A Compact and Accurate Model for Classification

Article *in* IEEE Transactions on Knowledge and Data Engineering · March 2004

DOI: 10.1109/TKDE.2004.1269598 · Source: IEEE Xplore

2 authors:

Mark Last
Ben-Gurion University of the Negev
211 PUBLICATIONS **3,131** CITATIONS
SEE PROFILE

Oded Maimon
Tel Aviv University
232 PUBLICATIONS **5,918** CITATIONS
SEE PROFILE

A Compact and Accurate Model for Classification

Mark Last[1]
Department of Information Systems Engineering, Ben-Gurion University of the Negev,
Beer-Sheva 84105, Israel
E-mail: mlast@bgumail.bgu.ac.il

Oded Maimon
Department of Industrial Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel
E-mail: maimon@eng.tau.ac.il

Abstract

We describe and evaluate an information-theoretic algorithm for data-driven induction of classification models based on a minimal subset of available features. The relationship between input (predictive) features and the target (classification) attribute is modeled by a tree-like structure termed an *information network* (*IN*). Unlike other decision-tree models, the information network uses the same input attribute across the nodes of a given layer (level). The input attributes are selected incrementally by the algorithm to maximize a global decrease in the conditional entropy of the target attribute. We are using the pre-pruning approach: when no attribute causes a statistically significant decrease in the entropy, the network construction is stopped. The algorithm is shown empirically to produce much more compact models than other methods of decision-tree learning, while preserving nearly the same level of classification accuracy.

Keywords: *Knowledge Discovery in Databases, Data mining, Classification, Dimensionality reduction, Feature selection, Decision Trees, Information theory, Information theoretic network.*

[1] Corresponding author

1. Introduction

The process of *Knowledge Discovery in Databases* (KDD) is defined by Fayyad *et al.* [8] as "the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data". A pattern is an expression, which describes the data at some level of abstraction. An example of a very simple pattern used by many credit card companies is *most students are profitable customers. Data mining* is the core step of the KDD process, which is concerned with a computationally efficient enumeration of patterns presenting in a database. *Classification* is a primary data mining task aimed at learning a function that classifies a database record into one of several pre-defined classes (e.g., classes of profitable vs. non-profitable customers) based on the values of the record attributes (age, occupation, etc.).

Common classification methods, like backpropagation, Naïve Bayes Classifier, and C4.5, are designed to optimize the predictive performance of the induced model, e.g., its ability to classify correctly new credit card applicants. Other aspects of knowledge discovery, such as validity of discovered patterns, simplicity of representation, and identification of relevant features, are given only secondary consideration by most existing algorithms. Consequently, classification models induced from real-world data tend to be overcomplex, statistically insignificant, and wasteful in the number of used features. The information-theoretic classification method presented in this paper, is aimed at solving these problems by using three guiding principles: maximizing the *mutual information* between a set of predictive attributes and the target (classification) attribute, finding a *minimal set* of database attributes involved in the induced model, and verifying the *statistical significance* of the discovered patterns. The importance of these objectives for the classification task is explained in the next sub-sections.

1.1. Information Theory and Classification

The data classification process is aimed at reducing the amount of uncertainty, or gaining *information*, about the target (classification) attribute. In Shannon's information theory (see [4]), information is defined as that which removes or reduces uncertainty. For a classification task, more information means higher accuracy of a classification model, since the predicted class of new instances is more likely to be identical to their actual class. A model that does not increase the amount of information is useless and its predictive accuracy is not expected to be better than just a random guess. We also realize that more information is needed to predict accurately a multi-valued outcome (e.g., medical diagnosis) than to predict a binary outcome (e.g., customer credibility).

Information theory (see [4]) suggests a general modeling of conditional dependency between random variables. If nothing is known on the causes of a variable *X*, its degree of uncertainty can be measured by the *unconditional entropy* $H(X) = -\Sigma\, p(x) \log_2 p(x)$ (expected value of $\log_2 [1/p(x)]$). The entropy reaches its maximum value of *log [domain size of X]*, when *X* is uniformly distributed in its domain, i.e. each value of *X* has the same probability. Entropy is different from statistical variance by its metric-free nature: it depends only on the probability distribution of a random variable rather than on its concrete values. Thus in classification tasks, where the metric of class labels is unimportant, minimizing the entropy of the target attribute can be a criterion for choosing the best hypothesis. Examples include the use of information gain in ID3 [19] and C4.5 [21] algorithms for finding the best feature to split[2] a node of a decision tree.

According to the information theory, adding information on attributes related to a random variable can decrease its entropy. Moreover, it is shown mathematically in [4] that additional information never increases the entropy. The entropy of a random variable *Y*, given another random variable *X*, (the *conditional entropy*) is given by $H(Y/X) = -\Sigma\, p(x,y) \log p(y/x)$ (expected value of $\log_2 [1/p(y/x)]$). A symmetrical association between two random variables *X* and *Y* (*mutual information)* is defined as a decrease in the entropy of *Y* as a result of knowing *X* and vice versa, namely:

$$I(X;Y) = \sum_{x,y} p(x,y) \bullet \log \frac{p(x/y)}{p(x)} = H(Y) - H(Y/X) = H(X) - H(X/Y) = I(Y;X) \tag{1}$$

where $p\,(x)$ is the unconditional probability of *x*, $p\,(x/y)$ is the conditional probability of *x* given *y*, and $p(x, y)$ is the joint probability of *x* and *y*. The intuition behind the definition of mutual information is as follows. If *Y* and *X* are independent, $\forall x,y$: $p(x) = p(x/y)$, resulting in $I(X; Y) = 0$. In all other cases, the ratio between conditional and unconditional probabilities of *x* will be either below or above 1.00, generating negative and positive terms respectively. Each

[2] To "split" a node in decision-tree learning means to partition the set of training samples associated with a node by creating a child node for each value of the tested feature (see [18]).

non-zero term is weighted by the joint probability of the corresponding value-pair. The above relationship between entropy and mutual information is proven formally in [4]. The decrease in entropy of Y as a result of knowing *n variables* ($X_1,\ldots, X_n$) can be calculated incrementally by using the following chain rule [4]:

$$I(X1,\ldots, Xn; Y) = \Sigma I(Xi;Y / Xi\text{-}1,\ldots, X1) \qquad (2)$$

The information-theoretic methodology of classification, initially introduced by us in [16], is aimed at finding a minimal set of predictive features that maximize a decrease in the entropy of the classification attribute. The method has already been applied to real-world tasks of knowledge discovery in time-series databases [12] and manufacturing data [14]. In this paper, we present, for the first time, a detailed numeric example of the network construction procedure, a new way of extracting rules from the network structure, and a comprehensive comparison of our method to other decision-tree algorithms.

1.2. Dimensionality Reduction and Feature Selection

Minimizing the number of relevant attributes, or features, in a classification model is important for several reasons from increasing the learning speed of a classification algorithm to dealing with the "curse of dimensionality" problem in parameter estimation. John *et al.* [11] distinguish between two models of selecting a “good” set of features under some objective function. The *feature filter* model assumes selecting the features *before* applying an induction algorithm (by using some evaluation measures), while the *wrapper* model uses the prediction accuracy of the induction algorithm itself to evaluate the features. The filter model is in line with the definition of the KDD process in [8]: selection of relevant features is considered a pre-processing step of knowledge discovery. The wrapper approach, on the other hand, is aimed at optimizing the generalization performance of a given data mining algorithm. An overview of existing filter and wrapper methods for feature selection can be found in [15].

The wrapper approach is usually associated with a considerable computational effort, since it requires re-running of an induction algorithm multiple times. The filter methods, on the other hand, are computationally cheaper, but, as indicated by [15], there is a danger that the features selected by a filter method will not allow a classification algorithm to fully exploit its potential. Unlike the filter and the wrapper approaches, the information-theoretic method presented in this paper implements automated feature selection “on the fly” as an *integral part* of the learning process. Thus, a minimal subset of features is found in a *single run* of the induction algorithm.

1.3. Statistical Significance of Classification Models

As indicated by [18], some learning algorithms tend to create very complex models that do not generalize well beyond the set of training examples. The problem of *overfitting* the training set arises whenever the constructed model incorporates some random patterns, which are unlikely to occur in the entire population. An ideal induction algorithm should be able to find every valid pattern presenting in data, while filtering out all the random factors and patterns.

The dilemma of increasing the complexity of a model (e.g., by splitting a decision tree node) vs. the danger of overfitting a given sample is well known in statistical hypothesis testing (see [17]). A *null hypothesis* H_0 (which is usually less complex than the alternative hypothesis H_1) can only be rejected at a given *significance level*, which specifies how rare the training cases must be, based on the assumption that H_0 is true. In other words, the level of significance represents the risk of a researcher in making a decision to reject H_0. One of the first decision-tree algorithms, ID3 [19], has applied the chi-square statistic to the null hypothesis about the irrelevance of a tested attribute. However, most other methods of decision-tree learning, like CART [3] and C4.5 [21], have adopted the post-pruning approach (grow a maximal tree and then prune it) for the sake of exploring a larger set of potentially valid patterns. The post-pruning methods include cost-complexity pruning [3] and pessimistic error pruning [21]. The MDL-based PUBLIC algorithm [24] attempts to save part of the post-pruning effort by implementing a "branch and bound" strategy, which does not expand nodes with guaranteed high encoding cost.

The straightforward approach of the information-theoretic algorithm is to completely *pre-prune* the model by ignoring statistically insignificant features and patterns. Thus, the information-theoretic network is not grown beyond necessity, following the famous *Occam's razor* principle[3]. An additional benefit of the pre-pruning approach is the ability to stop the model construction at any time in the case of limited computation resources. As shown later in this paper, statistical pre-pruning tends to produce more compact models than existing post-pruning techniques, without a substantial decrease in the predictive accuracy.

[3] "Nunquam ponenda est pluralitas sin necesitate": "Entities should not be multiplied beyond necessity" [18]

1.4. Paper Organization

Section 2 describes the structure of the information theoretic connectionist network and presents a detailed numeric example of the network construction procedure. We also analyze the computational complexity of the proposed method as a function of data dimensionality. In Section 3, we compare the information-theoretic methodology to the most common techniques of decision-tree construction and evaluate the algorithm performance on a variety of standard learning tasks. Section 4 concludes the paper with representing a number of issues for future research. More details of the algorithm are given in the Appendix. A beta version of the software is available at http://www.ise.bgu.ac.il/faculty/mlast/ifn.htm.

2. Method Description

2.1. Information-Theoretic Network Structure

Gorin et al. [9][10] have applied an information-theoretic connectionist network, having a fixed number of internal layers, to speech recognition tasks. Gorin used two types of networks: a single-layer network and a two-layer network. A single-layer network has an input node for each input variable and assumes that all variables are independent. The two-layer network has a second ("hidden") layer with a node for each variable-pair. In this paper, we present an algorithm for building a *multi-layer* information-theoretic network (IN), where the number of layers is determined automatically by the network construction algorithm. A multi-layer information network has the following components:

1) $|I|$ - total number of *hidden* layers (levels) in the network. Each layer is uniquely associated with an input (predicting) attribute by representing the interaction of that attribute and the input attributes of the previous layers. The first layer (layer 0) includes only the root node and is not associated with any input attribute. At each iteration of the network construction procedure, the last hidden layer is termed the *final layer*. The topology of the network differs from the decision-tree structure used by CART [3], ID3 [19], and C4.5 [21] in two aspects: all nodes of a given layer are labeled by the same input attribute and continuous input attributes are discretized to the same intervals at all nodes of the associated layer. In most decision-tree algorithms, the choice of attributes and discretization thresholds is done locally at each node. The structure of the information-

theoretic network is motivated by our belief that many datasets can be accurately represented by a compact model, based on a "global" set of predictive features. This belief is empirically validated in Section 3 of our paper.

2) L_l - a subset of nodes z in a hidden layer No. l. Each node represents a conjunction of values of the first l input attributes, which is similar to the definition of an internal node in a standard decision tree. If a hidden layer l is associated with a nominal input attribute, each outgoing edge of a non-terminal node corresponds to an attribute distinct value. For continuous features, the outgoing edges represent the intervals obtained from the discretization process.

3) K - a subset of distinct target nodes C_t (the target layer). Each target node is associated with a value (class) t in the domain of the target attribute T. For continuous target attributes, the target nodes represent disjoint intervals in the attribute range. A target layer is missing in the standard decision-tree structure.

4) *(z, t)*- connection between a terminal (unsplit) node z and a target node C_t. The information-theoretic meaning of the connection weights is explained in sub-section 2.5 below.

The connectionist nature of our system (each terminal node is connected to every target node) resembles the topological structure of multi-layer neural networks (see [18]), which also have input and output nodes and a variable number of hidden layers. Consequently, we refer to our system as a *network* and not as a *tree*. However, information networks differ from neural networks in that the information-theoretic weights are defined only for the connections to the target layer, whereas internal connections are associated with values or intervals of input attributes and do not have any weights at all. A neural network has, in contrast, a weight associated with every inter-layer connection.

2.2. Illustrative Example

To demonstrate the construction procedure of an information network presented in sub-section 2.3 below, we are using the Credit Approval (“Australian”) dataset from the UCI Repository [2]. This is an encrypted form of a proprietary database, containing data on 690 credit card applications and their outcomes. The data were originally provided by a large bank in

Australia. Each case represents an application for credit card facilities described by eight discrete and six continuous attributes, with two decision classes (Accept / Reject). In the UCI Repository, the original attribute names have been changed to meaningless symbols (A1 – A14) with the purpose of protecting the confidentiality of the data. However, the real names of the attributes are available at the site of Rulequest Research [http://www.rulequest.com/see5-examples.html]. This dataset has been used as a benchmark with a wide range of learning algorithms (starting from [20]).

The database attributes are shown in *Table 1* below. The original names of the attributes (provided by the Rulequest Research site) are given in parentheses. The *Domain* column shows the set or range of possible values for each attribute. In the *Type* column, we make a distinction between discrete (nominal) and continuously valued attributes. The last column (*Use in Network*) specifies how each attribute is treated by the information-theoretic algorithm.

Table 1. **Credit Approval Dataset – List of Attributes**

Attribute	**Domain**	**Type**	**Use in Network**
A1 (Sex)	0, 1	Nominal	Candidate input
A2 (Age)	13.75 - 80.25	Continuous	Candidate input
A3 (Mean time at addresses)	0 - 28	Continuous	Candidate input
A4 (Home status)	1, 2, 3	Nominal	Candidate input
A5 (Current occupation)	1 - 14	Nominal	Candidate input
A6 (Current job status)	1 - 9	Nominal	Candidate input
A7 (Mean time with employers)	0 - 28.5	Continuous	Candidate input
A8 (Other investments)	0, 1	Nominal	Candidate input
A9 (Bank account)	0, 1	Nominal	Candidate input
A10 (Time with bank)	0 - 67	Continuous	Candidate input
A11 (Liability reference)	0, 1	Nominal	Candidate input
A12 (Account reference)	1, 2, 3	Nominal	Candidate input
A13 (Monthly housing expense)	0 - 2000	Continuous	Candidate input
A14 (Savings account balance)	1 - 100001	Continuous	Candidate input
Class (Reject / Accept)	0, 1	Nominal	Target

2.3. Network Construction Procedure

2.3.1 Overview

The network construction algorithm starts with defining the target layer (one node for each target value, or class) and the "root" node representing an empty set of input attributes. Unlike CART [3] and C4.5 [21], IN is built only in one direction (top-down). After the construction process is stopped, there is no bottom-up post-pruning of the network branches. The process of pre-pruning the network is explained below.

A node is split if it provides a statistically significant increase in the *mutual information* of the node and the target attribute. Mutual information, or information gain, is defined as a decrease in the conditional entropy of the target attribute (see [4]). If the tested feature is nominal, the splits correspond to the feature values. Splits on continuous features represent thresholds, which maximize an increase in mutual information. For each new layer, the algorithm re-computes the best threshold splits of continuous attributes and chooses an input attribute (either discrete, or continuous), which provides the maximum increase in mutual information across all nodes of the final layer.

The nodes of a new hidden layer are defined for a Cartesian product of split nodes of the final layer and the values of the new input attribute. According to the chain rule (see sub-section 1.1 above), the *mutual information* between a set of input attributes and the target (defined as the overall decrease in the conditional entropy) is equal to the sum of drops in conditional entropy across all hidden layers. If there is no candidate input attribute significantly decreasing the conditional entropy of the target attribute, the network construction stops.

The network construction algorithm is summarized in Figure 1.

```
Input: the set of $n$ training instances; the set $CI$ of $m$ candidate input attributes (discrete and continuous); the target (classification) attribute $T$; the minimum significance level sign for splitting a network node (default: sign = 0.1%).
Output: a set $I$ of selected input attributes and an information-theoretic network $IN$ (see sub-section 2.1). Each input attribute has a corresponding hidden layer in the network.
Step 1 - Initialize the information-theoretic network: single root node representing all records, no hidden layers ($I = \varnothing$, $l = 0$), and a target layer for the values of the target attribute.
Step 2 - While the number of layers $l < m$ (number of candidate input attributes) do
Step 2.1 – for each candidate input attribute $A_i \notin I$ do
if $A_i$ is discrete then
Return the statistically significant conditional mutual information $cond\_MI_i$ between $A_i$ and $T$.
Else return the best threshold splits of $A_i$ and the statistically significant conditional mutual information $cond\_MI_i$ between $A_i$ and the target attribute $T$.
Step 2.2 – find the candidate input attribute $A_{i^*}$ maximizing $cond\_MI_i$
Step 2.3 –  If $cond\_MI_{i^*} = 0$, then
        End do.
Else
        Step 2.3.1 – expand the network by a new hidden layer associated with the attribute $Ai$ and increment the number of layers $l$.
        Step 2.3.2 – Update the set $I$ of selected input attributes: $I = I \cup A_{i^*}$
Step 3 – Return the set $I$ of selected input attributes and the network structure
```

Figure 1 Network Construction Algorithm

2.3.2 Selecting Nominal Attributes

The Credit Approval dataset (see sub-section 2.2 above) has 14 candidate input attributes (A1 – A14). At the initial stage of the algorithm, all 690 records of the training set belong to the root node. For each nominal attribute, the algorithm calculates the conditional mutual information of a candidate input attribute A_i and the target attribute T given a node z by the following formula (based on [4]):

$$MI\ (A_i; T\ /\ z) = \sum_{t=0}^{M_T-1} \sum_{j=0}^{M_i-1} P(C_t; V_{ij}; z) \bullet \log \frac{P(V_{ij}^{t}\ /\ z)}{P(V_{ij}\ /\ z) \bullet P(C_t\ /\ z)} \quad (3)$$

where

M_T / M_i - number of distinct values of the target attribute T /candidate input attribute i.

$P\ (V_{ij}\ /\ z)$ - an estimated conditional (*a posteriori*) probability of a value j of the candidate input attribute i given the node z (also called a *relative frequency estimator*)

$P\ (C_t\ /\ z)$ - an estimated conditional (*a posteriori*) probability of a value t of the target attribute T given the node z.

$P\ (V_{ij}^{t}/\ z)$ - an estimated conditional (*a posteriori*) probability of a value j of the candidate input attribute i and a value t of the target attribute T given the node z.

$P\ (C_t;\ V_{ij};\ z)$ - an estimated joint probability of a value t of the target attribute T, a value j of the candidate input attribute i, and the node z.

The contingency table[4] for a nominal attribute *A8* (*Other Investments*) and the target attribute (*Class*) at the root node ($z = 0$) is shown in Table 2. The dataset has 306 records of "bad" customers (Class = Reject), who do not have other investments with the bank (Other Investments = No). Only 23 customers with the same characteristics turned out to be "good" customers (Class = Accept). The customers with other investments (Other Investments = Yes) represent an opposite pattern: most of them (284 out of 361) should be accepted by the bank. The resulting conditional mutual information for this attribute, based on the estimated values of conditional and unconditional probabilities, is *0.426 bits*.

[4] A *contingency table* represents a joint frequency distribution, or *cross-tabulation*, of two discrete random variables [17]

Table 2 Contingency Table: Other Investments (Node 0)

j/t	0 (Reject)	$P(V_{ij}^{t}/z)$	$P(C_t; V_{ij}; z)$	1 (Accept)	$P(V_{ij}^{t}/z)$	$P(C_t; V_{ij}; z)$	Total	$P(V_{ij}/z)$
0 (No)	306	0.443	0.443	23	0.033	0.033	329	0.477
1 (Yes)	77	0.112	0.112	284	0.412	0.412	361	0.523
Total / $P(C_t/z)$	383	0.555		307	0.445		690	

The statistical significance of the estimated conditional mutual information between a candidate input attribute A_i and the target attribute T, is evaluated by using the likelihood-ratio statistic (based on [1]):

$$G2\ (Ai\ ;\ T / z) = 2\bullet(\ln 2)\bullet E^{*}(z)\ \bullet MI\ (Ai\ ;\ T / z) \qquad (4)$$

Where $E^{*}(z)$ is the number of records associated with the node z

The Likelihood-Ratio Test [23] is a general-purpose method for testing the null hypothesis H_0 that two discrete random variables are statistically independent. For example, if the customer credibility is independent of his/her other investments in the bank, the proportion of credible customers among those having other investments should be equal to their proportion among those who do not. The Likelihood-Ratio Test is directly related to the information theory, since independence of two attributes implies that their expected mutual information is zero. If H_0 holds, then the likelihood-ratio test statistic $G^2\ (A_i\ ;\ T / z)$ is distributed as chi-square with $(NI_i\ (z)\ -\ 1)\bullet(\ NT\ (z)\ -\ 1)$ degrees of freedom, where $NI_{\ i}\ (z)$ is the number of distinct values of a candidate input attribute i at node z and $NT\ (z)$ is the number of values (classes) of the target attribute T at node z.

The default significance level (*p-value*), used by the information-theoretic algorithm, is 0.1%. We have found empirically that larger *p-values* tend to decrease the generalization performance of the network for most real-world datasets. Thus, under normal circumstances, the user should keep the default level of *p-value* rather than running the algorithm repeatedly for a set of *p-values* and comparing the results. However, if a given dataset is known to contain mostly regular patterns rather than random noise, the user can weaken the significance requirement of the test, up to removing the significance testing completely.

In the Credit dataset, there are 690 records associated with the root node. This implies that the likelihood-ratio statistic of *Other Investments* is *2*ln2*690*0.426 = 407*. The attribute

Other Investments has two values (Yes / No) and the target attribute (*Class*) has two values as well. Consequently, the likelihood-ratio statistic calculated above has (2-1)*(2-1) = 1 degree of freedom. According to the chi-square distribution, the significance level of the obtained value is much higher than 0.1%, which means that the null hypotheses can be rejected and *Other Investments* is considered as a candidate for the next input attribute.

Calculated values of conditional mutual information for other candidate input attributes are shown in Table 3. Discretization of continuous attributes is demonstrated in the next sub-section. As one can see, *Other Investments* happens to be the best attribute at Layer 0 and it is selected as the first input attribute in the network. The next layer (Layer 1) is going to have two hidden nodes corresponding to 329 records of those customers who do not have other investments (Node 1) and 361 records of those who have them (Node 2).

Table 3 Conditional Mutual Information (Layer 0)

Attribute	Significant Conditional Mutual Information
A1 (Sex)	0
A2 (Age)	0.023
A3 (Mean time at addresses)	.041
A4 (Home status)	.030
A5 (Current occupation)	.109
A6 (Current job status)	.050
A7 (Mean time with employers)	.123
A8 (Other investments)	.426
A9 (Bank account)	.156
A10 (Time with bank)	.214
A11 (Liability reference)	0
A12 (Account reference)	0
A13 (Monthly housing expense)	.051
A14 (Savings account balance)	.123

2.3.3 Selecting Continuous Attributes

The conditional entropy of the target attribute can only be calculated with respect to attributes taking a finite number of values. The algorithm performs discretization of continuous attributes "on-the-fly" by using an approach, which is similar to the information-theoretic heuristic of Fayyad and Irani [7]: recursively finding a binary partition of an input attribute that minimizes the conditional entropy of the target attribute. However, the stopping criterion we are using is different from [7]. Rather than searching for a *minimum description length* (minimum number of bits for encoding the training data), we make use of a standard statistical *likelihood-ratio test* [23]. As indicated in sub-section 2.3.2 above, the significance level of the test can be adjusted to the noisiness of a given dataset. The MDL-based stopping criterion of [7] does not provide this flexibility. The search for the best partition of a continuous attribute is *dynamic*: it is performed each time a candidate input attribute is considered for selection. The dynamic discretization algorithm is described in the Appendix.

One of continuous attributes considered at Layer 1 is *A14* (*Savings Account Balance*). This attribute has 238 distinct values ranging from 1 to 100,001. The dynamic discretization algorithm tries to split the range of this attribute by each one of its values across all nodes of the final hidden layer (Nodes 1 and 2 in our case). This requires calculating conditional mutual information from 2 × 238 = 476 contingency tables. The algorithm finds the threshold providing the maximum significant conditional mutual information and applies the same procedure recursively to each one of resulting sub-intervals (again across all final nodes) as long as the increase in the conditional mutual information is statistically significant. For the *Balance* attribute, the first best threshold found by the algorithm is 401.

The contingency tables for the threshold of *Balance* = 401 at Nodes 1 and 2 are shown in Table 4 and Table 5 respectively. The total number of cases in each table is equal to the number of records associated with the corresponding nodes (329 and 361). The chi-square statistic G^2 for Node 1 is 0.052 (using expression (13) in the Appendix). The confidence level of this value (with one degree of freedom) is very low: about 18% only. Thus, we cannot reject the null hypothesis and split the *Balance* attribute at this node. However, at Node 2, we get $G^2 = 56.7$, which has a significance level much higher than 0.1%. Consequently, the conditional mutual information of *0.059 bits* that we have at this node can be considered statistically significant. Since the final layer (Layer 1) has only two nodes and subsequent partitioning of the *Balance* range has not caused a statistically significant improvement in the mutual information, the total conditional mutual information between *Balance* and the target attribute given the final layer is also equal to 0.059 *bits*. This number is higher than the conditional mutual information associated with any other attribute, which makes *Balance* the second selected attribute in the network. The new layer (Layer 2) has two nodes (No. 3 and 4) associated with the two intervals of *Balance* (below 401 and above 401). In the network, these nodes are connected by edges to Node 2 in Layer 1. Node 1 becomes a terminal node. The third and the last input attribute (*Bank Account*) has been selected by using a similar procedure.

Table 4 Contingency Table: Balance (Node 1)

y / t	**0**	***P*** **(*****S*** $_y$ **;** ***C*** $_t$ **/** ***S, z)***	***P*** **(*****S*** $_y$**;** ***C*** $_t$**;** ***z)***	**1**	***P*** **(*****S*** $_y$ **;** ***C*** $_t$ **/** ***S, z)***	***P*** **(*****S*** $_y$**;** ***C*** $_t$**;** ***z)***	**Total**	***P*** **(*****S*** $_y$**/** ***S, z)***
Balance **<= 401**	271	0.824	0.393	20	0.061	0.029	291	0.884
Balance **> 401**	35	0.106	0.051	3	0.009	0.004	38	0.116
Total / ***P (C*** $_t$***/ S, z)***	306	0.930		23	0.070		329	

Table 5 Contingency Table: Balance (Node 2)

y / t	**0**	***P*** **(*****S*** $_y$ **;** ***C*** $_t$ **/** ***S, z)***	***P*** **(*****S*** $_y$**;** ***C*** $_t$**;** ***z)***	**1**	***P*** **(*****S*** $_y$ **;** ***C*** $_t$ **/** ***S, z)***	***P*** **(*****S*** $_y$**;** ***C*** $_t$**;** ***z)***	**Total**	***P*** **(*****S*** $_y$**/** ***S, z)***
Balance **<= 401**	74	0.205	0.107	156	0.432	0.226	230	0.637
Balance **> 401**	3	0.008	0.004	128	0.355	0.186	131	0.363
Total / ***P (C*** $_t$***/ S, z)***	77	0.213		284	0.787		361	

2.3.4 Summary

The iterations of the network construction procedure applied to the Credit Approval data set are summarized in Table 6 below. The table shows the input attribute selected at each step and the associated change in the conditional entropy of the target attribute. The table also includes the increase in the network size (number of split nodes) and the ratio between the cumulative mutual information and the number of input attributes. Only three attributes (out of 14 candidates) were selected by the information-theoretic algorithm. As one can see from the first row of Table 6, the first input attribute (*Other Investments*) contributes more than 80% of the overall mutual information, which is equal to 0.516 bits (see the last row).

Table 6 Network Construction Procedure– Credit Approval

Iteration	Attribute Name	Mutual Information	Conditional MI	Conditional Entropy	Split Nodes	MI to Attributes
0	Other investments (A8)	0.426	0.426	0.566	1	0.426
1	Balance (A14)	0.485	0.059	0.506	1	0.243
2	Bank account (A9)	0.516	0.031	0.475	1	0.172

The resulting information-theoretic connectionist network is shown in Figure 2 below. Thick lines represent internal connections (standing for values or intervals of input attributes) and thin lines denote the connections between the terminal nodes and the nodes of the target layer. Dotted thin lines indicate the predicted target values, i.e., the values having maximum probability at a given terminal node. The classification performance of the networks induced from this and other datasets is evaluated in Section 3 below.

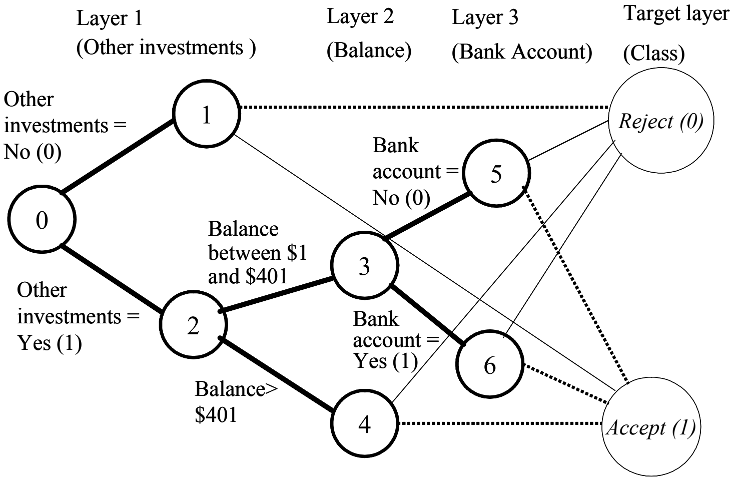

Figure 2 Information-Theoretic Network: Credit Dataset

The subset of attributes selected by the information-theoretic algorithm can be used as an input for any other data mining method, i.e., IN can be implemented as a *feature filter* method in the KDD process. The application of the information-theoretic methodology to feature selection is discussed by us in [13].

2.4. Time and Space Complexity

The computational complexity of the network construction procedure depends on the types of candidate input attributes presenting in the training data. In this sub-section, we compute the complexity bounds for "pure" datasets, which include either discrete or continuous attributes only. In the case of a "mixed" dataset, the overall complexity can be roughly estimated by the following expression:

$$Comp = \frac{D}{m} Comp_d + \frac{C}{m} Comp_c \qquad (5)$$

where m is the total number of candidate input attributes and D (C) is the number of discrete (continuous) attributes respectively. Accordingly, $Comp_d$ $(Comp_c)$ is the computational complexity in a purely discrete (continuous) dataset.

The computational complexity bounds are calculated by using the following notation:

n - total number of records in a training data set

m - total number of candidate input attributes

p - portion of significant input attributes, selected by the network construction procedure ($p \leq 1$)

L – maximum number of hidden nodes in a layer (bounded by the number of distinct conjunctions of input attribute values presenting in the training set)

M_C - maximum domain size of a candidate input attribute. For continuous attributes, this is the maximum number of thresholds considered by the discretization procedure. In our algorithm, like in the discretization algorithm of Fayyad and Irani [7], the number of potential thresholds is equal to the number of distinct attribute values, which is bounded by the size of the training set (n).

M_T - domain size of the target attribute (number of distinct classes)

The computational "bottleneck" of the algorithm is estimating the conditional mutual information of a candidate input attribute and the target attribute, given every hidden node. The calculation of the conditional mutual information is performed at each hidden layer of the information-theoretic network for all candidate input attributes at that layer. All frequency estimators used in the calculation of $MI\ (A_i\ ;\ T\ /\ z)$ are re-computed at each hidden node by a single pass over a subset of training examples associated with that node (bounded by *n*). If a candidate input attribute is discrete, the summation terms of *MI* $(A_i\ ; T / z)$ refer to a Cartesian product of values of a candidate input attribute and the target attribute. The number of hidden layers is equal to *pm*. This implies that for discrete attributes, the total number of calculations is bounded by:

$$Comp_d = \sum_{s=0}^{pm} L \bullet (n + M_T \bullet M_C) \bullet (m - s) \leq \frac{L \bullet (n + M_T \bullet M_C) \bullet m^2 \bullet p \bullet (2 - p)}{2} \qquad (6)$$

The number of possible partitions of a continuous attribute is bounded by M_C. For every possible partition, the summation terms *MI (Th; T / S, z)* are summed over all nodes of the final layer for a Cartesian product of two sub-intervals and the values of the target attribute. Consequently, for continuous attributes, the total number of calculations is bounded by:

$$Comp_c = \sum_{s=0}^{pm} L \bullet M_C \bullet (n + 2M_T) \bullet (m - s) \leq \frac{L \bullet M_C \bullet (n + 2M_T) \bullet m^2 \bullet p \bullet (2 - p)}{2} \qquad (7)$$

Roughly speaking, the additional computational effort associated with dynamic discretization of *m* continuous attributes, is proportional to the product $M_C\, n\, m^2$. Thus, the time complexity of the network construction procedure is linear in the number of records, linear in the number of distinct attribute values, and quadratic in the number of candidate input attributes. Moreover, it is reduced by the factor of *p (2-p)*, which is based upon the portion *p* of significant input attributes in a network.

Like most other algorithms for decision-tree construction (e.g., see [21]), our method requires all the training examples to reside in main memory (RAM). For each network node, the program has also to keep in memory the predicting discrete value (or a continuous threshold) associated with every input attribute. The resulting space complexity of the network construction procedure is *m (n + pmL)*. Scaling up the algorithm for very large datasets is a subject of ongoing research.

2.5. Classification and Rule Extraction

Due to the disjunctive nature of the information-theoretic multi-layer network, each record in a database can be associated with one and only one terminal node *z*. The predicted value t^* of a target attribute *T* at a terminal node *z* is found by the *maximum a posteriori* rule:

$$t^* = \arg\max_{t}\{P(C_t / z)\}\,. \qquad (8)$$

Terminal nodes represent conjunctions of values of input attributes. A connection between a terminal node *z* and a node of the target layer, associated with the value (class) C_t, can be interpreted as a probabilistic rule of the form

If terminal node = z then the value of the target attribute T is t with probability of P (C_t/z)

In our previous work [16], we have extracted a probabilistic rule from every connection between a terminal node and a node of the target layer. This has resulted in a large number of rules having positive and negative information-theoretic weights. In this paper, we are reducing the number of rules extracted from an information network by associating a single classification rule (*If z then t**) with every terminal node. The weight of that rule is calculated as the sum of information-theoretic weights over all edges connecting the corresponding terminal node to the nodes of the target layer:

$$w_z^t = \sum_{t=0}^{M_T-1} P(C_t;z)\cdot\log\frac{P(C_t/z)}{P(C_t)} = P(z)\cdot\sum_{j=0}^{M_T-1} P(C_t/z)\cdot\log\frac{P(C_t/z)}{P(C_t)} \qquad (9)$$

Where

$P(C_t ; z)$ - an estimated joint probability of the target value C_t and the node *z*.

$P(C_t / z)$ - an estimated conditional (*a posteriori*) probability of the target value C_t given the node *z*.

$P(C_t)$ - an estimated unconditional (*a priori*) probability of the target value C_t.

P(z) –probability (relative frequency) of a node *z*

The above expression is an extension of the average *information content* of a rule, defined by Smyth and Goodman [25] for binary-valued attributes, to a more general case of multi-valued input and target attributes. As indicated in [25], a measure of this form (called *J-measure*) is useful for relative evaluation of rules induced from data, since its value is always non-negative and it represents both the simplicity (probability of node occurrence *P(z)*) and goodness-of-fit (*cross entropy*) of a given rule.

Proposition 1. The sum of connection weights across all terminal nodes is equal to the estimated mutual information between the set of input attributes and the target attribute:

$$MI\,(T;\,I) = \sum_{z \in F} \sum_{t=0}^{M_T - 1} P(C_t; z) \bullet \log \frac{P(C_t / z)}{P(C_t)} \tag{10}$$

where

T – the target attribute

I - set of input attributes

z – hidden node in the information-theoretic network

F - subset of unsplit (terminal) nodes.

P(C_t ;z) - an estimated joint probability of the value C_t and the node *z*.

$P\ (C_t / z)$ - an estimated conditional (*a posteriori*) probability of the value C_t given the node *z*.

$P(C_t)$ - an estimated unconditional (*a priori*) probability of the value C_t.

Proof. This proposition is directly derived from the definition of mutual information between random variables *X* and *Y* [4]:

$$MI(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \bullet \log \frac{p(y / x)}{p(y)} \tag{11}$$

In the above expression, we have substituted *Y* with the target attribute *T* and *X* with the set of input attributes *I.* A node $z \in F$ represents a conjunction of input attribute values. Since the information-theoretic network represents a disjunction of these conjunctions, each conjunction is associated with one and only one node $z \in F$. Consequently, the summation over the terminal nodes covers all possible values of the input attributes. This completes the proof.

The most informative rules can be found by sorting the information-theoretic connection weights (w_z^i) in decreasing order. The four rules extracted from the network of the Credit Approval Dataset (see Figure 2 above) are presented in Table 7 below. It seems like the rules 2 – 4 can be replaced with a single classification rule *If Other investments is 1 then Class is 1.* This rule reduction will not affect the classification accuracy of the network, but will ignore an important fact that two input attributes *Balance* and *Bank Account* do affect the credibility of a potential customer.

Table 7 Information-Theoretic Rules: Credit Dataset

Rule No.	Terminal Node No.	Rule	Weight
1	1	If Other investments is 0 then Class is 0	0.2413
2	4	If Other investments is 1 and Balance is more than 445 then Class is 1	0.1906
3	6	If Other investments is 1 and Balance is between 1 and 445 and Bank account is 1 then Class is 1	0.0821
4	5	If Other investments is 1 and Balance is between 1 and 445 and Bank account is 0 then Class is 1	0.0019

3. Empirical Evaluation

3.1. Overview

The performance of the information-theoretic algorithm was evaluated on ten publicly available data sets: Breast Cancer, Chess Endgames, Credit Approval, Diabetes, Glass Identification, Heart Disease, Iris Plants, Liver, Lung Cancer, and Wine. All these data sets are posted at the UCI Machine Learning Repository [2] and widely used by the data mining community for evaluating learning algorithms. The data sets selected by us here comprise a diverse mixture of attribute types, ranging from purely continuous to purely nominal attribute domains. A summary of characteristics of these datasets appears in Table 8 below. The algorithm performance (in terms of dimensionality reduction and predictive accuracy) is compared to two decision-tree algorithms: ID3 (presented by Quinlan in [19]) and C4.5, which is a state-of-the-art decision tree algorithm introduced in [21] and improved in [22]. The classification accuracy of C4.5 is based on the "fine tuned" results published in literature, which represent the optimized performance of this algorithm. Other results were obtained with the default settings of all algorithms including the information-theoretic network.

Table 8 Description of Datasets

Dataset	Number of Records	Classes	Candidate Attributes Continuous	Nominal	Total
Breast	699	2	9	0	9
Chess	3196	2	0	36	36
Credit	690	2	6	8	14
Diabetes	768	2	8	0	8
Glass	214	6	9	0	9
Heart	270	2	6	7	13
Iris	150	3	4	0	4
Liver	345	2	6	0	6
Lung-cancer	32	3	0	57	57
Wine	178	3	13	0	13

3.2. Dimensionality Reduction

As indicated in sub-section 1.2 above, dimensionality reduction is an important objective of the knowledge discovery process. Most real-world datasets contain some portion of completely irrelevant attributes. Unlike the Naïve Bayes Classifier, which uses *all* attributes in a dataset, decision-tree algorithms tend to remove irrelevant attributes from the final tree (see [21]). The network construction algorithm, presented above, is also aimed at minimizing the set of input attributes in an information-theoretic network. Table 9 below shows the initial number of candidate input attributes in each dataset, the number of input attributes selected by the evaluated algorithms (ID3, C4.5, and IN), and the reduction in data dimensionality (the portion of candidate input attributes that were excluded from the model). The C4.5 trees were built by using Version 8 of the algorithm (described in [22]). For ID3 and C4.5, we have counted all the attributes that appear in at least one tree path. In the information-theoretic network (IN), the number of input attributes is equal to the number of internal layers. The training sets included all records of each dataset. Table 9 also compares the complexity of the resulting models in terms of the total number of nodes in a tree / network and the run times of the algorithms on a Pentium III computer.

Table 9 Dimensionality Reduction – Summary Table

	Candidate Input Attributes	Selected Input Attributes			Dim. Reduction (%)			Total nodes			Run time (sec.)		
Dataset		**ID3**	**C4.5**	**IN**	**ID3**	**C4.5**	**IN**	**ID3**	**C4.5**	**IN**	**ID3**	**C4.5**	**IN**
Breast	9	4	7	4	56%	22%	56%	15	23	15	0.06	0.05	0.11
Chess	36	19	22	9	47%	39%	75%	51	59	29	0.55	0.33	0.60
Credit	14	4	9	3	71%	36%	79%	9	43	7	0.33	0.11	1.04
Diabetes	8	4	6	4	50%	25%	50%	21	43	15	0.61	0.11	0.82
Glass	9	7	9	4	22%	0%	56%	27	45	13	0.27	0.11	0.44
Heart	13	5	10	4	62%	23%	69%	22	49	18	0.11	0.05	0.11
Iris	4	2	2	1	50%	50%	75%	7	9	5	0.05	0.00	0.05
Liver	6	4	6	3	33%	0%	50%	9	51	7	0.06	0.06	0.11
Lung-cancer	57	1	5	1	98%	91%	98%	4	16	4	0.05	0.00	0.00
Wine	13	3	3	3	77%	77%	77%	9	9	11	0.16	0.06	0.27
Mean	**16.9**	**5.3**	**7.9**	**3.6**	**57%**	**36%**	**68%**	**17.4**	**34.7**	**12.4**	**0.23**	**0.09**	**0.36**

The results show that the models produced by the information-theoretic algorithm are significantly smaller than the decision trees built by ID3 and C4.5. Thus, C4.5 failed to remove more than 50% of the attributes in eight datasets out of ten. On the other hand, the information-theoretic network never included more than 50% of available attributes. The average difference between the two methods is 32% of the number of available attributes. This means that the information-theoretic algorithm is a much more "aggressive" dimensionality reducer than C4.5. ID3 tends to use less attributes than C4.5, but still its average number of selected attributes (5.3) is higher than the IN average (3.6).

Table 9 also shows that, almost in all cases, the information-theoretic network produces a simpler model, compared to ID3 and C4.5. IN has the minimal number of nodes in nine data sets out of ten. This result is not completely surprising, since our algorithm is using less input attributes, and each node represents a conjunction of values of input attributes. Due to the repetitive partitioning of the training set (as opposed to the recursive approach of ID3 and C4.5), IN is slightly slower than other decision-tree methods. As mentioned above, scaling-up the IN algorithm and improving its computational efficiency is a subject of ongoing research.

In the next sub-section, we are examining the trade-off between the dimensionality reduction and the predictive accuracy of the information-theoretic network.

3.3. Predictive Accuracy

We are using here a common approach to estimating predictive accuracy, called *k-fold cross-validation* (see [18]). According to this approach, the data set is randomly partitioned into *k* disjoint subsets, with each subset being used once in a test set and *k-1* times in a training set. Following the common practice of other researchers (e.g., see [15]), we have chosen the value of *k* to be 10. Due to the high variance of cross-validation runs, we have performed 10 runs of 10-fold cross-validation, each based on a different random partitioning of the data set.

Table 10 shows, for each dataset, the estimated predictive accuracy of the information-theoretic network vs. other decision-tree methods. The results of ID3 were obtained with our own implementation of the algorithm (based on [19]), while C4.5 results were taken from [5]. The confidence intervals for the IN predictive accuracy have been calculated at the 0.95 confidence level, using *t*-distribution with *n-1* degrees of freedom, where *n* is the number of 10-fold cross-validation runs (10). An asterisk (*) next to the upper bound of a confidence interval denotes a statistically significant advantage of C4.5 over IN.

As one can see from Table 10, the predictive accuracy of the information-theoretic algorithm tends to be only slightly worse than the accuracy of C4.5. One exception is the Iris dataset, where the network has provided us with better results than C4.5 *along with* reducing dimensionality by 75%. In other datasets, a small loss of accuracy (the mean difference of less than 1%) is compensated by a considerable reduction in the number of input attributes (on average, the algorithm uses about 1/3 of the candidate input attributes, which appear in each data set). ID3 does not show any advantages at all, since it has the lowest average accuracy, while using more input attributes than IN. Though the choice of the best model (either the most accurate, or the simplest) depends on a specific application, we believe that in many cases, a small amount of accuracy can be sacrificed for the sake of obtaining a much more compact and interpretable model, like the one produced by the information-theoretic algorithm.

Table 10 Predictive Accuracy – Comparison to Other Methods

Dataset	ID3	C4.5	IN	IN - Min.	IN - Max.	
Breast	93.6%	94.4%	94.3%	94.1%	94.6%	
Chess	99.1%	99.2%	97.7%	97.7%	97.7%	*
Credit	83.1%	85.9%	84.1%	83.0%	85.1%	*
Diabetes	73.3%	73.5%	72.2%	71.2%	73.2%	*
Glass	63.8%	67.9%	60.9%	59.9%	61.9%	*
Heart	74.3%	77.5%	75.8%	74.7%	76.9%	*
Iris	94.9%	92.6%	95.6%	95.3%	95.9%	
Liver	63.5%	65.9%	63.7%	62.7%	64.8%	*
Lung-cancer	33.4%	40.9%	46.6%	40.1%	53.0%	
Wine	91.3%	92.4%	90.4%	89.5%	91.4%	*
Mean	**77.0%**	**79.0%**	**78.1%**	**76.8%**	**79.5%**	

4. Conclusion

In this paper, we have presented a novel algorithm for building simple and reasonably accurate classification models, termed *information-theoretic networks*. The underlying principles of our methodology include maximization of mutual information, dimensionality reduction, and statistical significance testing. The algorithm was evaluated on a wide range of standard datasets containing continuous, categorical, and binary-valued attributes. The related issues to be further studied include: integrating the information-theoretic network with other data mining methods (e.g., by using the algorithm as a feature selector only), utilizing prior knowledge in the network construction procedure, and applying the algorithm to non-relational (e.g., spatial) data.

Appendix: Dynamic Discretization Algorithm

Partition (Data Table r, Information Network, Attribute A_i, Interval S, Significance Level sign)

Input: the set of *n* training instances, an information-theoretic network, a continuous attribute A_i to be discretized, the interval *S* to be partitioned (the first and the last distinct values of A_i), and the minimum significance level *sign* for splitting an interval (default: *sign* = 0.1%).

Output: the total number of discretization intervals for A_i and the lower bound of each interval.

Step 1 – Initialize to zero the degrees of freedom and the estimated conditional mutual information of the candidate input attribute and the target attribute given the final hidden layer of nodes.

Step 2 – Repeat for every distinct value included in the interval S (except for the last value):

Step 2.1 – Define the value as a partitioning threshold (*Th*). All values below or equal to *Th* belong to the first sub-interval S_1. Distinct values above *Th* belong to the second sub-interval S_2.

Step 2.2 – Repeat for every node z of the final hidden layer:

Step 2.2.1 – Calculate the estimated conditional mutual information between the partition of the interval *S* at the threshold *Th* and the target attribute *T* given the node z by the following formula (based on [4]):

$$MI(Th; T / S, z) = \qquad (12)$$

$$\sum_{t=0}^{M_T-1} \sum_{y=1}^{2} P(S_y; C_t; z) \bullet \log \frac{P(S_y; C_t / S, z)}{P(S_y / S, z) \bullet P(C_t / S, z)}$$

Where

P (S $_y$/ S, z) - an estimated conditional (a posteriori) probability of a sub-interval S $_y$, given the interval S and the node z.

$P(C_t / S, z)$ - an estimated conditional (a posteriori) probability of a value C_t of the target attribute *T* given the interval *S* and the node *z*.

$P(S_y; C_t / S, z)$ - an estimated joint probability of a value C_t of the target attribute *T* and a sub-interval S_y given the interval *S* and the node *z*.

$P(S_y; C_t; z)$ - an estimated joint probability of a value C_t of the target attribute T, a sub-interval S_y, and the node *z*.

Step 2.2.2 - Calculate the likelihood-ratio test for the partition of the interval *S* at the threshold *Th* and the target attribute *T* given the node z by the following formula (based on [23]):

$$G^2(Th; T / S, z) = 2\sum_{j=0}^{M_i-1}\sum_{y=1}^{2} N_{ij}(S_y, z) \bullet \ln \frac{N_t(S_y, z)}{P(C_t / S, z) \bullet E(S_y, z)} \qquad (13)$$

Where

$N_t(S_y, z)$ - number of occurrences of a value C_t of the target attribute *T* in sub-interval S_y and the node z.

$E(S_y, z)$ - number of records in sub-interval S_y and the node z.

$P(C_t / S, z)$ - an estimated conditional (a posteriori) probability of a value C_t of the target attribute *T* given the interval *S* and the node *z*.

$P(C_t / S, z) \bullet E(S_y, z)$ - an estimated number of occurrences of a value C_t of the target attribute *T* in sub-interval S_y and the node *z* under the assumption that the conditional probabilities of the target attribute values are identically distributed given each sub-interval.

Step 2.2.3- Calculate the degrees of freedom of the likelihood-ratio statistic by:

$$DF(Th; T / S, z) = (NI\ i\ (S, z) - 1)\bullet(NT\ (S, z) - 1) = (2-1)\bullet(NT\ i\ (S, z) - 1) = NT\ i\ (S, z) - 1 \qquad (14)$$

Where

$NI_i(S, z)$ - number of sub-intervals of a candidate input attribute *i* at node *z* (2)

$NT(S, z)$ - number of values of the target attribute in the interval *S* at node *z*.

Step 2.2.4 - If the likelihood-ratio statistic is significant, mark the node as “split” by the threshold *Th* and update the estimated conditional mutual information of the candidate input attribute and the target attribute given the threshold *Th*; else mark the node as “unsplit” by the threshold *Th*.

Step 2.2.5 - Go to next node.

Step 2.3 – Go to next distinct value.

Step 3 – Find the threshold Th_{max} maximizing the estimated conditional mutual information between a partition of the candidate input attribute A_i and the target attribute T given the interval S and the set of input attributes I by:

$$Th_{\max} = \arg\max_{Th} MI(Th;T/I\ ,S) \qquad (15)$$

and update the estimated conditional mutual information $cond\_MI_i$ between the candidate input attribute A_i and the target attribute T.

Step 4 – If the maximum estimated conditional mutual information is greater than zero, then do:

Step 4.1 - Repeat for every node z of the final hidden layer: If the node z is split by the threshold Th_{max}, mark the node as split by the candidate input attribute A_i

Step 4.2 - If the threshold Th_{max} is the first distinct value in the interval S, mark Th_{max} as the lower bound of a new discretization interval, else ***Partition*** *(Data Table r, Network, Attribute A_i, Interval S_1).*

Step 4.3 - ***Partition*** *(Data Table r, Network, Attribute A_i, Interval S_2)*

Step 4.4 - EndDo.

Else:

Step 5 - Define a new discretization interval S and increment the domain size of A_i (number of discretization intervals).

References

[1] F. Attneave, *Applications of Information Theory to Psychology*, Holt, Rinehart and Winston, 1959.

[2] C.L. Blake and C.J. Merz, "UCI Repository of Machine Learning Databases," http://www.ics.uci.edu/~mlearn/MLRepository.html [19 July 2002].

[3] L. Breiman, J.H. Friedman, R.A. Olshen, and P.J. Stone, *Classification and Regression Trees*, Wadsworth, 1984.

[4] T. M. Cover and J.A. Thomas, *Elements of Information Theory*, Wiley, 1991.

[5] P. Domingos and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss," *Machine Learning*, no. 29, pp. 103-130, 1997.

[6] P. Domingos, "Occam's Two Razors: The Sharp and the Blunt," *Proc. Fourth Int'l Conf. on Knowledge Discovery and Data Mining*, pp. 37-43, Menlo Park , CA, 1998.

[7] U. Fayyad and K. Irani, "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning," *Proc. Thirteenth Int'l Joint Conference on Artificial Intelligence*, pp. 1022-1027, San Mateo, CA, 1993.

[8] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From Data Mining to Knowledge Discovery: An Overview," *Advances in Knowledge Discovery and Data Mining*, U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy eds., pp. 1-36. AAAI/MIT Press, 1996.

[9] A.L. Gorin, S.E. Levinson, A.N. Gertner and E. Goldman, "Adaptive Acquisition of Language," *Computer Speech and Language*, vol. 5, no. 2, pp. 101-132, 1991.

[10] A.L. Gorin, S.E. Levinson and A. Sankar, "An Experiment in Spoken Language Acquisition," *IEEE Transactions on Speech and Audio Processing,* vol. 2, no. 1, pp.224-239, 1994.

[11] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem," *Proc. 11th Int'l Conf. on Machine Learning*, pp. 121-129, 1994.

[12] M. Last, Y. Klein, A. Kandel, "Knowledge Discovery in Time Series Databases," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 31: Part B, no. 1, pp. 160-169, Feb. 2001.

[13] M. Last, A. Kandel, O. Maimon, "Information-Theoretic Algorithm for Feature Selection," *Pattern Recognition Letters*, vol. 22, no. 6-7, pp. 799-811, 2001.

[14] M. Last and A. Kandel, "Data Mining for Process and Quality Control in the Semiconductor Industry," *Data Mining for Design and Manufacturing: Methods and Applications*, D. Braha, ed., pp. 207-234, Boston: Kluwer Academic Publishers, 2001.

[15] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, Boston: Kluwer Academic Publishers, 1998.

[16] O. Maimon and M. Last, *Knowledge Discovery and Data Mining, The Info-Fuzzy Network (IFN) Methodology*, Boston: Kluwer Academic Publishers, 2001.

[17] E.W. Minium, R.B. Clarke, T. Coladarci, *Elements of Statistical Reasoning*, New York: Wiley, 1999.

[18] T.M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

[19] J.R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, no. 1, pp. 81-106, 1986.

[20] J.R. Quinlan, "Simplifying Decision Trees," *International Journal of Man-Machine Studies*, no. 27, pp. 221-234, 1987.

[21] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

[22] J.R. Quinlan, “Improved Use of Continuous Attributes in C4.5,” *Journal of Artificial Intelligence Research,* no. 4, pp. 77-90, 1996.

[23] C.R. Rao and H. Toutenburg, *Linear Models: Least Squares and Alternatives*, Springer-Verlag, 1995.

[24] R. Rastogi, K. Shim, “PUBLIC: A Decision Tree Classifier that Integrates Building and Pruning,” *Proc. 24th International Conference on Very Large Databases (VLDB’98)*, pp. 404-415, 1998.

[25] P. Smyth, R. M. Goodman, “An Information Theoretic Approach to Rule Induction from Databases,” *IEEE Transactions on Knowledge and Data Engineering* vol. 4, no. 4, pp. 301-316, 1992.