Artificial Intelligence in Medicine 16 (1999) 3–23

Artificial Intelligence in Medicine

# Selected techniques for data mining in medicine

Nada Lavrač *

*Department of Intelligent Systems, J. Stefan Institute, 1000 Ljubljana, Slovenia*

Received 1 November 1997; received in revised form 20 May 1998; accepted 7 July 1998

**Abstract**

Widespread use of medical information systems and explosive growth of medical databases require traditional manual data analysis to be coupled with methods for efficient computer-assisted analysis. This paper presents selected data mining techniques that can be applied in medicine, and in particular some machine learning techniques including the mechanisms that make them better suited for the analysis of medical databases (derivation of symbolic rules, use of background knowledge, sensitivity and specificity of induced descriptions). The importance of the interpretability of results of data analysis is discussed and illustrated on selected medical applications. © 1999 Elsevier Science B.V. All rights reserved.

*Keywords:* Data mining; Machine learning; Medical applications

## 1. Introduction

Modern hospitals are well equipped with monitoring and other data collection devices which provide relatively inexpensive means to collect and store the data in inter- and intra-hospital information systems. Extensive amounts of data gathered in medical databases require specialized tools for storing and accessing data, for data analysis, and for effective use of data. In particular, the increase in data volume causes great difficulties in extracting useful information for decision sup-

* Tel.: + 386-61-1773272; fax: + 386-61-1251038.
*E-mail address:* nada.lavrac@ijs.si (N. Lavrač)

PII: S0933-3657(98)00062-1

port. Traditional manual data analysis has become inadequate, and methods for efficient computer-based analysis indispensable. To satisfy this need, medical informatics may use the technologies developed in the new interdisciplinary field of knowledge discovery in databases (KDD) [16], encompassing statistical, pattern recognition, machine learning, and visualization tools to support the analysis of data and the discovery of regularities that are encoded within the data.

KDD typically consisting of the following steps [14,13]: understanding the domain, forming the dataset and cleaning the data, extracting regularities hidden in the data and formulating knowledge in the form of patterns or rules (this step in the overall KDD process is usually referred to as *data mining* (DM)), post-processing of discovered knowledge, and exploiting the results. This paper is only concerned with data mining, and presents some approaches to it which can be used to extract medical knowledge for diagnosis, screening, prognosis, monitoring, therapy support or overall patient management.

The results of computer-based analysis have to be communicated to people in an understandable way. In this respect, the analysis tools have to deliver transparent results and most often facilitate human intervention in the analysis process. A good example is a machine learning tool that, as a result of data analysis, aims to derive a symbolic model of high transparency and accuracy.

This paper presents selected data mining techniques that can be applied in medicine, and in particular those techniques that are well suited for the analysis of medical databases (derivation of symbolic rules, use of background knowledge, sensitivity and specificity of induced descriptions). Due to space limitations we omit the description of mechanisms for handling noise and missing data, as well as mechanisms for dealing with real valued and temporal data, which are also crucial for successful data mining in medicine. The importance of the interpretability of results of data analysis is discussed and illustrated on selected medical applications.

## 2. The nature of medical data

The well developed information infrastructure of modern hospitals provides relatively inexpensive means to store the data, which can become widely available via internet/intranet. The rapidly emerging globality of data requires standards in terminology, vocabularies and formats to support data sharing, standards for interfaces between different sources of data and integration of heterogeneous data

```
IF    Sex = male
  AND Age > 46
  AND Number_of_painful_joints > 3
  AND Skin_manifestations = psoriasis
THEN  Diagnosis = Crystal_induced_synovitis
```

Fig. 1. An example if–then rule induced by CN2 in the domain of early diagnosis of rheumatic diseases.

(including images), and standards in the design of electronic patient records. Many environments still lack such standards, which hinders the use of data analysis tools on large global datasets, limiting their application to datasets collected for specific diagnostic, screening, prognostic, monitoring, therapy support or other patient management purposes.

Patient records collected for diagnosis and prognosis typically encompass values of anamnestic, clinical and laboratory parameters, as well as results of particular investigations, specific to the given task. Such datasets are characterized by their incompleteness (missing parameter values), incorrectness (systematic or random noise in the data), sparseness (few and/or non-representable patient records available), and inexactness (inappropriate selection of parameters for the given task). The development of machine learning tools for medical diagnosis and prediction was frequently motivated by the requirements for dealing with these characteristics of medical datasets [3,5].

Datasets collected in monitoring (either acute monitoring of a particular patient in an intensive care unit, or discrete monitoring over long periods of time in the case of patients with chronic diseases) have additional characteristics: they involve the measurements of a set of parameters at different times, requesting the temporal component to be taken into account in data analysis. These data characteristics need to be considered in the design of analysis tools for prediction, intelligent alarming and therapy support.

## 3. Selected data mining techniques

Current trends in medical decision making show awareness of the need to introduce formal reasoning, as well as intelligent data analysis techniques in the extraction of knowledge, regularities, trends and representative cases from patient data stored in medical records. Formal techniques include decision theory [17] and symbolic reasoning technology [35], as well as methods at their intersection, such as probabilistic belief networks [44]. Intelligent data analysis techniques include machine learning, clustering, data visualization, and interpretation of time-ordered data (derivation and revision of temporal trends and other forms of temporal data abstraction).

This paper is concerned with data mining methods for intelligent data analysis in medicine [33], in particular machine learning methods [36]. Machine learning methods can be classified into three major groups [36]: inductive learning of symbolic rules (such as induction of rules [7], decision trees [47] and logic programs [31]), statistical or pattern-recognition methods (such as $k$-nearest neighbors or instance-based learning [9,1], discriminate analysis and Bayesian classifiers), and artificial neural networks [51] (such as networks with back-propagation learning, Kohonen's self-organizing network and Hopfield's associative memory).

Machine learning methods have been applied to a variety of medical domains in order to improve medical decision making [26]. These include diagnostic and prognostic problems in oncology [3], liver pathology [34], neuropsychology [39], and gynaecology [43]. Improved medical diagnosis and prognosis may be achieved through automatic analysis of patient data stored in medical records, i.e. by learning from past experiences.

Given patient records with corresponding diagnoses, machine learning methods are able to diagnose new cases. More specifically, suppose $E$ is a set of examples with known classifications. An example is described by the values of a fixed collection of features (attributes): $A_i$, $i \in \{1,..., N_{\mathrm{at}}\}$. Each attribute can either have a finite set of values (discrete) or take real numbers as values (continuous). An individual example $e_j$, $j \in \{1,..., N_{ex}\}$ is a $n$-tuple of values $v_{i_k}$ of attributes $A_i$. Each example is assigned one of $N_{\mathrm{cl}}$ possible values of the class variable $C$ (classifications): $c_i$, $i \in \{1,..., N_{\mathrm{cl}}\}$. For instance, in the domain of early diagnosis of rheumatic diseases [12,30], the patient records comprise 16 anamnestic attributes. Some of these are continuous (e.g. age, duration of morning stiffness) and some are discrete (e.g. joint pain, which can be arthrotic, arthritic, or not present at all). There are eight possible diagnoses: degenerative spine diseases, degenerative joint diseases, inflammatory spine diseases, other inflammatory diseases, extraarticular rheumatism, crystal-induced synovitis, non-specific rheumatic manifestations, and non-rheumatic diseases.

To classify (diagnose) new cases, machine learning methods can take different approaches. They can construct explicit symbolic rules that generalize the training cases (rule induction and decision tree induction). The induced rules or decision trees can then be used to classify new cases. Another approach is to store (some of) the training cases for reference (instance-based learning). New cases can then be classified by comparing them to the reference cases. Yet another approach is to compute, for a given case to be classified, the conditional probability of classes according to the Bayesian formula and assign the most probable class to the case. These approaches are outlined in some more detail in the rest of this section.

### 3.1. Rule induction

Given a set of classified examples, a rule induction system constructs a set of if–then rules. An if–then rule has the form:

IF *Conditions* THEN *Conclusion*.

The condition part of a rule contains one or more attribute tests. The conclusion part has the form $C = c_i$, assigning a particular value $c_i$ to the class $C$. We say that an example is covered by a rule if the attribute values of the example fulfil the conditions in the IF part of the rule.

An example rule induced in the domain of early diagnosis of rheumatic diseases [12,30] is given in Fig. 1. It assigns the diagnosis of crystal-induced synovitis to male patients older than 46 years that have more than three painful joints and psoriasis as a skin manifestation.

The rule induction system CN2 [6,7] uses the covering approach to construct a set of rules for each possible class $c_i$ in turn: when rules for class $c_i$ are being constructed, examples of this class are positive, all other examples are negative. The covering approach works as follows: CN2 constructs a rule that correctly classifies some examples, removes the positive examples covered by the rule from the training set and repeats the process until no more examples remain. To construct a single rule that classifies examples into class $c_i$, CN2 starts with a rule with an empty antecedent (IF part) and the selected class $c_i$ as a consequent (THEN part). The antecedent of this rule is satisfied by all examples in the training set, and not only those of the selected class. CN2 then progressively refines the antecedent by adding conditions to it, until only examples of the class $c_i$ satisfy the antecedent. To allow for handling imperfect data, CN2 may construct a set of rules which is imprecise, i.e. does not classify all examples in the training set correctly.

Consider a partially built rule. The conclusion part is already fixed and there are some (possibly no) conditions in the IF part. The examples covered by this rule form the current training set. For discrete attributes, all conditions of the form $A_i = v_{i_k}$, where $v_{i_k}$ is a possible value for $A_i$, are considered for inclusion in the condition part. For continuous attributes, all conditions of the form $A_i < (v_{i_k} + v_{i_{k+1}})/2$ and $A_i > (v_{i_k} + v_{i_{k+1}})/2$ are considered, where $v_{i_k}$ and $v_{i_{k+1}}$ are two consecutive values of attribute $A_i$ that actually appear in the current training set. For example, if the values 4.0, 1.0, and 2.0 for attribute $A_i$ appear in the current training set, the conditions $A_i < 1.5$, $A_i > 1.5$, $A_i < 3.0$, and $A_i > 3.0$ will be considered.

Note that both the structure (set of attributes to be included) and the parameters (values of the attributes for discrete ones and boundaries for the continuous ones) of the rule are determined by CN2. Which condition will be included in the partially built rule depends on the number of examples of each class covered by the refined rule and the heuristic estimate of the quality of the rule.

The heuristic estimates used in rule induction are mainly designed to estimate the performance of the rule on unseen examples in terms of classification accuracy. This is in accord with the task of achieving high classification accuracy on unseen cases. Suppose a rule covers $P$ positive and $N$ negative examples of class $c_j$. Its accuracy an be estimated by the relative frequency of positive examples of class $c_j$ covered, computed as $P/(P+N)$. This heuristic, used in early rule induction algorithms, prefers rules which cover examples of only one class. The problem with this metric is that it tends to select very specific rules supported by only a few examples. In the extreme case, a maximally specific rule will cover (be supported by) one example and hence have an unbeatable score using the metrics of apparent accuracy (scores 100% accuracy). Apparent accuracy on the training data, however, does not adequately reflect true predictive accuracy, i.e. accuracy on new testing data. It has been shown [19] that rules supported by few examples have very high error rates on new testing data.

The problem lies in the estimation of the probabilities involved, i.e. the probability that a new example is correctly classified by a given rule. If we use relative frequency, the estimate is only good if the rule covers many examples. In practice, however, not enough examples are available to estimate these probabilities reliably at each step. Therefore, probability estimates that are more reliable when few examples are given should be used, such as the Laplace estimate which, in two-class problems, estimates the accuracy as $(P+1)/(P+N+2)$ [42]. This is the search heuristic used in CN2. The $m$-estimate [4] is a further upgrade of the Laplace estimate, taking into account also the prior distribution of classes.

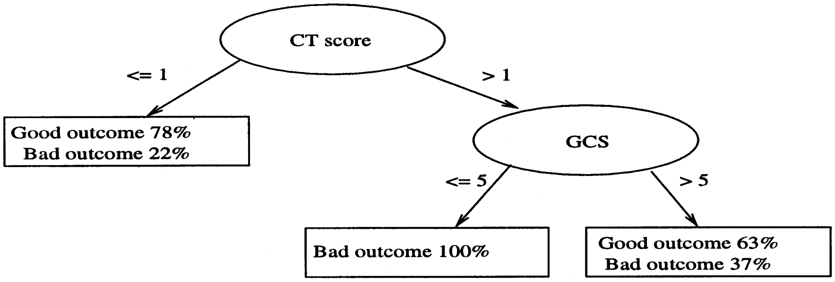


Fig. 2. Decision tree for outcome prediction after severe head injury. In the leaves, the percentages indicate the probabilities of class assignment.

### 3.2. Decision tree induction

Systems for top-down induction of decision trees (TDITD) [47] generate a decision tree from a given set of attribute-value tuples. Each of the interior nodes of the tree is labelled by an attribute, while branches that lead from the node are labelled by the values of the attribute.

The tree construction process is heuristically guided by choosing the 'most informative' attribute at each step, aimed at minimizing the expected number of tests needed for classification. Let $E$ be the current (initially entire) set of training examples, and $c_1,\ldots, c_{N_{\mathrm{cl}}}$ the decision classes. A decision tree is constructed by repeatedly calling a tree construction algorithm in each generated node of the tree. Tree construction stops when all examples in a node are of the same class (or if some other stopping criterion is satisfied). This node, called a leaf, is labelled by a value of the class variable. Otherwise the 'most informative' attribute, say $A_i$, is selected as the root of the (sub)tree, and the current training set $E$ is split into subsets $E_i$ according to the values of the most informative attribute. Recursively, a subtree $T_i$ is built for each $E_i$

Ideally, each leaf is labelled by exactly one class name. However, leaves can also be empty, if there are no training examples having attribute values that would lead to a leaf, or can be labelled by more than one class name (if there are training examples with same attribute values and different class names). One of the most important features is tree pruning, used as a mechanism for handling noisy data. Tree pruning is aimed at producing trees which do not overfit possibly erroneous data. In tree pruning, the unreliable parts of a tree are eliminated in order to increase the classification accuracy of the tree on unseen cases. The pruning techniques are based on the heuristic called the expected/predicted classification accuracy, or alternatively, the expected classification error [49].

An early decision tree learner ASSISTANT [5] that was developed specifically to deal with the particular characteristics of medical datasets, supports the handling of incompletely specified training examples (missing attribute values), binarization of continuous attributes, binary construction of decision trees, pruning of unreliable parts of the tree and plausible classification based on the 'naïve' Bayesian principle to calculate the classification in the leaves for which no evidence is available. A sample decision tree that can be used to predict outcome of patients after severe head injury [45] is shown in Fig. 2. The two attributes in the nodes of the tree are CT score (number of abnormalities detected by computer axial tomography) and GCS (evaluation of coma according to the Glasgow coma scale).

Recent implementations of the ASSISTANT algorithm include ASSISTANT-R and ASSISTANT-R2 [27]. Instead of the standard informativity search heuristic, ASSISTANT-R employs ReliefF as a heuristic for attribute selection [24,21]. This heuristic is an extension of RELIEF [20,21] which is a non-myopic heuristic measure that is able to estimate the quality of attributes even if there are strong conditional dependencies between attributes. In addition, wherever appropriate, instead of the relative frequency, ASSISTANT-R uses the $m$-estimate of probabilities, which typically improves the performance of machine learning algorithms [4]. ASSISTANT-R2 is a variant of ASSISTANT-R that, instead of building one general decision tree for the whole domain, generates one decision tree for each class (diagnosis). When classifying a new instance all trees are tried. If several trees classify the instance into its corresponding class the most probable class is selected. If none of the trees 'fires', the general tree for all the diagnoses generated by ASSISTANT-R is used.

The best known decision tree learner is C4.5 [49] (C5.0 is its recent upgrade) which is widely used and has been incorporated also into commercial data mining tools (e.g. Clementine and Kepler). The system is well maintained and documented, reliable, efficient and capable of dealing with large numbers of training examples. As such, it is considered to be one of the best data mining tools among those developed by the machine learning community.

### 3.3. Instance-based learning

Instance-based learning (IBL) algorithms [1] use specific instances to perform classification tasks, rather than generalizations such as induced if–then rules. IBL algorithms are also called lazy learning algorithms, as they simply save some or all of the training examples and postpone all effort towards inductive generalization until classification time. They assume that similar instances have similar classifications: novel instances are classified according to the classifications of their most similar neighbors.

IBL algorithms are derived from the nearest neighbor pattern classifier [15,8]. The nearest neighbor (NN) algorithm is one of the best known classification algorithms and an enormous body of research exists on the subject [9]. In essence, the NN algorithm treats attributes as dimensions of an Euclidean space and examples as points in this space. In the training phase, the classified examples are stored without any processing. When classifying a new example, the Euclidean distance between that example and all training examples is calculated and the class of the closest training example is assigned to the new example.

The more general $k$-NN method takes the $k$ nearest training examples and determines the class of the new example by majority vote. In improved versions of

$k$-NN, the votes of each of the $k$ nearest neighbors are weighted by the respective proximity to the new example [11]. An optimal value of $k$ may be determined automatically from the training set by using leave-one-out cross-validation [55]. In our experiments in early diagnosis of rheumatic diseases [12], using the $k$-NN algorithm implemented by Wettschereck [54], the best $k$ from the range [1, 75] was chosen in this manner. This implementation also incorporates feature weights determined from the training set. Namely, the contribution of each attribute to the distance may be weighted, in order to avoid problems caused by irrelevant features [56].

Let $n = N_{\mathrm{at}}$. Given two examples $x = (x_1,...,x_n)$ and $y = (y_1,...y_n)$, the distance between them is calculated as

$$\mathrm{distance}(x, y) = \sqrt{\sum_{i=1}^{n} w_i \times \mathrm{difference}(x_i, y_i)^2}$$

where $w_i$ is a non-negative weight value assigned to feature (attribute) $A_i$ and the difference between attribute values is defined as follows

$$\mathrm{difference}(x_i, y_i) = |x_i - y_i| \quad \text{if } A_i \text{ continuous}$$

$$\mathrm{difference}(x_i, y_i) = 0 \quad \text{if } A_i \text{ is discrete and } x_i = y_i$$

$$\mathrm{difference}(x_i, y_i) = 1 \quad \text{otherwise}$$

When classifying a new instance $z$, $k$-NN selects the set $K$ of $k$-nearest neighbors according to the distance defined above. The vote of each of the $k$ nearest neighbors is weighted by its proximity (inverse distance) to the new example. The probability $P(z, c_j, K)$ that instance $z$ belongs to class $c_j$ is estimated as

$$P(z, c_j, K) = \frac{\sum_{x \in K} x_{c_j}/\mathrm{distance}(z, x)}{\sum_{x \in K} 1/\mathrm{distance}(z, x)}$$

where $x$ is one of the $k$ nearest neighbors of $z$ and $x_{c_j}$ is 1 if $x$ belongs to class $c_j$. The class $c_j$ with largest value of $P(z, c_j, K)$ is assigned to the unseen example $z$.

Before training (respectively before classification), the continuous features are normalized by subtracting the mean and dividing by the standard deviation so as to ensure that the values output by the difference function are in the range [0, 1]. All features have then equal maximum and minimum potential effect on distance computations. However, this bias handicaps $k$-NN as it allows redundant, irrelevant, interacting or noisy features to have as much effect on distance computation as other features, thus causing $k$-NN to perform poorly. This observation has motivated the creation of many methods for computing feature weights.

The purpose of a feature weight mechanism is to give low weight to features that provide no information for classification (e.g. very noisy or irrelevant features), and to give high weight to features that provide reliable information. In the $k$-NN implementation of Wettschereck [54], feature $A_i$ is weighted according to the mutual information [53] $I(c_j, A_i)$ between the class $c_j$ and attribute $A_i$

### 3.4. Bayesian classifier

The Bayesian classifier uses the naive Bayesian formula to calculate the probability of each class $c_j$ given the values $v_{ik}$ of all the attributes for a given instance to be classified [22,23]. For simplicity, let $(v_1,\ldots, v_n)$ denote the $n$-tuple of values of example $e_k$ to be classified. Assuming the conditional independence of the attributes given the class, i.e. assuming $P(v_1,\ldots v_n|c_j) = \Pi_i P(v_i|c_j)$, then $P(c_j|v_1..v_n)$ is calculated as follows:

$$P(c_j|v_1..v_n) = \frac{P(c_j, v_1..v_n)}{P(v_1..v_n)} = \frac{P(v_1..v_n|c_j) \times P(c_j)}{P(v_1..v_n)} = \frac{\prod_i P(v_i|c_j) \times P(c_j)}{P(v_1..v_n)}$$

$$= \frac{P(c_j)}{P(v_1..v_n)} \prod_i \frac{P(c_j|v_i) \times P(v_i)}{P(c_j)} = P(c_j) \frac{\prod P(v_i)}{P(v_1..v_n)} \prod_i \frac{P(c_j|v_i)}{P(c_j)}$$

A new instance will be classified into the class with the maximal probability.

In the above equation, $\Pi i \; P(v_i)/P(v_1..v_n)$ is a normalising factor, independent of the class; it can therefore be ignored when comparing values of $P(c_j|v_1..v_n)$ for different classes $c_j$. Hence, $P(c_j|v_1..v_n)$ is proportional to:

$$P(c_j) \prod_i \frac{P(c_j|v_i)}{P(c_j)} \tag{1}$$

Different probability estimates can be used for computing the probabilities (e.g. the relative frequency, the Laplace estimate, the $m$-estimate). Instead of the simple relative frequeny estimate, computed as $N(c_j)/N_{\text{ex}}$, [22,23] use the Laplace law of succession for computing the prior probability [42]

$$P(c_j) = \frac{N(c_j) + 1}{N_{\text{ex}} + N_{cl}} \tag{2}$$

where $N_{\text{ex}}$ is the number of examples, $N_{\text{cl}}$ the number of classes, and $N(c_j)$ the number of examples of class $c_j$.

For computing the estimate of conditional probabilities [22,23] use the $m$-estimate [4]

$$P(c_j|v_i) = \frac{N(c_j \& v_i) + m \times p(c_j)}{N(v_i) + m}$$

where $N(Cond)$ stands for the number of examples for which $Cond$ is fulfilled, and $m$ is a user-defined parameter. The parameter $m$ trades-off the contribution of the relative frequency and the prior probability The default value $m = 2.0$ empirically gives good results [4].

The relative performance of the naive Bayesian classifier can serve as an estimate of the conditional independence of attributes.

Continuous attributes have to be pre-discretized in order to be used by the naive Bayesian classifier. The task of discretization is the selection of a set of boundary values that split the range of a continuous attribute into a number of intervals which are then considered as discrete values of that attribute. Discretization can be done manually by a domain expert or by applying a discretization algorithm [50].

The problem of (strict) discretization is that minor changes in the values of continuous attributes (or, equivalently, minor changes in boundaries) may have a drastic effect on the probability distribution and therefore on the classification. Fuzzy discretization may be used to overcome this problem by considering the values of the continuous attribute (or, equivalently, the boundaries of intervals) as fuzzy values instead of point values [23]. The effect of fuzzy discretization is that the probability distribution is smoother and the estimation of probabilities more reliable, which in turn results in more reliable classification.

### *3.5. Using background knowledge in learning*

The available patient data may be augmented with additional diagnostic knowledge which can be considered as additional information for the learner. In machine learning terminology, additional expert knowledge is usually referred to as *background knowledge*.

The main idea in LINUS [32,31] is to incorporate different attribute-value learning algorithms into an environment for inductive logic programming [40,31], which enables the effective use of specialist background knowledge in learning. LINUS also enables the induction of relational descriptions. Several attribute-value learners have been used within LINUS: a decision tree induction algorithm ASSISTANT [3,5], the rule-induction algorithms NEWGEM [38], the CN2 algorithm, and the *k*-NN implementation of Wettschereck [54].

LINUS thus extends attribute-value learners with the ability to use background knowledge and learn relational descriptions. From the ILP perspective, on the other hand, LINUS offers to use a variety of well-developed propositional learning tools to solve restricted forms ILP problems, involving the handling of noisy data and real numbers for which attribute-value learners have sophisticated mechanisms available.

In addition to training examples, LINUS is given background knowledge represented in the form of logical definitions of relations or functions, such as the functional definitions of groups of symptoms shown in Table 1. In the problem of early diagnosis of rheumatic diseases, a specialist for rheumatic diseases has namely provided his knowledge about the typical co-occurrences of symptoms. Six typical groupings of symptoms were suggested by the specialist as background knowledge

Table 1
Characteristic combinations of values for the attributes 'Joint pain' and 'Duration of morning stiffness'

| Joint pain | Morning stiffness | Grouping 1 value |
|---|---|---|
| No pain | $\leq 1$ h | No pain and dms $\leq 1$ h |
| Arthrotic | $\leq 1$ h | Arthrotic and dms $\leq 1$ h |
| Arthritic | $>1$ h | Arthritic and dms $>1$ h |

```
IF    Sex = male
  AND 28 < Age < 74
  AND Number_of_painful_joints < 17
  AND Number_of_swollen_joints < 8
  AND Eye_manifestations = no
  AND grouping3(Sex,Other_pain) = irrelevant
  AND grouping4(Joint_pain,Spinal_pain) = arthritic & no_pain)
THEN  Diagnosis = Crystal_induced_synovitis [0 1 0 1 0 12 0 0]
```

Fig. 3. An example if–then rule induced by CN2 in the domain of early diagnosis of rheumatic diseases, using the LINUS transformation for dealing with the available background knowledge. Tuple [0 1 0 1 0 12 0 0] denotes the number of examples, from each of the eight classes, covered by the rule (e.g. the rule covers 12 examples of crystal induced synovitis).

to be considered by the learner [30]. One of the groupings relates the attribute 'Joint pain' and the attribute 'Duration of morning stiffness'. The characteristic combinations are given in Table 1, whereas all other combinations are insignificant or irrelevant.

The second of the six groupings relates Spinal pain and Duration of morning stiffness. The following are the characteristic combinations: no spinal pain and morning stiffness up to 1 h, spondylotic pain and morning stiffness up to 1 hour, spondylitic pain and morning stiffness longer than 1 h. The third grouping relates the attributes Sex and Other pain. Indicative is the pain in the thorax or in the heels for male patients, all other combinations are non-specific: the corresponding values of this grouping are 'male and thorax' and 'male and heels'.

The fourth grouping relates joint pain and spinal pain. Grouping 5 relates Joint pain, Spinal pain and Number of painful joints, and the sixth grouping relates Number of swollen joints and Number of painful joints.

Using the background knowledge, LINUS generates attributes that are not present in the initially given attribute set and extends the training examples with these attributes. The transformed problem can be addressed by an attribute-value learner. A sample if-then rule induced by CN2 is shown in Fig. 3.

An approach to automating the acquisition of background knowledge of co-occurrences of groups of symptoms from a dataset is presented in [58]. This method, based on function decomposition, was evaluated on the same problem of early diagnosis of rheumatic diseases.

### 3.6. Inductive logic programming

LINUS is an environment for inductive logic programming (ILP) [31], enabling learning of relational descriptions by transforming the training examples and background knowledge into the form appropriate for attribute-value learners. In general, however, inductive logic programming systems learn relational descriptions without such a transformation to propositional learning. The best known ILP systems include FOIL [48], Progol [41] and Claudien [10].

```
class(Image, Segment, undermining) :-
  clockwise(Segment, Adjacent, 1),
  class_confirmed(Image, Adjacent, undermining).
```

Fig. 4. A Prolog clause induced by GKS in the domain of ocular fundus image classification for glaucoma diagnosis.

In ILP, induced rules typically have the form of Prolog clauses. A rule for ocular fundus image classification for glaucoma diagnosis, as induced by an ILP system GKS [37] specially designed to deal with low-level measurement data including images, is given to illustrate the output of an ILP algorithm.

Compared to rules induced by a rule learning algorithm of the form IF *Conditions* THEN *Conclusion*, Prolog rules have the form *Conclusion*:- *Conditions*. The rule in Fig. 4, for example, means that Segment of Image is classified as undermining (i.e. not normal) if the conditions of the right-hand side of the clause are fulfilled. Notice that the conditions consist of a conjunction of predicate clockwise/3 defined in the background knowledge, and predicate class_confirmed/3, added to the background knowledge in one of the previous iterative runs of the GKS algorithm. This shows one of the features of ILP learning, namely that learning can be done in several cycles of the learning algorithm in which definitions of new background knowledge predicates are learned and used in the subsequent runs of the learner; this may improve the performance of the learner.

## 4. Selected measures for performance evaluation

Measures for performance evaluation depend on the learning task. If the task is diagnosis or prognosis, classification accuracy is the most frequently used quality evaluation measure, in addition to the interpretability of results (the issue discussed in Section 5). However, even in medical diagnosis and prognosis, the classification accuracy is not necessarily the best quality measure of a classifier. Selected evaluation measures are outlined below.

### *4.1. Classification accuracy*

Classification accuracy was already mentioned in this paper in the context of rule induction, since it can be used as a heuristic for guiding the search and as a rule quality evaluation criterion for deciding when to stop the search. Recall that, for two-class problems, the accuracy of a rule can be estimated as $P/(P+N)$ (relative frequency of positive examples covered by the rule), or $(P+1)/(P+N+2)$ (Laplace estimate), where $P$ is the number of positive and $N$ the number of negative examples of the selected class covered by the rule.

As opposed to the above classification accuracy as used for the evaluation of a single rule on the covered training examples, consider the evaluation of the quality of a classifier (a rule or a set of rules) on unseen cases, i.e. cases from a separate testing set. Let us assume a two-class classification problem (classes ‘positive’ and ‘negative’).

Consider four subsets: True positives (TP): True positive answers of a classifier denoting correct classifications of positive cases; True negatives (TN): True negative answers denoting correct classifications of negative cases; False positives (FP): False positive answers denoting incorrect classifications of negative cases into class positive; False negatives (FN): False negative answers denoting incorrect classifications of positive cases into class negative.

The *classification accuracy* measures the proportion of correctly classified cases:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

### 4.2. Sensitivity and specificity

For medical applications, two other measures are more frequently used than the classification accuracy: sensitivity and specificity [29,52,37]. *Sensitivity* measures the fraction of positive cases that are classified as positive. *Specificity* measures the fraction of negative cases classified as negative.

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

In many medical problems, high classification accuracy is less important than high sensitivity and/or specificity of a classifier’s answers.

Sensitivity can be viewed as a detection rate that one wants to maximize. If the goal is to increase the sensitivity of answers, the learner should try to increase the correct classifications of positive cases (*TP*) and/or decrease the number of incorrect classifications of positive cases into class negative (*FN*).

On the other hand, in order to increase the specificity, the learner should try to increase the number of correct classifications of negative cases (*TN*) and/or decrease the number of incorrect classifications of negative cases into class positive (*FP*). Note that 1-*Specificity* can be interpreted as a false alarm rate which one wants to minimize.

A ROC curve (*receiver operating characteristic*) indicates a trade-off that one can achieve between the false alarm rate (1-*Specificity*, plotted on *X*-axis) that needs to be minimized, and the detection rate (*Sensitivity*, plotted on *Y*-axis) that needs to be maximized (see e.g. [29]). An appropriate trade-off, determined by the expert, can be achieved by applying different algorithms, as well as by different parameter settings of a selected data mining algorithm.As shown in [29], improved performance in terms of sensitivity, specificity and classification accuracy can be achieved by the adaptation of selected data mining methods (Bayesian classifier, decision tree learning, neural networks and $k$-nearest neighbors) to dealing with misclassification costs in the estimation of probabilities.

### 4.3. Post-test probability

It is important to recognise that medical diagnosis and prognosis are typically stepwise processes. Diagnosis starts by first collecting anamnestic data. In the study of early diagnosis of rheumatic diseases [46,30] subsequent steps involve the evaluation of clinical manifestations, laboratory findings and finally radiological findings.

In a recent study in coronary artery disease (CAD) prognosis [18,29], the first next step concerns the evaluation of signs, symptoms and some laboratory results that are evaluated clinically and ECG performed at rest. Positive and/or inconclusive results require a further step: the sequential ECG testing during a controlled exercise. The next step involves myocardial perfusion scintigraphy to be performed. At each step of this sequential diagnostic process, the diagnostic value of each test is most often assessed by its sensitivity and specificity [18]. Recall that sensitivity represents the fraction of patients with positive results of the test where the diagnosis CAD is confirmed, and specificity the fraction of patients without the disease which will be excluded by the test. The test results are interpreted probabilistically, i.e. after the test, the probability of the presence of CAD is given as the *post-test probability* ($P_2$), computed according to the Bayes theorem from the *pre-test probability* ($P_1$), sensitivity ($Se$) and specificity ($Sp$) as follows:

$$P_2 = \frac{P_1 \times Se}{P_1 \times Se + (1 - P_1) \times (1 - Sp)}, \quad \text{if the test result is positive}$$

$$P_2 = \frac{P_1 \times (1 - Se)}{P_1 \times (1 - Se) + (1 - P_1) \times Sp}, \quad \text{if the test result is negative}$$

Sufficient diagnostic certainty is considered if the post-test probability is either higher than 90 or lower than 10%. In particular if the post-test probability is below 10% the patient can be, according to the doctrine, excluded from medical control and diagnostic procedures [18].

The application of data mining tools, appropriately adapted to improve the sensitivity and specificity of results, may thus improve the results of post-test probability computation. Consequently, many patients could avoid being submitted to invasive and costly diagnostic procedures, thus improving the cost-benefit trade-off of diagnostic procedures applied.

### 4.4. Information score

The information score of induced rules [25] is a performance measure for classifiers. The most general answer a classifier can give is a probability distribution over the $N_{\mathrm{cl}}$ classes. The information score takes into account the prior probabilities of classes and can appropriately deal with probabilistic answers of the classifiers.

Let the correct class of example $e_k$ be $c_i$, its prior probability $P_a(c_i)$ and the probability returned by the classifier $p(c_i)$. The information score of this answer $I(e_k)$ is computed as follows:

$$I(e_k) = -\log P_a(c_i) + \log P(c_i) \quad P(c_i) \geq P_a(c_i)$$

$$I(e_k) = \log(1 - P_a(c_i)) - \log(1 - P(c_i)), \quad P(c_i) < P_a(c_i)$$

As $I(e_k)$ indicates the amount of information about the correct classification of $e_k$ gained by the classifier's answer, it is positive if $P(c_i) > P_a(c_i)$, negative if the answer is misleading $P(c_i) < P_a(c_i)$ and zero if $P(c_i) = P_a(c_i)$

The *average information score* $I_{av}$ of the answers of a classifier on a testing set, consisting of examples $e_1, e_2, \ldots, e_t$, is calculated as:

$$I_{\mathrm{av}} = \frac{1}{t} \times \sum_{k=1}^{t} I(e_k)$$

To scale the evaluation of a classifier to the difficulty of the problem, the relative information score takes into account the prior probability of classes (diagnoses). Namely, a correct classification into a more probable diagnosis provides less information than a correct classification into a rare diagnosis, which is only represented by a few training examples. For example, in domains where one of the diagnostic classes is highly likely, it is easy to achieve high classification accuracy. The 'default' classifier that assigns the most common diagnosis to all patients would in that case have undeservedly high classification accuracy. However, its relative information score is zero, indicating that the classifier provides no information at all.

### 4.5. Misclassification cost

Consider a learning problem with $N_{\mathrm{cl}}$ classes. Let $e_k$ be an example from a set of $t$ testing examples: $e_1, e_2, \ldots, e_t$. Let the class of example $e_k$ be $c_i$, and the probability returned by a classifier $p(c_i)$.

A classifier *correctly classifies* example $e_k$ if $P(c_i) > P(c_j)$, $\forall j \in \{1, \ldots, N_{\mathrm{cl}}\}$ Example $e_k$ is *misclassified* by a classifier if there exists class $c_{\mathrm{l}}$ such that $P(c_l) = \max_j P(c_j)$, $j \in \{1, \ldots, N_{\mathrm{cl}}\}$, and $P(c_{\mathrm{l}}) > \mathrm{P}(c_i)$

Let us now introduce the misclassification cost matrix **C**, consisting of elements $c_{ij} = cost(c_i, c_j)$ representing the cost of misclassifying an example from class $c_i$ as class $c_j$, and $\forall i$, $cost(c_i, c_i) = 0$.

Suppose that a classifier assigns a most probable class to an example to be classified. If the example is misclassified to class $c_{\mathrm{l}}$ then the cost of misclassifying example $e_k$ is $C(e_k) = \mathrm{cost}(c_i, c_{\mathrm{l}})$.

In the more general case when a classifier returns a probability distribution over the $N_{\mathrm{cl}}$ classes, and an example belonging to class $c_i$ (with prior probability $P_{\mathrm{a}}(c_i)$) is misclassified then, according to [29], the expected cost of misclassifying the example belonging to class $c_i$ is

$$C(e_k) = \frac{1}{1 - P_a(c_i)} \times \sum_{j \neq i} P_a(c_j) \ \mathrm{cost}\,(c_i, c_j)$$

The *average misclassification cost* of the answers of a classifier on a testing set of examples $e_1, e_2, \ldots, e_t$ is computed as follows:

$$C_{\mathrm{av}} = \frac{1}{t} \times \sum_{k=1}^{t} C(e_k)$$

See [29] for further information on misclassification costs.

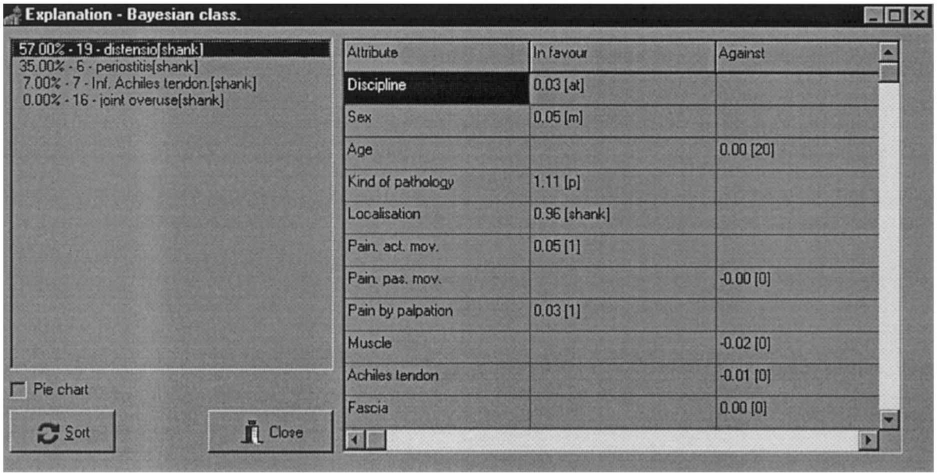
## 5. Interpretability of results of selected applications

In medical diagnosis it is crucial that any computerised system is able to explain and justify its decisions when diagnosing a new patient. Especially when faced with an unexpected solution of a new problem, the user requires substantial justification and explanation.

Decision tree learners often give an appropriate explanation: induced decision trees are fairly easy to understand and can be used to support diagnosis without using the computer—this is particularly valuable in situations which require prompt decisions and in situations when computer interaction is psychologically unacceptable. Positions of attributes in the tree, especially the top (most informative) ones, often directly correspond to domain expert's knowledge. However, in order to produce general rules, these methods use pruning [47,5] which drastically reduces tree sizes. Consequently, the paths from the root to the leaves are shorter, containing only few, most informative attributes. Frequently physicians dislike such trees since too few parameters are taken into account and the tree describes the patients too poorly (not sufficiently detailed) to provide reliable decisions. Another problem is the variability of decision trees—frequently a small change in the dataset causes a substantial restructuring of the decision trees—this also decreases the physician's trust in the proposed diagnosis and in its explanation.

As reported in [2], in the application of the ASSISTANT decision tree learner in early diagnosis of rheumatic diseases [46], a specialist was asked to evaluate the interpretability of induced decision trees. The expert considered many of the induced trees as 'unnatural' and therefore unsatisfactory. Cited from [2]: 'This was despite the fact that the measured diagnostic accuracy of the trees was significantly above the measured accuracy of medical experts themselves in this domain.' The study [2] reports that one specific expert's comment was that 'the trees didn't tell enough, they should give more information'. The expert was told that the trees had in fact been pruned, because of noise in the learning data, in order to optimize their accuracy. The expert was then encouraged to modify the trees to include the 'missing information', by adding parts of the pruned tree. In spite of the decrease in accuracy, the expert found larger trees preferable to the ones originally induced.

Similar observations were recorded in our experiments in which the rule learner CN2 was applied to induce rules for early diagnosis of rheumatic diseases [30,31]. Again many of the induced rules seemed unnatural since, according to the expert, a lot of information needed for diagnosis was not present in the induced rules. The evaluation of induced rules [30] showed the expert's dissatisfaction with the interpretability of induced rules. The expert's satisfaction slightly increased (but not substantially) when background knowledge was added to the definition of the learning task and used by LINUS to provide additional attributes for the CN2 rule learner.



Explanation - Bayesian class.

57.00% - 19 - distensio[shank]
35.00% - 6 - periostitis[shank]
7.00% - 7 - Inf. Achiles tendon.[shank]
0.00% - 16 - joint overuse[shank]

| Attribute | In favour | Against |
|---|---|---|
| Discipline | 0.03 [at] | |
| Sex | 0.05 [m] | |
| Age | | 0.00 [20] |
| Kind of pathology | 1.11 [p] | |
| Localisation | 0.96 [shank] | |
| Pain. act. mov. | 0.05 [1] | |
| Pain. pas. mov. | | -0.00 [0] |
| Pain by palpation | 0.03 [1] | |
| Muscle | | -0.02 [0] |
| Achiles tendon | | -0.01 [0] |
| Fascia | | 0.00 [0] |

Fig. 5. Naive Bayes: An explanation of the decision in the diagnosis of sport injuries.

Similar experiences with physicians who typically disliked short decision trees and short rules led to the study of explanation capabilities of the naive Bayesian classifier [23,28]. Bayesian classifiers induce a table of conditional probabilities which indicate how much a feature (an attribute value) contributes to a diagnosis. When explaining a decision for an individual patient, the explanation of a decision is provided by the indicated ‘weight’ of a feature, i.e. the information gain for each patient’s feature, as well as the sum of information gains of all features that are in favor or against the individual decision (diagnosis).

Applying the function $-\log_2$ to Eq. (1) shows that $-\log_2 P(c_j|v_1..v_n)$ is proportional to:

$$-\log_2 P(c_j) - \sum_i (\log_2 P(c_j|v_i) - \log_2 P(c_j)) \tag{3}$$

The *information gain* of attribute value $v_i$ for class $c_j$, measured in bits, is defined as the difference between the prior and posterior information necessary to determine the class $c_j$:

$$\log_2 P(c_j|v_i) - \log_2 P(c_j) \tag{4}$$

where $c_j$ denotes an individual diagnosis, $P(c_j)$ the prior probability, and $P(c_j|v_i)$ the conditional probability.

Eq. (3) shows that the explanation of the decision is the difference between the prior information necessary to determine the class $c_j$ and the sum of information gains of all the features. One of the main advantages of this approach, which is appealing to physicians, is that all the available information is used to explain the decision; such an explanation seems to be 'natural' for medical diagnosis and prognosis [26].

Physicians like the explanation of decisions given by the naive Bayesian classifier since in their opinion the sum of information gains in favor/against a given diagnosis appears to be close to the way how they diagnose patients. For example, Fig. 5 provides a sample explanation for a diagnosis proposed by the naive Bayesian classifier in an application that is trying to reveal the yet unclear influence of anamnestic and clinical parameters for individual diagnoses of sports injuries [57]. The expert involved liked the explanation provided by the naive Bayesian classifier since it uses all the available attributes for classification.

## 6. Discussion

The aim of this paper is to present a variety of data mining techniques and to discuss some of their features used for medical problem solving. In addition to the high accuracy requirement, the derivation of symbolic rules is a must when the goal of data mining is to enable the interpretation of proposed solutions. Sensitivity, specificity, information score, post-test probability and misclassification cost are discussed as alternative measures to classification accuracy for evaluating the quality of a classifier.

This paper is based on the author's invited talk at the Second International Conference on the Practical Application of Knowledge Discovery and Data Mining, London, 25-27 March 1998 (PADD 1998 Proceedings, The Practical Application Company, pages 11–31). The paper is not comprehensive in the sense that the presented techniques are mainly selected from the arsenal of techniques developed and used by the members of the Department of Intelligent Systems at J. Stefan Institute and the Artificial Intelligence Laboratory of the Faculty of computer and information sciences in Ljubljana, Slovenia.

## Acknowledgements

Thanks to Igor Kononenko, Matjaž Kukar, Ivan Bratko, Igor Mozetič, Bojan Cestnik, Sašo Džeroski and other members of the two research groups for their contributions to this area which influenced the views expressed in this paper. Thanks also to Igor Zelič and the medical experts Jure Fettich, Ciril Groselj, Iztok Pilih, Vlado Pirnat, Tine Prevec and Vanja Vuga, involved in the medical problem domains. I wish to thank Elpida Keravnou and Blaz Zupan for the joint work in the area of intelligent data analysis in medicine that resulted in IDAMAP workshops organised in 1996 and 1997, an edited book published by Kluwer in 1997 [33] and this special issue of AIM. The work on this paper was financially supported by the Slovenian Ministry of Science and Technology.

## References

[1] Aha D, Kibler D, Albert M. Instance-based learning algorithms. Mach Learn 1991;6:37–66.

[2] Bratko I, Machine learning: Between accuracy and interpretability. In: Della Riccia G, Lenz HJ, Kruse R, editors. Learning, Networks and Statistics. Berlin: Springer, CISM Courses and Lectures No.382. 1986:163–177.

[3] Bratko I, Kononenko I. Learning diagnostic rules from incomplete and noisy data. In: Phelps B, editor. AI Methods in Statistics. London: Gower Technical Press 1987.

[4] Cestnik B. Estimating probabilities: A crucial task in machine learning. In: Proceedings European Conference on Artificial Intelligence 1990:147–149.

[5] Cestnik B, Kononenko I, Bratko I. ASSISTANT 86: a Knowledge Elicitation Tool for Sophisticated Users. In: Bratko I, Lavrač N, editors. Progress in Machine learning. Wilmslow: Sigma Press 1987.

[6] Clark P, Boswell R. Rule induction with CN2: Some recent improvements. In: Proceedings of the Fifth European Working Session on Learning. Berlin: Springer 1991, 151–163.

[7] Clark P, Niblett T. The CN2 induction algorithm. Mach Learn 1989;3(4):261–83.

[8] Cover TM, Hart PK. Nearest neighbor pattern classification. IEEE Trans Inf Theory 1968;13:21–7.

[9] Dasarathy BV. editor. Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. Los Alamitos: IEEE Computer Society Press, 1990.

[10] De Raedt L, Dehaspe L. Clausal discovery. Mach Learn 1997;26:99–146.

[11] Dudani SA. The distance-weighted $k$-nearest neighbor rule. IEEE Trans Syst Man Cybern 1975;6(4):325–7.

[12] Džeroski S, Lavrač N. Rule induction and instance-based learning applied in medical diagnosis. Technol Health Care 1996;4(2):203–21.

[13] Fayyad UM, Uthurusamy R. Data mining and knowledge discovery in databases (editorial). Commun ACM 1996;39(11):24–6.

[14] Fayyad UM, Piatetsky-Shapiro G, Smyth P. The KDD process for extracting useful knowledge from volumes of data. Commun ACM 1996;39(11):27–41.

[15] Fix E, Hodges JL. Discriminatory Analysis. Nonparametric Discrimination; Consistency Properties. Technical Report, 4 Randolph Field, TX: US Air Force School of Aviation Medicine, 1957.

[16] Frawley W, Piatetsky-Shapiro G, Matheus C. Knowledge discovery in databases: an overview. In Piatetsky-Shapiro G, Frawley W, editors. Knowledge discovery in databases. Menlo Park, CA: The AAAI Press, 1991.

[17] French S. Decision Theory. Chichester: Ellis Horwood, 1986.

[18] Grošelj C, Kukar M, Fettich JJ, Kononenko I. Impact of machine learning to the diagnostic certainty of the patient's group with low coronary artery disease probability. In: Proceedings of Workshop on Computer-Aided Data Analysis in Medicine CADAM-97, Ljubljana, J. Stefan Institute, 1997;68–74.

[19] Holte R, Acker L, Porter B. Concept learning and the problem of small disjuncts. In: Proceedings Tenth International Joint Conference on Artificial Intelligence. San Mateo, CA: Morgan Kaufmann 1989.

[20] Kira K, Rendell L. A practical approach to feature selection. In: Proceedings International Conference on Machine Learning. San Mateo, CA: Morgan Kaufmann 1992;249–256.

[21] Kira K, Rendell L. The feature selection problem: Traditional methods and new algorithm. In: Proceedings AAAI 1992, San Jose, CA, 1992.

[22] Kononenko I. Semi-naive Bayesian classifier. In: Proceedings European Working Session on Learning-91. Berlin: Springer 1991;206–219.

[23] Kononenko I. Inductive and Bayesian learning in medical diagnosis. Appl Artif Intell 1993;7:317–37.

[24] Kononenko I. Estimating attributes: Analysis and extensions of Relief. In: Proceedings European Conference on Machine Learning. Berlin: Springer 1994;171–182.

[25] Kononenko I, Bratko I. Information based evaluation criterion for classifier's performance. Mach Learn 1991;6(1):67–80.

[26] Kononenko I, Kukar M. Machine learning for medical diagnosis. In: Proceedings Workshop on Computer-Aided Data Analysis in Medicine; CADAM-95. Ljubljana: IJS Scientific Publishing, 1995.

[27] Kononenko I, Šimec E. Induction of decision trees using RELIEFF. In: Proceedings ISSEK Workshop on Mathematical and Statistical Methods in Artificial Intelligence. Berlin: Springer 1995;199–220.

[28] Kukar M, Kononenko I, Silvester T. Machine learning in prognosis of the femoral neck fracture recovery. Artif Intell Med 1996;8:431–51.

[29] Kukar M, Kononenko I, Grošelj C, Kralj K, Fettich JJ. Analysing and improving the diagnosis of ischaemic heart disease with machine learning, Artif. Intell. Med. this special issue on Data Mining Techniques and Applications in Medicine. Elsevier 1998.

[30] Lavrač N, Džeroski S, Pirnat V, Križman V. The utility of background knowledge in learning medical diagnostic rules. Appl Artif Intell 1993;7:273–93.

[31] Lavrač N, Džeroski S. Inductive Logic Programming: Techniques and Applications. Chichester: Ellis Horwood, 1994.

[32] Lavrač N, Džeroski S, Grobelnik M. Learning nonrecursive definitions of relations with LINUS. In: Proceedings Fifth European Working Session on Learning. Berlin: Springer 1991;265–281.

[33] Lavrač N, Keravnou E, Zupan B, editors. Intelligent Data Analysis in Medicine and Pharmacology, Kluwer, 1997.

[34] Lesmo L, Saitta L, Torasso P. Learning of fuzzy production rules for medical diagnosis. In: Gupta M, Sanchez E, editors. Approximate Reasoning in Decision Analysis. Amsterdam: North-Holland, 1982.

[35] Lucas PJF. Logic engineering in medicine. Knowl Eng Rev 1995;10(2):153–79.

[36] Michie D, Spiegelhalter DJ, Taylor CC, editors. Machine learning; neural and statistical classification. Chichester: Ellis Horwood, 1994.

[37] Mizoguchi F, Ohwada H, Daidoji M, Shirato S. Using inductive logic programming to learn classification rules that identify glaucomatous eyes. In: Lavrač N, Keravnou E, Zupan B, editors. Intelligent Data Analysis in Medicine and Pharmacology. Kluwer 1997;227–242.

[38] Mozetič I. NEWGEM: Program for Learning from Examples; Technical Documentation and User's Guide. Reports of Intelligent Systems Group UIUCDCS-F-85-949, Department of Computer Science, University of Illinois. Urbana Champaign, IL.

[39] Muggleton S. Inductive Acquisition of Expert Knowledge. Wokingham: Addison-Wesley, 1990.

[40] Muggleton S. Inductive logic programming. New Generation Comp 1991;8(4):295–318.

[41] Muggleton S. Inverse entailment and Progol. New Generation Comp Special Issue on Inductive Logic Programming 1995;13(3–4):245–86.

[42] Niblett T, Bratko I. Learning decision rules in noisy domains. In: Bramer M, editor. Research and Development in Expert Systems III. Cambridge University Press 1986;24–25.

[43] Nunez M. Decision tree induction using domain knowledge. In: Wielinga B, editor. Current Trends in Knowledge Acquisition. Amsterdam: IOS Press, 1990.

[44] Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Mateo, CA: Morgan Kaufmann, 1988.

[45] Pilih I.A, Mladenič D, Lavrač N, Prevec TS. Data analysis of patients with severe head injury. In: Lavrač N, Keravnou E, Zupan B, editors. Intelligent Data Analysis in Medicine and Pharmacology. Kluwer, 1997;131–148.

[46] Pirnat V, Kononenko I, Janc T, Bratko I. Medical estimation of automatically induced decision rules. Proceedings 2nd Europ. Conf. on Artificial Intelligence in Medicine, London, 1989;24–36.

[47] Quinlan JR. Induction of decision trees. Mach Learn 1986;1(1):81–106.

[48] Quinlan JR. Learning logical definitions from relations. Mach Learn 1990;5(3):239–66.

[49] Quinlan JR. C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann, 1993.

[50] Richeldi M, Rossotto M. Class-driven statistical discretization of continuous attributes. In: Machine Learning: Proceedings ECML-95. Berlin: Springer 1995;335–342.

[51] Rumelhart DE, McClelland JL, editors. Parallel Distributed Processing, vol. 1: Foundations. Cambridge, MA: MIT Press, 1986.

[52] Shankle W.R, Mani S, Pazzani M.J, Smyth P. Dementia screening with machine learning methods In: Lavrač N, Keravnou E, Zupan B, editors. Intelligent Data Analysis in Medicine and Pharmacology. Kluwer. 1997;149–166.

[53] Shannon CE. A mathematical theory of communication. Bell Syst Tech J 1948;27:379–423.

[54] Wettschereck D. A study of distance-based machine learning algorithms, PhD Thesis. Corvallis, OR: Department of Computer Science, Oregon State University, 1994.

[55] Weiss SM, Kulikowski CA. Computer Systems that Learn. San Mateo, CA: Morgan Kaufmann, 1991.

[56] Wolpert D. Constructing a generalizer superior to NETtalk via mathematical theory of generalization. Neural Netw 1989;3:445–52.

[57] Zelič I, Kononenko I, Lavrač N, Vuga V. Induction of decision trees and Bayesian classification applied to diagnosis of sport injuries. J Med Syst 1997;21(6):429–44.

[58] Zupan B, Džeroski S. Acquiring and validating background knowledge for machine learning using function decomposition. In: Proceedings 6th Conference on Artificial Intelligence in Medicine, Berlin: Springer 1997;86–97.