Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Pattern Recognition Letters 26 (2005) 43–56

Pattern Recognition Letters

www.elsevier.com/locate/patrec

A feature selection technique for classificatory analysis

Amir Ahmad [a], Lipika Dey [b,*]

[a] Solid State Physics Laboratory, Timarpur, Delhi 110054, India
[b] Department of Mathematics, Indian Institute of Technology, Hauz Khas, New Delhi 110016, India

Received 19 July 2004
Available online 18 October 2004

Abstract

Patterns summarizing mutual associations between class decisions and attribute values in a pre-classified database, provide insight into the significance of attributes and also useful classificatory knowledge. In this paper we have proposed a conditional probability based, efficient method to extract the significant attributes from a database. Reducing the feature set during pre-processing enhances the quality of knowledge extracted and also increases the speed of computation. Our method supports easy visualization of classificatory knowledge. A likelihood-based classification algorithm that uses this classificatory knowledge is also proposed. We have also shown how the classification methodology can be used for cost-sensitive learning where both accuracy and precision of prediction are important.

Keywords: Feature selection; Significance of attributes; Classificatory knowledge extraction

1. Introduction

Feature selection for classification is a well-researched problem, aimed at reducing the dimensionality and noise in data sets (Dash and Liu, 1997). In this paper we propose a feature selection technique using conditional-probability-based significance measures for features. Each feature is assigned a significance value determined by its separability and capability to distinguish elements of different classes. Our significance computations are motivated by the feature-value distance measures proposed in (Stanfill and Waltz, 1986) and (Cost and Salzberg, 1993) for categorical features. In (Cost and Salzberg, 1993) the frequency-based distance measures are used to determine the weights to be assigned for different exemplars while implementing a weighted k-nearest neighbour algorithm for classification (PEBLS). However, we do not compute inter-value distances for features. Our scheme looks for mutual exclusion of distribution of feature values in different classes. The measure of frequency of a value in one class and non-occurrence of the same value in other

[*] Corresponding author. Tel.: +91 11 26591487.
E-mail addresses: amir_ahmed/sspl@ssplnet.org (A. Ahmad), lipika@maths.iitd.ernet.in (L. Dey).

0167-8655/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2004.08.015

classes simultaneously, is used to determine the significance of a feature. The significance of the value of a feature determines the contribution of the feature towards classificatory decision.

In this paper, we have also presented a likelihood-based classification algorithm which exploits the class–attribute associations extracted for feature selection. We have shown how the feature selection and the classification procedures can be employed for designing cost-sensitive learning schemes.

2. Related work

Feature selection is a mature area of research. We will first present a brief overview of the different approaches followed and then present the distinguishing features of our work in comparison to the existing approaches.

2.1. Feature selection techniques—a brief survey

Blum and Langley (1997) classify the feature selection techniques into three basic approaches. In the first approach, known as the *embedded approach*, a basic induction method is used to add or remove features from the concept description in response to prediction errors on new instances. Quinlan's ID3 (Quinlan, 1986) and C4.5 (Quinlan, 1993), CART proposed in (Breiman et al., 1984), are some of the most successful supervized learning algorithms. These algorithms use a greedy search through the space of decision trees, at each stage using an evaluation function to select the attribute that has the best ability to discriminate among the classes.

The second approach is known as the *filtering approach*, in which, various subsets of features are explored to find an optimal subset, which preserves the classification knowledge. Michalski (1980) proposed the AQ learning algorithm, which uses positive and negative examples of a class along with a user defined criterion function, to identify a disjunctive feature set that can maximize the positive events and minimize the negative events. Narendra and Fukunaga (1977) presented a Branch and Bound algorithm for finding the optimal feature set that uses a top-down approach with back-tracking. Pudil et al. proposed a set of suboptimal algorithms called the floating search methods (Pudil et al., 1994) that do not require the fulfillment of monotonicity condition for feature selection criterion function. Somol et al. provides a modified and efficient branch and bound algorithm for feature selection in (Somol et al., 2000). Though computationally less expensive than the Branch-and Bound algorithms, there exists no theoretical upper bound on the computational costs of the algorithms because of their heuristic nature.

John et al. proposed another feature selection framework (John et al., 1994), known as the *wrapper technique.* The wrapper methods evaluate alternative feature sets by running some induction algorithm on the training data and using the estimated accuracy of the resulting classifier as its metric. The major disadvantage of the wrapper methods is in the computational cost involved in running the induction algorithm repeatedly for each feature set considered.

A number of feature selection techniques based on the *evolutionary approaches* have also been proposed. Casillas et al. (2001) presents a genetic feature selection technique which is integrated into a multi-stage genetic learning process to obtain a Fuzzy Rule Based Classification system (FRBCS). In the first phase of this method, a filtering approach is used to determine an optimal feature subset for a specific classification problem using class-separability measures. This feature subset along with expert opinion is used to obtain the adequate feature subset cardinality in the second phase, which is used as the chromosome length. Xiong (2002) proposed a hybrid approach to input selection, which distinguishes itself from existing filter and wrapper-based techniques, but utilizes the advantages of both. This process uses case-based reasoning to select candidate subsets of features which are termed as "hypothesis". The performance of case-based reasoning under a hypothesis is estimated using training data on a "leave-one-out" procedure. The error estimate is then combined with the subset of selected attributes to provide an evaluation function for the GA to find the optimal hypothesis. Kuncheva and Bezdek proposed a genetic algorithm for simultaneous editing and feature selection to design 1-nn classifiers (Kuncheva and Bezdek, 1998). They had posed the problem as bi-criteria combinatorial optimization problem having an NP-hard search space. Ho et al. (2002) proposed the design of an optimal nearest neighbor classifier using intelligent genetic algorithm. Thawonmas and Abe (1997) suggests a feature selection technique to eliminate irrelevant features, based on analysis of class regions generated by a fuzzy classifier. The degree of overlaps in a class region is used to define exception ratio, and the features that have the lowest sum of exception ratios are the relevant ones. Irrelevant features are eliminated using a backward selection search technique.

Kira and Rendell proposed a different approach to feature selection in (Kira and Rendell, 1992). The RELIEF algorithm proposed by them assigns a weight to each feature based on the ability of the feature to distinguish among the classes and then selects those features whose weights exceed a user defined threshold, as relevant. The weight computation is based on the probability of the nearest neighbors from two different classes having different values for an attribute and the probability of two nearest neighbors of the same class having same value of the attribute. Higher the difference between these two probabilities, more significant is the attribute. Inherently, the measure is defined for a two-class problem which can be extended to handle multiple classes, by splitting the problem into a series of two-class problems. Kononenko suggests the use of *k*-nearest neighbours to increase the reliability of the probability approximation (Kononenko, 1994). It also suggests how RELIEF can be extended to work with multiple sets more efficiently. Weighting schemes are easier to implement and are preferred for their efficiency.

Learning to classify objects is inherently difficult problem for which several approaches like instance-based learning or nearest neighbor-based algorithms are used. However, the nearest neighbor algorithms need some kind of distance measure. Cost and Salzberg (1993) emphasized on the need to select appropriate metrics for symbolic values. Stanfill and Waltz (1986) proposed the Value Difference Metric (VDM) which measures distance between values of symbolic features. It takes into account the overall similarity of classification of all instances for each possible value of each feature. Based on this, Cost and Salzberg (1993) proposed the Modified Value Distance Metric (MVDM) which is symmetric, and satisfies all the metric properties. They have shown that nearest neighbour algorithms perform well even for symbolic data using this metric. It is observed that distance-values are similar if the pairs occur with the same relative frequency for all classes.

2.2. Distinct aspects of the proposed work

It may be observed from the earlier discussion, that the majority of the feature selection techniques have not considered the problem of classification as an integrated problem. ID3 and its derivatives like C4.5 are exceptions to this approach. These are decision-tree-based supervised learning systems. Another popular classification algorithm that is used with a selected subset of features is the *k*-nearest neighbour whose results depend on the choice of the correct value of *k*. PEBLS provides a means of learning the weights for weighing these *k* neighbours appropriately, to get good classification results.

As mentioned earlier, our work was motivated by Cost and Salzberg (1993). It may be observed that while most feature selection or weighing methods do consider the relative frequencies of a feature value in different classes, the mutual exclusion of occurrence of a value in different classes is not usually considered. Our proposed method for computing the significance measure of an attribute is based on the rationale that a significant feature is likely to have different values for different classes while this may not be so for an insignificant feature. The relative frequency of an attribute value across different classes gives a measure of the attribute value-to-class and class-to-attribute value associations. We store these associations, and show how they can form a part of classificatory knowledge. This also provides a good visualization of the distinguishing characteristics of the different classes.

A unique aspect of the proposed approach is the integration of the feature selection technique to a classification algorithm. We have proposed a likelihood-based classification algorithm, which uses the significance of a feature value for classification decision making. We have obtained very good results for a large number of data sets including the high-dimensional ones like DNA data sets.

Finally we have shown how the proposed technique can be employed for designing cost-sensitive learning schemes. Cost sensitivity related to classification has generated a lot of interest in recent times, since it is being increasingly realized that different classes of errors should incur different penalties for most of the real-world problems. If the cost of an error is known a priori, it is possible to build a cost matrix for the misclassification model. Rather than making a series of weighted classifiers, which is very expensive, appropriately biased classification techniques can be evolved based on this cost-matrix (Domingos, 1999). Our likelihood-based classifier can be easily biased to develop a cost-sensitive learning scheme.

The rest of the paper is organized as follows. Section 3 describes the design principles of the proposed algorithms to compute significance of features and thereby, select the relevant features for classification. Section 4 presents how we represent the classificatory knowledge and the design of the classification algorithm. In Section 5 we have presented performance evaluation measures obtained on some well-known data sets.

3. Determining significance of symbolic attributes—a probabilistic approach

In this paper we have proposed a feature selection technique, in which features are assigned significance values based on the intuition that if an attribute is significant, then there is a strong possibility that elements with complementary sets of values for this attribute will belong to complementary sets of classes. Alternatively, given that the class decisions for two sets of elements are different, it is expected that the significant attribute values for these two sets of elements should also be different. We compute the significance of an attribute as a two-way function of its association to the class decision.

For each attribute A_i, we compute the overall *attribute-to-class association* denoted by Æ(A_i). Æ(A_i) captures the cumulative effect of all possible values of A_i and their associations to class decisions. Similarly, we take note of how an attribute's values change with a change in the class decision. We capture this effect in a quantity Œ(A_i) for the attribute A_i. This represents the *class-to-attribute association* for attribute A_i. An attribute is really *significant* if both attribute-to-class association and class-to-attribute association for the attribute are high. In the remaining part of this section, we elaborate on the physical significance and computational aspects of these two quantities.

3.1. Computing Æ( ) for all attributes

We start with the observation, that for a significant attribute, a change in the attribute value should cause a change in the class decision. Let U be the collection of pre-classified data elements and let $A_1, A_2, \ldots, A_g$ be the attributes which describe the elements of this data set. We assume that the elements of U are members of $\boldsymbol{m}$ different classes denoted by natural numbers $1, 2, \ldots, \boldsymbol{m}$. Let $\boldsymbol{J}$ represent the set of all class labels i.e. $\boldsymbol{J} = \{1, 2, 3, \ldots, \boldsymbol{m}\}$.

To compute the overall association of A_i to the different classes, let us assume that it can take k different symbolic values. We use the notation A_i^r to denote the rth attribute value of A_i. The notation $A_i^{\sim r}$ is used to denote a value of A_i which is not equal to A_i^r. This is a short hand notation for all values not equal to A_i^r, and can actually take $(k-1)$ different values.

We introduce a set of notations which we will use hereafter.

- Let w be a proper subset of $\boldsymbol{J}$.
- Let $P_i^r(w)$ denote the probability that elements of U with ith attribute value equal to A_i^r belong to classes contained in w. This can be computed from U using frequency counts.
- Let $P_i^{\sim r}(\sim w)$ denote the probability that elements not having the ith attribute value equal to A_i^r (i.e. elements with ith attribute value equal to anything other than A_i^r) do not belong to classes contained in w. This can also be computed from U using frequency counts.

Our first observation is that if an attribute value is very significant, then both $P_i^r(w)$ and $P_i^{\sim r}(\sim w)$ are high. This implies that objects with ith attribute (A_i) value equal to A_i^r and those with $A_i^{\sim r}$ *classify to different groups of complementary classes.*

We term the quantity $(P_i^r(w) + P_i^{\sim r}(\sim w))$ as the *separating power of* A_i^r *with respect to* w. This quantity reaches a maximum, when both the terms individually reach their maxima. Since there are $(2^m - 1)$ possible values of w, we associate with each value A_i^r, the subset w_i^r, which yields the maximum value for the summation $(P_i^r(w) + P_i^{\sim r}(\sim w))$.

Definition 3.1.1. The subset $w = w_i^r$ that maximizes the term $(P_i^r(w) + P_i^{\sim r}(\sim w))$ is termed as the *support set* for the value A_i^r.

Since w_i^r yields the maximum value for the above quantity, this subset can be said to have the strongest association to the value A_i^r. We present an efficient algorithm in Section 3.1.1 which can find this maximizing set without actually considering all the $(2^m - 1)$ subsets.

Definition 3.1.2. Let $\vartheta_i^r = (P_i^r(w_i^r) + P_i^{\sim r}(\sim w_i^r))$. ϑ_i^r is defined as the *discriminating power* of an attribute value A_i^r, where w_i^r is the support set for the value A_i^r.

An attribute will be significant if all it's values have high discriminating power. The following properties establish some crucial properties of the separating power of an attribute value.

Property 1. For any i, for any r and for any w, $0 \leqslant P_i^r(w) \leqslant 1$, and $0 \leqslant P_i^{\sim r}(\sim w) \leqslant 1$.

Property 2. The value of the discriminating power of an attribute value A_i^r lies between 1.0 and 2.0 i.e. $1.0 \leqslant (P_i^r(w_i^r) + P_i^{\sim r}(\sim w_i^r)) \leqslant 2.0$, where w_i^r is the support set for A_i^r.

Definition 3.1.3. The *attribute-to-class association* of an attribute A_i, denoted by Æ(A_i), is a function of the mean of the discriminating powers of all possible values of an attribute A_i. For an attribute A_i with k different attribute values, Æ(A_i) lies between 0.0 and 1.0, and is computed as follows:

$$\text{Æ}(A_i) = \left(1/k \sum_{r=1,2,\ldots,k} \vartheta_i^r\right) - 1.0 \tag{3.1}$$

In the next section, we elaborate on how to obtain the maximizing support set for an attribute value efficiently. We also discuss how Æ(A_i) is calculated thereof.

3.1.1. An incremental approach to finding the support set for an attribute value

We will now present a linear incremental approach to finding the support set w_i^r for an attribute value A_i^r. Thus all possible subsets of w do not have to be explored, to find the maximizing subset. The incremental approach ensures that one class is examined at most once for inclusion into the support set. This is particularly significant, since otherwise computation complexity would grow exponentially with the number of classes in the data set.

To compute $P_i^r(w)$ for any $w \subset \boldsymbol{J}$, we note that $P_i^r(w) = \sum_{t \in W}(P(t \mid A_i^r))$, where $P(t \mid A_i^r)$ denotes the conditional probability that an element belongs to class t given that the value for its ith attribute is A_i^r. This is because an element can belong to exactly one class contained in w. This can be directly computed for any given pre-classified data set.

Computation of $P_i^r(w_i^r) + P_i^{\sim r}(\sim w_i^r)$ where w_i^r is the support set, involves the following basic tasks:

Task 1. Finding the maximizing subset, w_i^r.
Task 2. Computing $P_i^r(w_i^r)$.
Task 3. Computing $P_i^{\sim r}(\sim w_i^r)$.

Task 1. *Finding* w_i^r: We use a linear incremental algorithm to find the support set w_i^r for each attribute value A_i^r of attribute A_i. This is done through algorithm FIND_SUPPORT_SET(A_i^r) explained next.

Algorithm. (FIND-SUPPORT-SET(A_i^r))

```
(i)  Initialize $w_i^r$ = NULL and $\vartheta_i^r = 0.0$.
(ii) For each class $t$
         If $P(t \mid A_i^r) > P(t \mid A_i^{\sim r})$
         then {add $t$ to $w_i^r$;
```

```
        $\vartheta_i^r = \vartheta_i^r + P(t \mid A_i^r)$;
        }
    else {add $t$ to $\sim w_i^r$;
        $\vartheta_i^r = \vartheta_i^r + P(t/A_i^{\sim r})$;}

    End FIND_SUPPORT_SET
```

Thus, if the conditional probability of an element belonging to class $\boldsymbol{t}$ is higher with a given attribute value A_i^r, than with the values $A_i^{\sim r}$, then $\boldsymbol{t}$ will be included in w_i^r, while it will be included in $(\sim w_i^r)$, if it is the other way round. Obviously, no class can belong to both w_i^r and $(\sim w_i^r)$. Thus, when all the classes t are taken care of, w_i^r accumulates those classes which occur more frequently in association to the value A_i^r for A_i, while $(\sim w_i^r)$ accumulates those classes which occur more frequently in association with $A_i^{\sim r}$. Theorem 1 given in the Appendix A considers all possible alternatives and proves that Algorithm FIND-SUPPORT-SET will indeed find the maximizing support set correctly.

Task 2. *Computing* $P_i^r(w_i^r)$: Let $\boldsymbol{n}$ denote the total number of elements in the data set. For each class $t \in \boldsymbol{J}$, let $\boldsymbol{N}(t)$ denote the number of elements belonging to class t. Let $\boldsymbol{T}_i^r$ denote the total number of elements in the data set having A_i^r as the value for A_i. Let $\boldsymbol{M}_i^r(\boldsymbol{t})$ denote the number of elements that belong to class t and have attribute value A_i^r for A_i. Thus

$$P(t \mid A_i^r) = \boldsymbol{M}_i^r(\boldsymbol{t})/\boldsymbol{T}_i^r \tag{A.1}$$

Hence,

$$P_i^r(w_i^r) = \sum_{t \in w_i^r} P(t \mid A_i^r) \tag{A.2}$$

where t is selected as described in task 1.

Task 3. *Computing* $P_i^{\sim r}(\sim w_i^r)$: Now, to compute this, we first have to compute $P(t \mid A_i^{\sim r})$, i.e. the proportion of elements which belong to class t but does not have attribute value A_i^r for A_i, out of all the elements of the data set.

The quantity $(\boldsymbol{N}(t) - \boldsymbol{M}_i^r(t))$ denotes the number of elements which belong to class t but does not have attribute value A_i^r for A_i. The total number of elements in the data set which does not have the attribute value A_i^r for A_i is given by $(\boldsymbol{n} - \boldsymbol{T}_i^r)$. Thus the required conditional probability $P(t \mid A_i^{\sim r})$ is given by

$$P(t \mid A_i^{\sim r}) = (\boldsymbol{N}(t) - \boldsymbol{M}_i^r(t))/(\boldsymbol{n} - \boldsymbol{T}_i^r) \tag{A.3}$$

Hence, as earlier,

$$P_i^{\sim r}(\sim w_i^r) = \sum_{t \in (\sim w_i^r)} P(t \mid A_i^{\sim r}) \tag{A.4}$$

3.1.2. Complexity of the proposed method

It requires one scan of the database to compute the probabilities defined in tasks 2 and 3. If there are $\boldsymbol{g}$ attributes in the database, then the computation of ϑ_i^r has to be done for all the $\boldsymbol{g}$ attributes, for all values. If an attribute has $\boldsymbol{k}$ categorical values, then the steps in Task 1 are repeated once for each attribute for each of its value. Since, step 2 of task 1 is an iterative step over the number of classes, so the total number of times this step is executed is $\boldsymbol{gkm}$, where $\boldsymbol{m}$ is the number of classes in the database. Hence, the total time complexity of this algorithm is O($\boldsymbol{gn} + \boldsymbol{gkm}$). Thus in a very large data base, where usually $\boldsymbol{n} \gg \boldsymbol{gkm}$ (Cost and Salzberg, 1993), the proposed algorithm becomes effectively linear in terms of the total number of elements in the database i.e. it is O($\boldsymbol{gn}$).

3.2. Computing Œ( ) for all attributes

Œ(A_i) finds the association between the attribute A_i and various class decisions, by observing how a change in the class decision causes a change in the attribute's value. It is expected that objects belonging to different classes will tend to have different values for a really significant attribute. The computation of Œ(A_i) is very similar to the earlier computation.

Let V be a subset of attribute values of A_i. As in Section 3.1, we introduce two quantities $P_i^j(V)$ and $P_i^{\sim j}(\sim V)$.

- $P_i^j(V)$ denotes the probability that elements belonging to class j, have those attribute values of A_i which are contained in the set V.
- $P_i^{\sim j}(\sim V)$ denotes the probability that elements not belonging to class j, have those attribute values of A_i which are not contained in the set V.

- Now, for each class $j \in \boldsymbol{J}$, we find the subset V_i^j comprised of values of A_i, that maximizes the quantity $(P_i^j(V) + P_i^{\sim j}(\sim V))$. Thus V_i^j contains those values of attribute A_i, which occur predominantly in association to class j. High values for both $P_i^j(V_i^j)$ and $P_i^{\sim j}(\sim V_i^j)$ indicate that the values contained in V_i^j have a high association factor with class j, and the remaining classes have high association with other values of attribute A_i.

Definition 3.2.1. The quantity $(P_i^j(V_i^j) + P_i^{\sim j}(V_i^{\sim j}))$ is denoted by Λ_i^j and is called the *separability* of the attribute values of A_i with respect to class j.

We now define the quantity called +(A_i), which denotes the *class-to-attribute association* for the attribute A_i, as the mean of the separability of its values. Further, we restrict +(A_i) to lie between 0.0 and 1.0 and hence we define it as follows:

$$\text{Œ} + (A_i) = (1/\boldsymbol{m}) * \left(\sum_{j=1,2,\ldots,m} \Lambda_i^j \right) - 1.0 \qquad (3.2)$$

where the database D has elements of $\boldsymbol{m}$ different classes.

Definition 3.2.2. The significance of an attribute A_i is computed as the average of Æ(A_i) and Œ(A_i) and is denoted by $\sigma(A_i)$.

The significance of each attribute in the database is computed using Definition 3.2.2.

3.3. Attributes and their support sets—physical significance

In this section we will illustrate the practical use of ranking of attributes and also explain the physical significance of the support sets associated with the significant attributes. We will illustrate the significance of the support sets with some practical data. A database containing heart patients' data, obtained from www.niaad.liacc.up.pt/statlog/datasets.html, contains elements of two classes—patients with and without heart disease. This set contains 13 attributes. Table 1 shows the ranking of the most significant attributes obtained by our method along with the attribute values in their respective support sets for the class of patients with heart disease. The most significant attribute is *thal* which had three possible values: 3 = normal; 6 = fixed defect; 7 = reversible defect. The support sets for diseased category contains the attribute values 6 and 7 only indicating that heart patients have either "fixed defect" or "reversible defect" for the *thal* factor. The support set for non-heart patient class contains the value 3, indicating that "normal" value of this attribute would most likely belong to a person who is not a heart patient. Similarly, the attribute *number of major blood vessels colored by fluoroscopy* has values 0, 1, 2 or 3. The support set for diseased patient category in this case includes values 1, 2 and 3, indicating that heart patients have one or more vessels blocked.

Table 1
Significant attributes and their support sets for the heart disease data

Rank	Attribute name	Support set for heart-patient class	Support set for non-heart-patient class
1	Thal	{6—fixed defect, 7—reversible defect}	{3—normal}
2	Number of major blood vessels colored by fluoroscopy	{1, 2, 3}	{0}
3	Chest pain type	{4}	{1, 2, 3}
4	Exercise induced angina	{Yes}	{No}
5	Slope of peak exercise ST segment	{Medium, High}	{Low}
6	Oldpeak = ST depression induced by exercise relative to rest	{2.06–6.2}	<2.06
7	Maximum heart rate achieved	{71.0–136.0}	{136.1–168.7}
8	Sex	Female	Male

Using another practical data set, we now illustrate how the support sets can be used to represent classificatory knowledge. The image segmentation data also obtained from the above site, contains pixels classified into categories BRICKFACE, SKY, FOLIAGE, CEMENT, WINDOW, PATH and GRASS. Each pixel is described with 19 attributes, of which we find the seven most significant ones are *intensity*, *rawred-mean*, *rawgreen-mean*, *rawblue-mean*, *value-mean*, *saturation-mean and hue-mean* respectively. The other attributes convey positional information only and are correctly identified as insignificant for classification by our approach. On analysis of support sets for the significant attributes, we can extract the following classificatory knowledge for image segmentation:

Rule 1: If *intensity* = HIGH and *rawred-mean* = HIGH and *rawgreen-mean* = HIGH and *rawblue-mean* = HIGH and *value-mean* = HIGH, then *class* = SKY.
Rule 2: If *Hue-mean* = HIGH then *class* = GRASS.
Rule 3: If *Hue-mean* = MEDIUM then *class* = WINDOW.
Rule 4: If *Hue-mean* = LOW then *class* = BRICKFACE.

Looking at the support sets for classes CEMENT and PATH, it was found that for all significant attributes, all of them were identical for these two classes. Thus it can be predicted that it would be difficult to distinguish between the tuples of these two classes. The confusion matrix for classification of test data for this data set confirms this.

The support sets help in identifying correlated features very easily. Strongly correlated features have similar partitioning of support sets.

4. Classification of new elements

In this section, we propose a classification scheme using the support set of an attribute value. The support set of an attribute value contains those classes, which have a high degree of association with that particular value. Thus we compute the likelihood of a class for a new data element by considering whether it belongs to the support set of the attribute value or not.

For a given attribute value A_i^r of the new element, the likelihood of the element belonging to a particular class $\boldsymbol{t}$, on the basis of this attribute's value is given by Eq. (A.3), otherwise by Eq. (A.4).

$$\text{Likelihood}(\boldsymbol{t}) = (\vartheta_i^r - 1.0) * P_i^r(\boldsymbol{t}), \quad \text{if } \boldsymbol{t} \in w_i^r \tag{4.1}$$

$$\text{Likelihood}(\boldsymbol{t}) = 0.0 \quad \text{otherwise} \tag{4.2}$$

where w_i^r is the support set for value A_i^r. The likelihood of each class for a data element is given by its summation over all the significant attribute values. The class that receives the maximum total contribution is predicted as the actual class of the data element. Since ϑ_i^r and $P_i^r(t)$ are pre-computed quantities, this computation is of the order of O($\boldsymbol{g}'\boldsymbol{m}$) only, where $\boldsymbol{g}'$ is the number of significant attributes and $\boldsymbol{m}$ is the total number of classes.

5. Performance evaluation

The best validation for any classification data mining system can be obtained by judging its classification performance. In this section we will illustrate the performance of the proposed algorithms on a number of standard data sets obtained from the sites www.niaad.liacc.up.pt/statlog/datasets.html and the UCI repository. These data sets contain pre-classified data from various domains. For each data set on which we have experimented, our first aim was to extract the significance of the attributes used for that set. Numeric attributes were discretized using equal interval discretization. Based on Dougherty et al.'s study (Dougherty et al., 1995), which suggest that 10 intervals are satisfactory for equal value discretization, we have also used 10 intervals for all the results reported in this paper. After extracting the significance of the attributes, we ordered them and selected a suitable subset of attributes for classification purposes. The results reported in this section were obtained with a 10-fold cross-validation over each data set.

5.1. Classification using significant attributes

We will first establish the correctness of the significance values computed for the features through three different exercises.

We first compare the results of simple k-nearest neighbour classification to a weighted k-nearest neighbour algorithm, where the weights are the significance of the attributes. In simple k-nearest neighbour-based classification technique (Duda et al., 2000), one finds k-nearest neighbours of the element to be classified. The class assigned to the new element is taken by a majority decision. The distance between the new element and a training sample is given by $(\sum(x_i - y_i)^2)^{1/2}$. In the weighted approach we have used the distance measure $(\sum(\sigma_i(x_i - y_i))^2)^{1/2}$, where σ_i denotes the significance of feature A_i, computed using our technique.

Columns 2 and 3 of Table 2 show the results for the unweighted and the weighted approaches respectively. The results are better for the weighted approach, which shows that the significance values of features are computed correctly. Next we compared the results of weighted-k-nearest neighbor classification done with all attributes, against that obtained by applying the same algorithm but for a selected subset of the attributes only. Column 5 of Table 2 shows the number of attributes selected by our approach for each domain. Columns 6 and 7 of Table 2 show the results of k-nearest classification using the weighted and unweighted distance measure, using a selected features only. It may be noted that there is always a reduction in classification error with weighted k-nn. There is also a significant gain in performance when irrelevant attributes are eliminated. This also shows the effectiveness of the proposed feature selection algorithm.

One of the most crucial steps for the implementation of our feature selection method is to decide on a threshold for selecting the significant attributes. Empirical observations show that if for the most significant attribute A_i (say) in the data base $\sigma(A_i)$ is less than 0.8, then only those attributes for which both attribute-to-class association i.e. Æ() and class-to-attribute association i.e. Œ() are greater than sixty percent of $\sigma(A_i)$, contribute significantly to the prediction of class of a new instance. If in a database the most significant attribute has $\sigma(A_i)$ value greater than 0.8, then for that database only those attributes contribute significantly to the class of an instance which have Æ() and Œ() values greater than eighty percent of the highest value. However, these are only empirical observations and we are yet to provide any theoretical basis for the selection of the threshold value. Table 3 shows the highest significance values obtained for an attribute in nine data sets and also the threshold value for each of them.

We will now show the performance of our proposed likelihood-based classification method based on feature selection.

Table 4 presents a comparative study of the proposed classification algorithm against those obtained by C4.5 (obtained from www.niaad.liacc.up.pt/statlog/datasets.html and Wu, 1999), and PEBLS (Cost and Salzberg, 1993). It may be observed that classification performance obtained

Table 2
Classification accuracy improves with weighted k-nn. It also improves with reduction in insignificant attributes

Data set	Number of attributes in dataset	Error with all attributes (%)		Number of significant attributes	Error with significant attributes only (%)	
		Distance function without significance of attributes	Distance function with significance of attributes		Distance function without weight	Distance function with significance of attributes
Iris	4	6.9	6.6	2	4.9	4.8
Credit	8	32.7	29.9	3	29.9	27.5
Wine	13	4.5	3.3	9	3.2	2.9
Vehicle	18	33.7	32.9	7	34.9	34.1
Ionosphere	34	25.8	14.6	17	22.5	10.9
Image segment	19	5.7	2.9	11	3.4	2.5

Table 3
Significance value of the most significant attribute and the threshold value of cut-off for various data sets

Dataset	Highest value of significance of an attribute	Threshold significance of attribute
Iris	0.84	0.67
Aus-Credit	0.71	0.43
Diabetes	0.37	0.22
Hayes-Roth	0.34	0.20
Vote	0.89	0.71
Wine	0.62	0.37
Heart	0.47	0.28
Vehicle	0.45	0.27
DNA	0.51	0.31

by using our approach is quite encouraging for most of the domains. There is an improvement in classification accuracy. The only exception is that of the domain "Hayes Roth". The reason for poor performance in this domain may be attributed to the fact that none of the attributes were really very significant in this data set. Our feature selection algorithm indicated that all features had very low significance values. Only one feature was found to be irrelevant. We applied our algorithm for high dimensional data sets like the DNA data set also. Classification results for the DNA data set is reported in www.niaad.liacc.up.pt/statlog/. Classification accuracy obtained for this set using C4.5 is reported there as 96% for training samples and 92.4% for the test samples. Using our approach, the number of significant features detected is 18 and the accuracy of classification is 93.5% with 10-fold cross-validation.

We also tried to see whether there was performance gain by using only the significant attributes chosen by our method and using C4.5 with default settings and pruning option set to yes, as mentioned in (Quinlan, 1993) as the classification algorithm. However, results in this case were not so encouraging. This was expected since every classification algorithm works best with its own evaluation function.

Though it is beyond the scope of the proposed work, it is worth mentioning that the choice of the discretization method plays a significant role in the performance of inductive learning algorithms as has been shown in (Ching et al., 1995). A class-dependent discretization technique has also been proposed in the above paper. This paper reports an improvement in performance for learning algorithms like AQ and ID3 with pre-pruning and post-pruning, using their discretization technique. As shown there for the Iris data set, the best classification accuracy was 95.2% (error—4.8%) for ID3 with pre-pruning and 96.3% (error 3.7%) for M-APACS, which is what we also obtained for our proposed classification method (shown in Table 4) with equal value discretization.

5.2. Classification using multiple features at a time

To observe the effect of combining multiple features over classification performance, we consid-

Table 4
Comparison of classification accuracy using proposed method and C4.5 and PEBLS

Database	Number of instances	Total no of attributes	Classification error with C4.5	Classification error with PEBLS	Number of attributes selected by our method	Classification error using support sets and likelihood function
Iris	150	4	6.7	6.3	2	3.7
Aus-Credit	690	14	15.5	17.8	2	14.4
Diabetes	768	8	27.0	30.6	3	20.7
Hayes-Roth	160	4	14.4	14.5	3	18.1
Vote	435	16	3.0	5.8	2	3.9
Wine	178	13	1.9	3.6	9	5.8
Heart	270	13	30.1	22.9	10	13.0
Vehicle	846	18	33.3	39.5	7	38.0
DNA	3186	60	7.6	7.3	18	6.5

Table 5
Classification results using two feature subsets

Data set	Classification error with one feature-class	Classification error with a pair of features-class correlation
Hayes-Roth	18.1	16.2
Vehicle	38.0	35.6
DNA	6.5	5.3

ered all possible two-feature subsets and computed the significance of these combinations using the attribute-to-class and class-to-attribute associations. Thus with g attributes we consider ${}^{g}C_2$ combinations of feature pairs. Our computation mechanism remains same, while the categorical values for the attributes are now derived from combined labels of the two attributes in the feature subset under consideration. On experimenting with the data sets mentioned in Table 4, most of the data sets did not show any remarkable improvement in classification accuracy. Only three data sets showed some improvement which are illustrated in Table 5.

We observe that there can be some improvement in classification accuracy obtained by using more than one feature at a time, particularly when none of the attributes have very high significance values and thereby do not produce very accurate classification results. However, the time complexity of computing the significance of feature sets grows significantly. As the number of features in a subset is increased, the problem grows combinatorially. Since there is very little improvement in classification accuracy, so we did not proceed with this approach any further. However, it may be worth investigating this approach in combination with heuristic branch-and-bound techniques.

5.3. Cost-sensitive learning schemes

For some domains, it is not enough to report the classification accuracy. Rather they call for a more detailed cost-sensitive error analysis. Typically, medical data or credit-card data analyses fall under this category. The results in these cases are analyzed using a 2 by 2 matrix recording the number of True Positives (TP), True Negatives (TN), False Positives (FP) and False negatives (FN). Witten and Frank (2000) suggests a general technique for building cost-sensitive classifiers by varying the proportion of instances in the training set. A cost matrix is designed in which each entry has a cost associated with it, which determines the total reward or the total penalty. Rewards are for correct identification of TP and TN instances. The other two incur penalty. The total penalty is calculated as a sum of individual penalty multiplied by the number of entries in that category.

We implemented cost-sensitive learning schemes for two data sets—the heart-patients database and German Credit Card database. In both the cases, FP and FN instances have to be minimized in order to minimize the total penalty. However, we may associate a larger penalty for not diagnosing a heart patient than for diagnosing a non-patient as a patient. If we call absence of heart disease as a positive case, then this can be achieved by penalizing a FP (a heart patient not diagnosed) five times more than that of a FN (a non-heart patient incorrectly diagnosed). Similarly, for the credit card database, a larger penalty is incurred if a bad customer is identified as a good customer, since this may lead to a greater loss. Table 6 shows the cost matrix for the two domains.

A cost-sensitive classifier can be easily designed using our likelihood-based classification scheme. We skew the likelihood of class “Present” for the heart data base, by adding a positive quantity (Δ) to the right hand side of Eq. (4.1), while keeping the likelihood of “Absent” as earlier. We started with a value of 0.05 for Δ and increased

Table 6
Cost matrix for German credit data (left) heart-patients data (right)

Class	ACTUAL Good	ACTUAL Bad	Class	ACTUAL Absent	ACTUAL Present
PREDICTED Good	0 (TP)	5 (FP)	PREDICTED Absent	0 (TP)	5 (FP)
PREDICTED Bad	1 (FN)	0 (TN)	PREDICTED Present	1 (FN)	0 (TN)

Table 7
Cost-sensitive learning using biased likelihood for "negative" classes

Data set	Penalty with our proposed biased likelihood functions	Penalty with C4.5	Other best reported results	
			Algorithm name	Penalty
Heart	39.9	78.1	Bayes	37.4
German credit	53.2	98.5	Discrim	53.5

it by 0.05 till performance starts dropping. For the German Credit Card database, the class "bad" was biased. The final value of Δ was found to be 0.35 for the heart database and 0.5 for the credit card database. All classification results were obtained by using the significant attributes only. Table 7 summarizes the total penalty incurred by our method against the best such results reported in literature which we obtained from the site www.niaad.liacc.up.pt/statlog/datasets.html. All results are averages for 10-fold cross-validation, with respect to the cost matrix presented in Table 6.

6. Conclusions

As real-world databases are normally large and noisy, the problem of focusing on relevant information has become increasingly important in data mining. In this paper, we have presented a new algorithm to compute significance of attributes and then select a subset of features to be used for classification purposes. The proposed algorithm works with initial conditional probabilities which is computed through one scan of the data base. We have also proposed a classification method based on this approach. Results show that the performance of this algorithm is comparable to some of the well-known algorithms. We have also shown that this approach can be used very effectively for cost-sensitive learning. Cost-sensitive learning mechanisms are very useful for real-world data sets like those derived from medical and financial domains.

This work is being currently extended to extract multi-level classification rules using the support sets associated to attribute values. We are also working on computing inter-object distances based on similar measures for unsupervized learning or clustering of data sets.

Appendix A

Theorem 1. *Algorithm FIND_SUPPORT_SET-(A_i^r) finds the set $w_i^r \subset \boldsymbol{J}$, such that for any other $w \subset J$ and $w \neq w_i^r$, $(P_i^r(w) + P_i^{\sim r}(\sim w)) < (P_i^r(w_i^r) + P_i^{\sim r}(\sim w_i^r))$.*

Proof. Let $J = \{1, 2, 3, \ldots, m\}$, the set of all classes. Since, w and w_i^r are subsets of J, therefore the following relations hold:

$$P_i^r(w_i^r) = \sum_{t \in w_i^r} P(t \mid A_i^r) \tag{A.2}$$

$$P_i^{\sim r}(\sim w_i^r) = \sum_{t \in (\sim w_i^r)} P(t \mid A_i^{\sim r}) \tag{A.4}$$

Therefore,

$$P_i^r(w) = \sum_{t \in w} P(t \mid A_i^r) \tag{A.2}$$

$$P_i^{\sim r}(\sim w) = \sum_{t \in (\sim w).} P(t \mid A_i^{\sim r}) \tag{A.4}$$

As stated in Section 3.1.1 Algorithm FIND_SUPPORT_SET (A_i^r) ensures the following

$$\forall t \in w_i^r, \quad P(t/A_i^r) > P(t/A_i^{\sim r}) \tag{A.5}$$

$$\forall t \in \sim w_i^r, \quad P(t/A_i^r) < P(t/A_i^{\sim r}) \tag{A.6}$$

Now there are two possible relations between w and w_i^r. We deal with them separately to prove that $(P_i^r(w) + P_i^{\sim r}(\sim w)) < P_i^r(w_i^r) + P_i^{\sim r}(\sim w_i^r)$ is always true.

Case I: $w \cap w_i^r = \Phi$ i.e. w and w_i^r are disjoint subsets of J.

In this case, the following relations hold good.

$$w \subset \sim w_i^r \tag{A.7}$$

$$\sim w = (w_i^r) \cup (\sim w_i^r - w) \tag{A.8}$$

$$\sim w_i^r = w \cup (\sim w_i^r - w) \tag{A.9}$$

Therefore,

$$\begin{aligned}
&(P_i^r(w) + P_i^{\sim r}(\sim w))\\
&\quad = \sum_{t\in w} P(t \mid A_i^r) + \sum_{t\in(\sim w)} P(t \mid A_i^{\sim r}) \text{ by (A.2)}\\
&\quad = \sum_{t\in w} P(t \mid A_i^r) + \sum_{t\in(w_i^r)} P(t \mid A_i^{\sim r})\\
&\qquad + \sum_{t\in(\sim w_i^r - w)} P(t \mid A_i^{\sim r}) \text{ by (A.7)}
\end{aligned}$$

Since $w \subset \sim w_i^r$, therefore

$$\sum_{t\in w} P(t \mid A_i^r) < \sum_{t\in w} P(t \mid A_i^{\sim r}) \text{ by (A.6).}$$

Hence,

$$\begin{aligned}
&(P_i^r(w) + P_i^{\sim r}(\sim w))\\
&\quad < \sum_{t\in w} P(t|A_i^{\sim r}) + \sum_{t\in w_i^r} P(t|A_i^{\sim r}) + \sum_{t\in(\sim w_i^r - w)} P(t|A_i^{\sim r})\\
&\quad = \sum_{t\in w} P(t|A_i^{\sim r}) + \sum_{t\in(\sim w_i^r - w)} P(t|A_i^{\sim r}) + \sum_{t\in w_i^r} P(t|A_i^{\sim r})\\
&\quad = \sum_{t\in\sim w_i^r} P(t|A_i^{\sim r}) + \sum_{t\in w_i^r} P(t|A_i^{\sim r}) \text{ by clubbing}\\
&\qquad \text{the first two terms and using (A.9),}\\
&\quad < \sum_{t\in\sim w_i^r} P(t|A_i^{\sim r}) + \sum_{t\in w_i^r} P(t|A_i^r) \text{ by (A.5)}\\
&\quad = P_i^{\sim r}(\sim w_i^r) + P_i^r(w_i^r) \text{ by (A.2) and (A.4)}
\end{aligned}$$

Hence proved.

Case II: $w \cap w_i^r \neq \Phi$.

In this case, we will be using the following obvious set-theoretic relations

$$w = (w - w_i^r) \cup (w \cap w_i^r) \tag{A.10}$$

$$w_i^r = (w_i^r - w) \cup (w \cap w_i^r) \tag{A.11}$$

$$w = (w_i^{\sim r} - w) \cup (\sim w_i^r - w) \tag{A.12}$$

$$\sim w_i^r = (w - w_i^r) \cup (\sim w_i^r - w) \tag{A.13}$$

Therefore,

$$\begin{aligned}
&P_i^r(w) + P_i^{\sim r}(\sim w)\\
&\quad = \sum_{t\in w} P(t|A_i^r) + \sum_{t\in(\sim w)} P(t|A_i^{\sim r}) \text{ by (A.2)}\\
&\quad = \sum_{t\in(w-w_i^r)} P(t|A_i^r) + \sum_{t\in(w\cap w_i^r)} P(t|A_i^r)\\
&\qquad + \sum_{t\in(w_i^r-w)} P(t|A_i^{\sim r}) + \sum_{t\in(\sim w_i^r-w)} P(t|A_i^{\sim r})\\
&\qquad \text{using (A.10) and (A.12)}\\
&\quad < \sum_{t\in(w-w_i^r)} P(t|A_i^r) + \sum_{t\in(w\cap w_i^r)} P(t|A_i^r)\\
&\qquad + \sum_{t\in(w_i^r-w)} P(t|A_i^r) + \sum_{t\in(\sim w_i^r-w)} P(t|A_i^{\sim r})\\
&\qquad \text{by using (A.5) for the third term}\\
&\quad = \sum_{t\in(W-w_i^r)} P(t|A_i^r) + \sum_{t\in w_i^r} P(t|A_i^r) + \sum_{t\in(\sim w_i^r-W)} P(t|A_i^{\sim r})\\
&\qquad \text{by using (A.11)}\\
&\quad < \sum_{t\in(W-w_i^r)} P(t|A_i^{\sim r}) + \sum_{t\in w_i^r} P(t|A_i^r) + \sum_{t\in(\sim w_i^r-w)} P(t|A_i^{\sim r})\\
&\qquad \text{by using (A.6) for the first term,}\\
&\quad = \sum_{t\in\sim w_i^r} P(t|A_i^{\sim r}) + \sum_{t\in w_i^r} P(t|A_i^r) \text{ by using (A.13)}\\
&\quad = P_i^{\sim r}(\sim w_i^r) + P_i^r(w_i^r) \text{ by using (A.2)and (A.4)}
\end{aligned}$$

Thus, it is proved that $P_i^r(w) + P_i^{\sim r}(\sim w) < P_i^{\sim r}(\sim w_i^r) + P_i^r(w_i^r)$ for all $w \neq w_i^r$. □

References

Blum, L.A., Langley, P., 1997. Selection of relevant features and examples in machine learning. Artificial Intell. 97, 245–271.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. Wadsworth, Belmont, CA.

Casillas, J., Cordon, O., Del Jesus, M.J., Herrera, F., 2001. Genetic feature selection in a fuzzy rule-based classification system learning process for high-dimensional problems. Informance Sci. 136 (August), 135–157.

Ching, J.Y., Wong, A.K.C., Chan, K.C.C., 1995. Class-dependent discretization for inductive learning from continuous and mixed mode data. IEEE Trans. Pattern Anal. Machine Intell. 17 (7), 641–651.

Cost, S., Salzberg, S., 1993. In: A Weighted Nearest Algorithm with Symbolic FeaturesMachine Learning, vol. 10. Kluwer Publishers, Boston, MA, pp. 57–78.

Dash, M., Liu, H., 1997. Feature selection for classification. Intelligent Data Analysis, vol. 1, pp. 131–156.

Domingos, P., 1999. MetaCost—A general method to make classifier cost sensitive. In: Proc. Fifth Internat. Conf. Knowledge Discovery and Data Mining. ACM Press, San Diego, CA, pp. 155–164.

Dougherty, J., Kohavi, R., Sahami, M., 1995. Supervised and unsupervised discretization of continuous features. In: Proc. 12th Internat. Conf. on Machine Learning. Morgan Kaufmann, Tahoe City, CA.

Duda, R.O., Hart, P.E., Stork, D.G., 2000. Pattern Classification, second ed. John Wiley and Sons Inc.

Ho, S.Y., Liu, C.C., Liu, S., 2002. Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm. Pattern Recognition Lett. 23 (13), 1495–1503.

John, G.H., Kohavi, R., Pfleger, K., 1994. Irrelevant features and the subset selection problem. In: Proc. 11th Internat. Conf. on Machine Learning. Morgan Kaufmann, New Brunswick, NJ, pp. 121–129.

Kira, K., Rendell, L., 1992. A practical approach to feature selection. In: Proc. Ninth Internat. Conf. on Machine Learning. Aberdeen, Scotland, pp. 249–256.

Kononenko, I., 1994. Estimating attributes: Analysis and extensions of RELIEF. In: Proc. of Eur. Conf. on Machine Learning.

Kuncheva, L.I., Bezdek, J.C., 1998. Nearest prototype classification: Clustering, genetic algorithms or random search. IEEE Trans. Systems Man Cybernet. C 28 (1), 160–164.

Michalski, R.S., 1980. Pattern recognition as rule-guided inductive learning. IEEE Trans. Pattern Anal. Machine Intell. 2 (4), 349–361.

Narendra, P.M., Fukunaga, K., 1977. A branch and bound algorithm for feature subset selection. IEEE Trans. Comput. c-26 (9), 917–922.

Pudil, P., Novovicová, J., Kittler, J., 1994. Floating search methods in feature selection. Pattern Recognition Lett. 15 (11), 1119–1125.

Quinlan, J.R., 1986. Induction of decision trees. Machine Learning 1, 81–106.

Quinlan, J.R., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco.

Somol, P., Pudil, P., Ferri, F., Kittler, J., 2000. Fast branch and bound algorithm in feature selection. In: Sanchez, B., Pineda, J., Wolfmann, J., Bellahsense, Z., Ferri, F. (Eds.), World Multiconference on Systemics, Cybernetics and Informatics (SCI), Proceedings of SCI/ISAS 2000, vol. VII, pp. 646–651.

Stanfill, C., Waltz, D., 1986. Towards memory based reasoning. Comm. ACM 29 (12), 1213–1228.

Thawonmas, R., Abe, S., 1997. A novel approach to feature selection based on analysis of class regions. IEEE Trans Systems Man Cybernet. 27 (2), 196–207.

Witten, I.H., Frank, E., 2000. Data Mining. Morgan Kaufmann Publishers, San Francisco, California.

Wu, X., 1999. Induction by attribute elimination. IEEE Trans. Knowledge Data Eng. 11 (5), 805–812.

Xiong, N., 2002. A hybrid approach to input selection for complex processes. IEEE Trans. Systems Man Cybernet. Part A 32 (4), 532–536.