	$x_{k,1}^{(3)}$	$x_{k,2}^{(3)}$	$\dots$ $x_{k,n_k}^{(3)}$	$r_+^{(3)}$	$r_-^{(3)}$
$\vdots$		$\vdots$				$\vdots$		$\vdots$	$\vdots$
m	$x_{1,1}^{(m)}$	$x_{1,2}^{(m)}$	$\dots$ $x_{1,n_1}^{(m)}$	$\cdots$	$x_{k,1}^{(m)}$	$x_{k,2}^{(m)}$	$\dots$ $x_{k,n_k}^{(m)}$	$r_+^{(m)}$	$r_-^{(m)}$
$\sum$								$\lvert S^+\rvert$	$\lvert S^-\rvert$

where $(x_{k,1}^{(j)}, x_{k,2}^{(j)}, \dots, x_{k,l}^{(j)}, \dots, x_{k,n_k}^{(j)}) = (0, 0, \dots, 1, \dots, 0)$, where 1 is in the l-th coordinate of the vector.

In this database $(x_{k,1}^{(j)}, x_{k,2}^{(j)}, \ldots, x_{k,n_k}^{(j)}) = (0, \ldots, 0, 1, 0, \ldots, 0)$, where the 1 is in the (l)-th place if the (j)-th example of the C_k attribute is S_{k_l}. The weights are:

$$w^+ = \frac{|S^+|}{|S|}, \qquad w^- = \frac{|S^-|}{|S|}.$$

For a fast calculation of x_i^+ and x_i^-, we have to multiply the columns $\mathbf{R}^+$ and $\mathbf{R}^-$ by the columns of the table.

Multiplying $\mathbf{S_{k,j}} = \begin{pmatrix} S_{k,j}^{(1)} \\ \vdots \\ S_{k,j}^{(m)} \end{pmatrix}$ by $\mathbf{R}^+ = \begin{pmatrix} r_+^{(1)} \\ \vdots \\ r_+^{(m)} \end{pmatrix}$ componentwise, we get

$$\mathbf{R}^+(\mathbf{S_{k,j}}) = \begin{pmatrix} S_{k,j}^{+(1)} \\ \vdots \\ S_{k,j}^{+(m)} \end{pmatrix} = \begin{pmatrix} S_{k,j}^{(1)} r_+^{(1)} \\ \vdots \\ S_{k,j}^{(m)} r_+^{(m)} \end{pmatrix}.$$

After calculating these new vectors, we get the following table (see Table 4), where

$$x_{k,j}^+ = \frac{|S_{k,j}^+|}{|S^+|}. \qquad (j = 1, \ldots, n_k)$$

Table 4: Multiplying the columns by $\mathbf{R}^+$

	$\overbrace{\mathbf{R}^+(\mathbf{S_{1,1}})\ \mathbf{R}^+(\mathbf{S_{1,2}})\ \ldots\ \mathbf{R}^+(\mathbf{S_{1,n_1}})}^{C_1}$				$\cdots$	$\overbrace{\mathbf{R}^+(\mathbf{S_{k,1}})\ \mathbf{R}^+(\mathbf{S_{k,2}})\ \ldots\ \mathbf{R}^+(\mathbf{S_{k,n_k}})}^{C_k}$			
1	$S_{1,1}^{+(1)}$	$S_{1,2}^{+(1)}$	$\ldots$	$S_{1,n_1}^{+(1)}$	$\cdots$	$S_{k,1}^{+(1)}$	$S_{k,2}^{+(1)}$	$\ldots$	$S_{k,n_k}^{+(1)}$
2	$S_{1,1}^{+(2)}$	$S_{1,2}^{+(2)}$	$\ldots$	$S_{1,n_1}^{+(2)}$	$\cdots$	$S_{k,1}^{+(2)}$	$S_{k,2}^{+(2)}$	$\ldots$	$S_{k,n_k}^{+(2)}$
3	$S_{1,1}^{+(3)}$	$S_{1,2}^{+(3)}$	$\ldots$	$S_{1,n_1}^{+(3)}$	$\cdots$	$S_{k,1}^{+(3)}$	$S_{k,2}^{+(3)}$	$\ldots$	$S_{k,n_k}^{+(3)}$
$\vdots$		$\vdots$					$\vdots$		
m	$S_{1,1}^{+(m)}$	$S_{1,2}^{+(m)}$	$\ldots$	$S_{1,n_1}^{+(m)}$	$\cdots$	$S_{k,1}^{+(m)}$	$S_{k,2}^{+(m)}$	$\ldots$	$S_{k,n_k}^{+(m)}$
$\sum$	$S_{1,1}^{+}$	$S_{1,2}^{+}$	$\ldots$	S_{1,n_1}^{+}	$\cdots$	$S_{k,1}^{+}$	$S_{k,2}^{+}$	$\ldots$	S_{k,n_k}^{+}
x^+	$x_{1,1}^{+}$	$x_{1,2}^{+}$	$\ldots$	x_{1,n_1}^{+}	$\cdots$	$x_{k,1}^{+}$	$x_{k,2}^{+}$	$\ldots$	x_{k,n_k}^{+}

Similarly, multiplying $\mathbf{S_{k,j}} = \begin{pmatrix} S_{k,j}^{(1)} \\ \vdots \\ S_{k,j}^{(m)} \end{pmatrix}$ by $\mathbf{R}^- = \begin{pmatrix} r_-^{(1)} \\ \vdots \\ r_-^{(m)} \end{pmatrix}$ componentwise, we get

$$\mathbf{R}^-(\mathbf{S_{k,j}}) = \begin{pmatrix} S_{k,j}^{-(1)} \\ \vdots \\ S_{k,j}^{-(m)} \end{pmatrix} = \begin{pmatrix} S_{k,j}^{(1)} r_-^{(1)} \\ \vdots \\ S_{k,j}^{(m)} r_-^{(m)} \end{pmatrix}.$$

, where

$$x_{k,j}^- = \frac{|S_{k,j}^-|}{|S^-|}. \qquad (j = 1, \ldots, n_k)$$

Table 5: Multiplying the columns by $\mathbf{R}^-$

	$\overbrace{\qquad\qquad\qquad\qquad}^{C_1}$					$\overbrace{\qquad\qquad\qquad\qquad}^{C_k}$			
	$\mathbf{R}^-(\mathbf{S_{1,1}})$	$\mathbf{R}^-(\mathbf{S_{1,2}})$	$\ldots$	$\mathbf{R}^-(\mathbf{S_{1,n_1}})$	$\cdots$	$\mathbf{R}^-(\mathbf{S_{k,1}})$	$\mathbf{R}^-(\mathbf{S_{k,2}})$	$\ldots$	$\mathbf{R}^-(\mathbf{S_{k,n_k}})$
1	$S_{1,1}^{-(1)}$	$S_{1,2}^{-(1)}$	$\ldots$	$S_{1,n_1}^{-(1)}$	$\cdots$	$S_{k,1}^{-(1)}$	$S_{k,2}^{-(1)}$	$\ldots$	$S_{k,n_k}^{-(1)}$
2	$S_{1,1}^{-(2)}$	$S_{1,2}^{-(2)}$	$\ldots$	$S_{1,n_1}^{-(2)}$	$\cdots$	$S_{k,1}^{-(2)}$	$S_{k,2}^{-(2)}$	$\ldots$	$S_{k,n_k}^{-(2)}$
3	$S_{1,1}^{-(3)}$	$S_{1,2}^{-(3)}$	$\ldots$	$S_{1,n_1}^{-(3)}$	$\cdots$	$S_{k,1}^{-(3)}$	$S_{k,2}^{-(3)}$	$\ldots$	$S_{k,n_k}^{-(3)}$
$\vdots$		$\vdots$					$\vdots$		
m	$S_{1,1}^{-(m)}$	$S_{1,2}^{-(m)}$	$\ldots$	$S_{1,n_1}^{-(m)}$	$\cdots$	$S_{k,1}^{-(m)}$	$S_{k,2}^{-(m)}$	$\ldots$	$S_{k,n_k}^{-(m)}$
$\sum$	$S_{1,1}^-$	$S_{1,2}^-$	$\ldots$	S_{1,n_1}^-	$\cdots$	$S_{k,1}^-$	$S_{k,2}^-$	$\ldots$	S_{k,n_k}^-
x^-	$x_{1,1}^-$	$x_{1,2}^-$	$\ldots$	x_{1,n_1}^-	$\cdots$	$x_{k,1}^-$	$x_{k,2}^-$	$\ldots$	x_{k,n_k}^-

Hence, the input values of the algorithm are given. Now we can use (13) or the weighted conjunctive operator of Dombi (15) to calculate the entropies.

Example 2.: Fast calculation of PTD.
The database in the example could be written in the following way:

Table 6: based on Table 1

	C_1		C_2			C_3		R	
	A	B	1	2	3	a	b	R^+	R^-
1	0	1	0	0	1	0	1	1	0
2	1	0	0	0	1	1	0	0	1
3	1	0	0	1	0	0	1	1	0
4	0	1	1	0	0	0	1	0	1
5	1	0	1	0	0	0	1	0	1
6	1	0	0	0	1	0	1	1	0
7	1	0	1	0	0	1	0	0	1
8	0	1	0	0	1	1	0	0	1
$\sum$								$3^{(+)}$	$5^{(-)}$

So

$$w^+ = \frac{3}{8}, \qquad w^- = \frac{5}{8}.$$

For a fast calculation of x_i^+ and x_i^-, we have to multiply columns $\mathbf{R}^+$ and $\mathbf{R}^-$ by the columns of the tables!.

Multiplying by $\mathbf{R}^+$, we get:

Table 7: based on Table 2

	C_1		C_2			C_3	
	$\mathbf{R}^+(\mathbf{A})$	$\mathbf{R}^+(\mathbf{B})$	$\mathbf{R}^+(\mathbf{1})$	$\mathbf{R}^+(\mathbf{2})$	$\mathbf{R}^+(\mathbf{3})$	$\mathbf{R}^+(\mathbf{a})$	$\mathbf{R}^+(\mathbf{b})$
1	0	1	0	0	1	0	1
2	0	0	0	0	0	0	0
3	1	0	0	1	0	0	1
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	1	0	0	0	0	0	1
7	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0
$\sum$	2	1	0	1	2	0	3
x^+	$\frac{2}{3^{(+)}}$	$\frac{1}{3^{(+)}}$	$\frac{0}{3^{(+)}}$	$\frac{1}{3^{(+)}}$	$\frac{2}{3^{(+)}}$	$\frac{0}{3^{(+)}}$	$\frac{3}{3^{(+)}}$

Multiplying by $\mathbf{R}^-$, we get:

Table 8: based on Table 3

	C_1		C_2			C_3	
	$\mathbf{R}^-(\mathbf{A})$	$\mathbf{R}^-(\mathbf{B})$	$\mathbf{R}^-(\mathbf{1})$	$\mathbf{R}^-(\mathbf{2})$	$\mathbf{R}^-(\mathbf{3})$	$\mathbf{R}^-(\mathbf{a})$	$\mathbf{R}^-(\mathbf{b})$
1	0	0	0	0	0	0	0
2	1	0	0	0	1	1	0
3	0	0	0	0	0	0	0
4	0	1	1	0	0	0	1
5	1	0	1	0	0	0	1
6	0	0	0	0	0	0	0
7	1	0	1	0	0	1	0
8	0	1	0	0	1	1	0
$\sum$	3	2	3	0	2	3	2
x^-	$\frac{3}{5^{(-)}}$	$\frac{2}{5^{(-)}}$	$\frac{3}{5^{(-)}}$	$\frac{0}{5^{(-)}}$	$\frac{2}{5^{(-)}}$	$\frac{3}{5^{(-)}}$	$\frac{2}{5^{(-)}}$

So we get:

$$E_D(C_1) = \frac{1}{1+\frac{3}{8}\frac{1-\frac{3}{5}}{\frac{3}{5}}+\frac{5}{8}\frac{1-\frac{2}{3}}{\frac{2}{3}}} + \frac{1}{1+\frac{3}{8}\frac{1-\frac{2}{5}}{\frac{2}{5}}+\frac{5}{8}\frac{1-\frac{1}{3}}{\frac{1}{3}}} = \frac{224}{225} = 0.9955$$

$$E_D(C_2) = \frac{1}{1+\frac{3}{8}\frac{1-\frac{3}{5}}{\frac{3}{5}}+\frac{5}{8}\frac{1-\frac{0}{3}}{\frac{0}{3}}} + \frac{1}{1+\frac{3}{8}\frac{1-\frac{0}{5}}{\frac{0}{5}}+\frac{5}{8}\frac{1-\frac{1}{3}}{\frac{1}{3}}}\frac{1}{1+\frac{3}{8}\frac{1-\frac{2}{5}}{\frac{2}{5}}+\frac{5}{8}\frac{1-\frac{2}{3}}{\frac{2}{3}}} = \frac{8}{15} = 0.53$$

$$E_D(C_3) = \frac{1}{1+\frac{3}{8}\frac{1-\frac{3}{5}}{\frac{3}{5}}+\frac{5}{8}\frac{1-\frac{0}{3}}{\frac{0}{3}}} + \frac{1}{1+\frac{3}{8}\frac{1-\frac{2}{5}}{\frac{2}{5}}+\frac{5}{8}\frac{1-\frac{3}{3}}{\frac{3}{3}}} = \frac{16}{25} = 0.64$$

From this since we want to minimize the vagueness we would choose $C2$, however first lets examine the problem from another perspective.
We can split $C2$ into three different criterion's in a one versus all manner so the interpretation of the new criterions would be 1 or not one, for the second 2 or not 2 and so on. We are able to do this because in $C_k = \{S_{k,i}\}\ \sum_{i=1}^{N} r_{k,i}^l = 1$. After splitting the previous dataset and recalculating $E_D(C_k)$ we get the following results, since C_1 and C_5(previously C_3) remains the same we will not show these. The results are the following.

$$E_D(C_2) = 0.64 \quad E_D(C_3) = 0.76 \quad E_D(C_4) = 0.9333 \tag{19}$$

As we can see after splitting C_2 the new criterions vagueness is higher than the previous C_2's and now we can either select C_3 or C_5 since both of them has the same vagueness. This means that we choose criterions with more attributes, to avoid this we will have to split the tree according to previous steps. In the following we will introduce a more efficient equation for this kind of trees.

5.1 Modifying the Vagueness measure

We have the equation

$$K = \frac{X_1^+ X_1^-}{w^+ X_1^+ + w^- X_1^-} + \frac{X_2^+ X_2^-}{w^+ X_2^+ + w^- X_2^-}$$

Since

$$X_2^+ = 1 - X_1^+ \qquad X_2^- = 1 - X_1^-$$

after substituting it into the equation we get

$$K = \frac{X^+X^-}{w^+X^+ + w^-X^-} + \frac{(1-X^+)(1-X^-)}{w^+(1-X^+) + w^-(1-X^-)}$$

. We can reformulate the followings

$$\sum_i^N r_i^+ = R \qquad \sum_i^N (1 - r_i^+) = N - R$$

$$w^+ = \frac{R}{N} \qquad w^- = \frac{N-R}{N}$$

$$X^+ = \frac{1}{R}\sum a_i r_i \qquad X^- = \frac{1}{N-R}\sum a_i(1-r_i)$$

$$w^+X^+ = \frac{R}{N}\frac{1}{R}\sum a_i r_i \qquad w^-X^- = \frac{N-R}{N}\frac{1}{N-R}\sum a_i(1-r_i)$$

$$w^+X^+ + w^-X^- = \frac{1}{N}\sum a_i$$

$$K_1 = \frac{N}{R^+R^-}\frac{(\sum a_i r_i)(\sum a_i(1-r_i))}{\sum a_i}$$

Now we transform the second half of the equation too,

$$K_2 = \frac{(1-X^+)(1-X^-)}{w^+(1-X^+) + w^-(1-X^-)}$$

$$w^+(1-X^+) = \frac{R}{N}\left(1 - \frac{1}{R}\sum a_i r_i\right)$$

$$w^-(1-X^-) = \frac{N-R}{N}\left(1 - \frac{1}{N-R}\sum a_i(1-r_i)\right)$$

$$= \frac{1}{N}\left(R^+ - \sum a_i r_i\right) + \frac{1}{N}\left(R^- - \sum a_i(1-r_i)\right) = \frac{1}{N}\left(R^+ + R^- - \sum a_i\right) = \frac{1}{N}\left(N - \sum a_i\right)$$

So the second half of the equation is

$$N\frac{(1 - \frac{1}{R^+}\sum a_i r_i)(1 - \frac{1}{R^-}\sum a_i(1-r_i))}{N - \sum a_i} = \frac{N}{R^+R^-}\frac{(R^+ - \sum a_i r_i)(R^- - \sum a_i(1-r_i))}{N - \sum a_i}$$

and the whole equation

$$K = \frac{N}{R^+R^-}\frac{(\sum a_i r_i)(\sum a_i(1-r_i))}{\sum a_i} + \frac{(R^+ - \sum a_i r_i)(R^- - \sum a_i(1-r_i))}{N - \sum a_i}$$

Lets use the following notations

$$A = \sum a_i r_i \qquad B = \sum a_i(1-r_i) = Z - A \qquad Z = \sum a_i$$

then we get the final equation.

$$K = \frac{N}{R^+R^-}\left(\frac{AB}{Z} + \frac{(R^+ - A)(R^- - B)}{N - Z}\right) \tag{20}$$

Table 9: Transformed database

	C_1	C_2	C_3	C_4	C_5	R
1	0	0	0	1	0	1
2	1	0	0	1	1	0
3	1	0	1	0	0	1
4	0	1	0	0	0	0
5	1	1	0	0	0	0
6	1	0	0	1	0	1
7	1	1	0	0	1	0
8	0	0	0	1	1	0
$\sum$	5	3	1	4	3	3

5.2 Example

We will show the usage of the new method through an example. First we transform the original database into a simplified version. We use the table6.1 and transform the database so that every attribute of each criterion is a separate criterion then we get the following table:
We then calculate the value of the constants:

$$N = 8 \qquad w^+ = \frac{R}{N} = \frac{3}{8} \qquad w^- = \frac{N-R}{N} = \frac{5}{8}$$

Then we calculate the K for each criterion:

$$C_1 \quad Z = 5 \quad A = 2 \quad B = 3 \quad K_1 = \frac{8}{15}\left(\frac{6}{5} + \frac{(3-2)(5-3)}{8-5}\right) = 0.9955$$

$$C_2 \quad Z = 3 \quad A = 0 \quad B = 3 \quad K_2 = \frac{8}{15}\left(\frac{0}{5} + \frac{(3-0)(5-3)}{8-3}\right) = 0.64$$

$$C_3 \quad Z = 1 \quad A = 1 \quad B = 0 \quad K_3 = \frac{8}{15}\left(\frac{0}{1} + \frac{(3-1)(5-0)}{8-1}\right) = 0.76$$

$$C_4 \quad Z = 4 \quad A = 2 \quad B = 2 \quad K_4 = \frac{8}{15}\left(\frac{4}{4} + \frac{(3-2)(5-2)}{8-4}\right) = 0.9333$$

$$C_5 \quad Z = 3 \quad A = 0 \quad B = 3 \quad K_5 = \frac{8}{15}\left(\frac{0}{3} + \frac{(3-0)(5-3)}{8-3}\right) = 0.64$$

As we can see we got the same results as with the equation 17.

6 PDT with multi-class outputs

In the real world there are several cases with multiple possible output classes. To make it possible for our PDT to handle such cases we are going to use a one versus all approach. When we have more than two output classes we are going to divide the database into $j-1$ separate database where j is the number of possible output class, and build a tree on each one of them. This way every tree makes a decision on whether our example is the selected class or not.

6.1 Example

Table 10: based on Table 1

	C_1	C_2	C_3	C_4	C_5	R_1	R_2	R_3
1	0	0	0	1	0	1	0	0
2	1	0	0	1	1	0	0	1
3	1	0	1	0	0	1	0	0
4	0	1	0	0	0	0	1	0
5	1	1	0	0	0	0	0	1
6	1	0	0	1	0	1	0	0
7	1	1	0	0	1	0	1	0
8	0	0	0	1	1	0	0	1
$\sum$						3	2	3

Now as we discussed before we will separate the dataset.

Table 11: Updated database

	C_1	C_2	C_3	C_4	C_5	R
1	0	0	0	1	0	1
2	1	0	0	1	1	0
3	1	0	1	0	0	1
4	0	1	0	0	0	0
5	1	1	0	0	0	0
6	1	0	0	1	0	1
7	1	1	0	0	1	0
8	0	0	0	1	1	0
$\sum$						3

First lets calculate the K values for this dataset.

$$K_1 = 0.995 \qquad K_2 = 0.64 \qquad K_3 = 0.76 \qquad K_4 = 0.933 \qquad K_5 = 0.64$$

We will choose the C_5 as the new node's criterion.

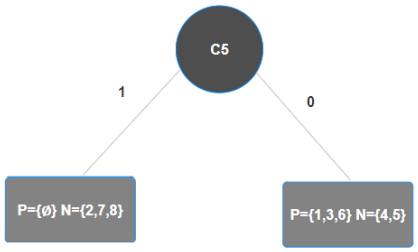

Figure 6: Tree after the first step

Then we get: The K values:

Table 12: Updated database

	C_1	C_2	C_3	C_4	R
1	0	0	0	1	1
3	1	0	1	0	1
4	0	1	0	0	0
5	1	1	0	0	0
6	1	0	0	1	1
$\sum$					3

$$K_1 = 0.972 \qquad K_2 = 0 \qquad K_3 = 0.83 \qquad K_4 = 0.55$$

Based on these we chose C_2 and the new tree is: for R_1 this will be the final tree now we will build the

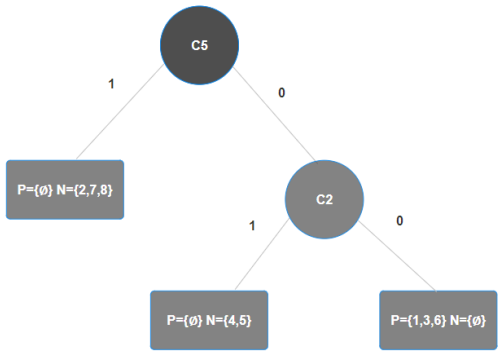

Figure 7: The tree after the second iteration

second tree based on the new dataset, for this we only have to consider the elements where the first tree is negative.

Table 13: Updated database

	C_1	C_2	C_3	C_4	C_5	R_2	R_3
2	1	0	0	1	1	0	1
4	0	1	0	0	0	1	0
5	1	1	0	0	0	0	1
7	1	1	0	0	1	1	0
8	0	0	0	1	1	0	1
$\sum$						2	3

$$K_1 = 0.97 \qquad K_2 = 0.55 \qquad K_3 = 0 \qquad K_4 = 0.55 \qquad K_5 = 0.97$$

We would choose the C_3 based on the equation however since there are no different values in this criterion it would not help us progress the tree so since it does not contain any useful information we can get rid of it.

Based on the new tree we will get a reduced database containing 4,5,7.

Table 14: Updated database

	C_1	C_2	C_4	C_5	R_2	R_3
2	1	0	1	1	0	1
4	0	1	0	0	1	0
5	1	1	0	0	0	1
7	1	1	0	1	1	0
8	0	0	1	1	0	1
$\sum$					2	3

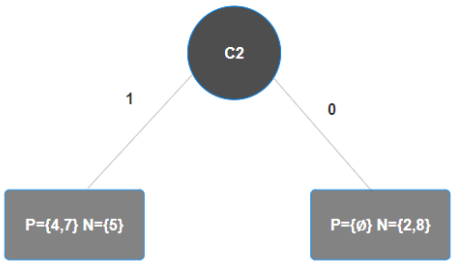

Figure 8: The new tree after the first iteration

Table 15: Updated database

	C_1	C_5	R_2	R_3
4	0	0	1	0
5	1	0	0	1
7	1	1	1	0
$\sum$			2	3

Note that we deleted C_4 too since it only contained 0s.

$$K_1 = 0.75 \qquad K_2 = 0.75$$

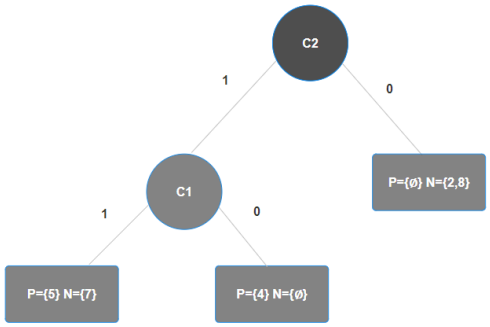

Figure 9: The new tree after the second iteration

And the final criterion will be the C_5 and the tree is:

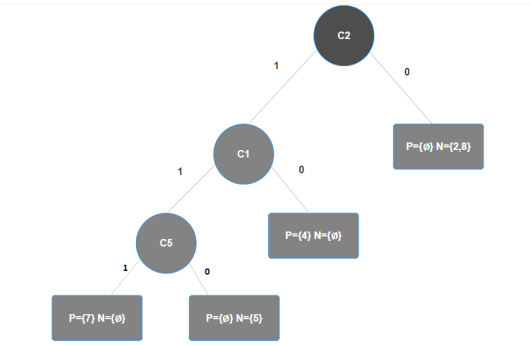

Figure 10: The new tree after the third iteration

Since we already checked the two other output classes we do not have to check the last one as after the second tree all negative examples will belong to the last class.

7 The PDT when attributes are probability values

In a real world problem it is sometimes not easy to calculate the attributes of a property.

In the following we will use the same representation as that in the fast calculation case:

$$C_k = \{\alpha_1^k, \alpha_2^k \dots \alpha_{k_n}^k\}$$

and if it belongs to the l-th example, then

$$C_{l_k} = \{\alpha_{l_1}^k, \alpha_{l_2}^k \dots \alpha_{l_{k_n}}^k\}$$

So it is easy to define the probabilistic values for C_{l_k}. And we will suppose that for all the $\alpha_{l_i}^k$ probability values.

$$0 \leq \alpha_{l_i}^k \leq 1$$

$$\sum_{i=1}^{n_k} \alpha_{l_i}^k = 1$$

for all values of k and l.

Example: C_k = fever. The attributes are: no fever, fever, high fever. In a certain case we measure a temperature of 37.1°C. Then the C_{l_k} vector might be:

$$C_{l_k} = (0.3, 0.6, 0.1)$$

If $\alpha_{l_i}^k$ have no probabilistic values, then $\alpha_{l_i}^k \epsilon \{0, 1\}$.

Let us suppose that C_k is the minimum entropy (17). Now we shall use the values of

$$S_{k,j}^{+(i)} \text{ and } S_{k,j}^{-(i)}, \quad \text{where} \quad S_{k,j}^{+(i)} + S_{k,j}^{-(i)} = 1.$$

Here k is the criteria, j is the element of the criteria and i denotes the item. Now we can select the items in the following way.
If $S_{k,j}^{+(i)}$ is greater than $\frac{1}{2}$, then it is a positive item; otherwise it is a negative item. So we get the two sets, namely P and N.
We have to calculate $|S_{ki}^+|, |S_{ki}^-|$.
So

$$|S_{ki}^+| = \sum_{i=1}^{M} r_+^{(i)} \alpha_{l_i}^k$$

$$|S_{ki}^-| = \sum_{i=1}^{M} r_-^{(i)} \alpha_{l_i}^k$$

All the other steps of the algorithm are similar to the fast calculation procedure.

Example 3.: When the attributes are probability values

Database:

	C_1		C_2			C_3		R	
	A	B	1	2	3	a	b	R^+	R^-
1	0.4	0.6	0.1	0.1	0.8	0.0	1.0	1	0
2	0.6	0.4	0.3	0.3	0.4	1.0	0.0	0	1
3	0.7	0.3	0.0	1.0	0.0	0.0	1.0	1	0
4	0.3	0.7	0.9	0.1	0.0	0.0	1.0	0	1
5	0.8	0.2	0.8	0.2	0.0	0.0	1.0	0	1
6	0.8	0.2	0.2	0.2	0.6	0.0	1.0	1	0
7	0.7	0.3	0.4	0.3	0.3	1.0	0.0	0	1
8	0.1	0.9	0.0	0.0	1.0	1.0	0.0	0	1

We can calculate this in a similar way as we did in Example 1.
For property (C_1), we get

$$x_{11}^{+} = \frac{1.9}{3} \qquad x_{11}^{-} = \frac{2.5}{5}$$
$$x_{12}^{+} = \frac{1.1}{3} \qquad x_{12}^{-} = \frac{2.5}{5}$$

$$E(C_1) = \frac{1}{1 + \frac{3}{8}\frac{1-\frac{2.5}{5}}{\frac{2.5}{5}} + \frac{5}{8}\frac{1-\frac{1.9}{3}}{\frac{1.9}{3}}} + \frac{1}{1 + \frac{3}{8}\frac{1-\frac{2.5}{5}}{\frac{2.5}{5}} + \frac{5}{8}\frac{1-\frac{1.1}{3}}{\frac{1.1}{3}}} = \frac{292}{297} = 0.9832$$

For property (C_2), we get

$$x_{21}^{+} = \frac{0.3}{3} \qquad x_{21}^{-} = \frac{2.4}{5}$$
$$x_{22}^{+} = \frac{1.3}{3} \qquad x_{22}^{-} = \frac{0.9}{5}$$
$$x_{23}^{+} = \frac{1.4}{3} \qquad x_{23}^{-} = \frac{1.7}{5}$$

$$E(C_2) = \frac{1}{1 + \frac{3}{8}\frac{1-\frac{2.4}{5}}{\frac{2.4}{5}} + \frac{5}{8}\frac{1-\frac{0.3}{3}}{\frac{0.3}{3}}} + \frac{1}{1 + \frac{3}{8}\frac{1-\frac{0.9}{5}}{\frac{0.9}{5}} + \frac{5}{8}\frac{1-\frac{1.3}{3}}{\frac{1.3}{3}}} + \frac{1}{1 + \frac{3}{8}\frac{1-\frac{1.7}{5}}{\frac{1.7}{5}} + \frac{5}{8}\frac{1-\frac{1.4}{3}}{\frac{1.4}{3}}} = 0.8353$$

For property (C_3), we get

$$x_{31}^{+} = \frac{0}{3} \qquad x_{31}^{-} = \frac{3}{5}$$
$$x_{32}^{+} = \frac{3}{3} \qquad x_{32}^{-} = \frac{2}{5}$$

$$E(C_3) = \frac{1}{1 + \frac{3}{8}\frac{1-\frac{3}{5}}{\frac{3}{5}} + \frac{5}{8}\frac{1-\frac{0}{3}}{\frac{0}{3}}} + \frac{1}{1 + \frac{3}{8}\frac{1-\frac{2}{5}}{\frac{2}{5}} + \frac{5}{8}\frac{1-\frac{3}{3}}{\frac{3}{3}}} = \frac{16}{25} = 0.64$$

In this case C_3 has a minimum value, so the decision tree is

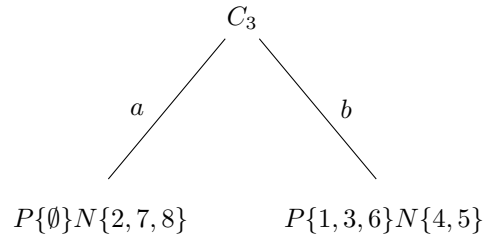

Figure 11: The decision tree after the first step of the calculation

We then get the reduced database:

	C_1		C_2			R^+	R^-
1	0.4	0.6	0.1	0.1	0.8	1	0
3	0.7	0.3	0.0	1.0	0.0	1	0
4	0.3	0.7	0.9	0.1	0.0	0	1
5	0.8	0.2	0.8	0.2	0.0	0	1
6	0.8	0.2	0.2	0.2	0.6	1	0

After a similar calculation, we find that C_2 has the minimum value. Setting the threshold to 0.5, we get the final decision tree

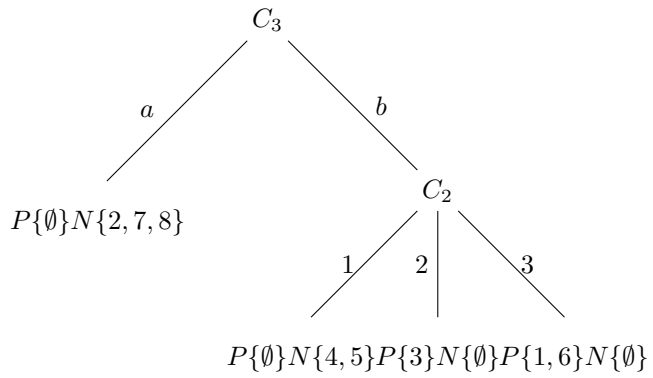

Figure 12: The resulting decision tree

References

[Bre+84] L. Breiman et al. *Classification and Regression Trees*. Belmont, CA: Wadsworth, 1984.

[Qui93] J. Quinlan. *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgam Kaufmann, 1993.

[KV94] M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. Cambridge, Massachusetts: The MIT Press, 1994.

[RN95] S. Russel and P. Norvig. *Artificial Intelligence - A Modern Approach*. Englewood Cliffs: Prentice-Hall, 1995.

[Mit97] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[CL] K. J. Cios and N. Liu. "A machine learning method for generation of a neural network achitecture: a continuous ID3 algorithm". In: 3, no. 2, pp. 280-291, March 1992 ().

[DT] A. DeLuca and S. Termini. "A definition of non-probabilistic entropy in the setting of fuzzy sets theory". In: *Inform and Control* 20, pp. 301-312, 1972 ().

[HR] L. Hyafil and R. Rivest. "Constructing optimal binary decision trees is np-complete". In: 5, pp. 15-17, 1976 ().

[MKS] S. Murthy, S. Kasif, and S. Salzberg. "A system for induction of oblique decision trees". In: 2, pp. 1-32, 1994 ().

[Ste] J. Stefandowski. "Classification and decision supporting based on rough set theory". In: *Foundations of Computing and Decision Sciences* 18, no. 3-4, pp. 371-380, 1993 ().

[Vet] R. Vetschera. "Entropy and the value of information". In: *Central European Journal of Operations Research* 8, pp. 195-208, 2000 ().