Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 156 (2004) 483–494
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

Computing, Artificial Intelligence and Information Technology

Evaluating feature selection methods for learning in data mining applications

Selwyn Piramuthu *

Decision and Information Sciences, University of Florida, 351 Stuzin Hall, P.O. Box 117169, Gainesville, FL 32611-7169, USA

Received 31 May 2001; accepted 18 November 2002

Abstract

Recent advances in computing technology in terms of speed, cost, as well as access to tremendous amounts of computing power and the ability to process huge amounts of data in reasonable time has spurred increased interest in data mining applications to extract useful knowledge from data. Machine learning has been one of the methods used in most of these data mining applications. It is widely acknowledged that about 80% of the resources in a majority of data mining applications are spent on cleaning and preprocessing the data. However, there have been relatively few studies on preprocessing data used as input in these data mining systems. In this study, we evaluate several inter-class as well as probabilistic distance-based feature selection methods as to their effectiveness in preprocessing input data for inducing decision trees. We use real-world data to evaluate these feature selection methods. Results from this study show that inter-class distance measures result in better performance compared to probabilistic measures, in general.

Keywords: Artificial intelligence; Feature selection; Decision trees; Credit risk analysis

1. Introduction

Data mining is the process of finding patterns and relations in large databases (Kerber et al., 1995). Data mining is especially advantageous in high-volume, frequently changing data such as in financial application areas (Whitebread and Jameson, 1995). The primary purpose of data mining is to extract information from huge amounts of raw data (Krivda, 1995). Data mining using statistical methods as well as machine learning methods such as induced decision trees, neural networks, among others, have been used for this purpose with good results (O'Reilly, 1995; Seshadri et al., 1995).

It is widely recognized that around 80% of the resources in data mining applications are spent on cleaning and preprocessing the data. The actual mining or extraction of patterns from the data requires the data to be clean since input data are the primary, if not the only, source of knowledge in these systems. Cleaning and preprocessing data involves several steps including procedures for handling incomplete, noisy, or missing data; sampling of appropriate data; feature selection; feature construction; and also formatting the data as per

* Tel.: +1-352-392-8882; fax: +1-352-392-5438.
E-mail address: selwyn@ufl.edu (S. Piramuthu).

0377-2217/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0377-2217(02)00911-6

the representational requirements of methods (e.g., decision trees, neural networks) used to extract knowledge from these data.

Invariably and rather unknowingly, along with relevant variables, irrelevant as well as redundant variables are included in the data, to better represent the domain in these applications. A relevant variable is neither irrelevant nor redundant to the target concept of interest (John et al., 1994). Whereas an irrelevant feature does not affect describing the target concept in any way, a redundant feature does not add anything new to describing the target concept. Redundant features might possibly add more noise than useful information in describing the concept of interest.

Feature selection is the problem of choosing a small subset of features that ideally is necessary and sufficient to describe the target concept (Kira and Rendell, 1992). Feature selection is of paramount importance for any learning algorithm which when poorly done (i.e., a poor set of features is selected) may lead to problems associated with incomplete information, noisy or irrelevant features, not the best set/mix of features, among others. The learning algorithm used is slowed down unnecessarily due to higher dimensions of the feature space, while also experiencing lower prediction accuracies due to learning irrelevant information. The ultimate objective of feature selection is to obtain a feature space with (1) low dimensionality, (2) retention of sufficient information, (3) enhancement of separability in feature space for example in different categories by removing effects due to noisy features, and (4) comparability of features among examples in same category (Meisel, 1972).

Although seemingly trivial, the importance of feature selection cannot be overstated. Consider for example a data mining situation where the concept to be learned is to classify good and bad creditworthy customers. The data for this application could possibly include several variables including social security number, asset, liability, past credit history, number of years with current employer, salary, and frequency of credit evaluation requests. Here, regardless of the variables included in the data, the social security number can uniquely determine a customer's creditworthiness. The learned knowledge consisting just the social security number as predictor will, of course, have extremely poor generalizability when applied to new customers. Clearly, in this case, to avoid such a problem we can exclude social security numbers from the input data. Since it is not always clear-cut as to which of the variables could result in such spurious patterns. A similar problem could possibly exist among one or more other variables in the data. Feature selection methods can be used in similar situations to cull out such problematic features before the data enters the pattern extraction stage in data mining systems.

The use of appropriate input data can result in improvements in performance, with minor effort. This study explores this idea of effectively utilizing input data. Several studies have shown that selecting and appropriately transforming features influence learning performance of feed-forward neural networks significantly (e.g., Battiti, 1994; Malki and Moghaddamjoo, 1991; Piramuthu and Shaw, 1994). The current pattern recognition literature attempts feature selection through varied means, such as statistical (e.g., Kerber et al., 1995), geometrical (e.g., Elomaa and Ukkonen, 1994), information-theoretic measures (e.g., Battiti, 1994), mathematical programming (e.g., Bradley et al., 1998), among others. We evaluate several distance-based feature selection methods in this study, as to their effectiveness on preprocessing input data for inducing decision trees. The classification performance of the resulting decision trees are used as the performance measure in this study.

This paper is organized as follows: Section 2 provides an overview of recent developments in feature selection methods. Experimental results using several feature selection methods with five real-world data sets are given in Section 3 and Section 4 conclude the paper with a brief discussion of lessons learned from this study.

2. Recent developments in feature selection

Feature selection is the problem of choosing a small subset of features that ideally is necessary and sufficient to describe the target concept (Kira and Rendell, 1992). The terms *features*, *variables*,

measurements, and *attributes* are used interchangeably in the literature. Selecting the appropriate set of features is extremely important since the feature set selected is the only source of information for any learning algorithm using the data of interest.

A goal of feature selection is to avoid selecting too many or too few features than is necessary. If too few features are selected, there is a good chance that the information content in this set of features is low. On the other hand, if too many (irrelevant) features are selected, the effects due to noise present in (most real-world) data may overshadow the information present. Hence, this is a tradeoff which must be addressed by any feature selection method.

The marginal benefit resulting from the presence of a feature in a given set plays an important role. A given feature might provide more information when present with certain other feature(s) than when considered by itself. Cover (1974), Elashoff et al. (1967), and Toussaint (1971), among others, have shown the importance of selecting features as a set, rather than selecting the best features to form the (supposedly) best set. They have shown that the best individual features do not necessarily constitute the best set of features.

However, in most real-world situations, it is not known what the best set of features is nor the number (n) of features in such a set. Currently, there is no means to obtain the value of n, which depends partially on the objective of interest. Even assuming that n is known, it is extremely difficult to obtain the best set of n features since not all n of these features may be present in the data comprising the available set of features.

It should be noted that feature selection is not appropriate for certain classes of data sets. Clear examples are those of parity problems, an example of which is the exclusive-OR (XOR) problem. This is because all the attributes are necessary to determine the category (here, odd or even parity). Deletion of even one attribute would result in half the cases being categorized incorrectly.

There exists a vast amount of literature on feature selection, including books that specifically cover the topic (e.g., Liu and Motoda, 1998a; Liu and Motoda, 1998b). Researchers have attempted feature selection through varied means, such as statistical (e.g., Kittler, 1975), geometrical (e.g., Elomaa and Ukkonen, 1994), information-theoretic measures (e.g., Battiti, 1994), mathematical programming (e.g., Bradley et al., 1998), among others.

In statistical analyses, forward and backward stepwise multiple regression (SMR) are widely used to select features, with forward SMR being used more often due to the lesser magnitude of calculations involved. The output here is the smallest subset of features resulting in an R^2 (correlation coefficient) value that explains a significantly large amount of the variance. In forward SMR, the analyses proceeds by adding features to a subset until the addition of a new feature no longer results in a significant (usually at the 0.05 level) increment in explained variance (R^2 value). In backward SMR, the full set of features are used to start with, while seeking to eliminate features with the smallest contribution to R^2.

Malki and Moghaddamjoo (1991) apply the K–L transform on the training examples to obtain the initial training vectors. Training is started in the direction of the major eigenvectors of the correlation matrix of the training examples. The remaining components are gradually included in their order of significance. The authors generated training examples from a synthetic noisy image and compared the results obtained using the proposed method to those of standard backpropagation algorithm. The proposed method converged faster than standard backpropagation with comparable classification performance.

Siedlecki and Sklansky (1989) use genetic algorithms for feature selection by encoding the initial set of n features as n-element bit string with 1 and 0 representing the presence and absence respectively of features in the set. They used classification accuracy, as the fitness function (for genetic algorithms while selecting features) and obtained good neural network results compared to branch and bound and sequential search (Stearns, 1976) algorithms. They used a synthetic data as well as digitized infrared imagery of real scences, with classification accuracy as the objective function. Yang and Honavar (1997) report a similar

study. However, later Hopkins et al. (1994) shows that classification accuracy may be a poor fitness function measure when searching for reducing the dimension of the feature set.

Using rough sets theory (Pawlak, 1982), PRESET (Modrzejewski, 1993) determines the degree of dependency (γ) of sets of attributes for selecting binary features. Features leading to a minimal preset decision tree, which is the one with minimal length of all path from root to leaves, are selected. Kohavi and Frasca (1994) use best-first search, stopping after a predetermined number of non-improving node expansions. They suggest that it may be beneficial to use a feature subset that is not a reduct, which has a property that a feature cannot be removed from it without changing the independence property of features. A table-majority inducer was used with good results.

The *wrapper* method (Kohavi, 1995) searches for a good feature subset using the induction algorithm as a black box. The feature selection algorithm exists as a wrapper around the induction algorithm. The induction algorithm is run on data sets with subsets of features, and the subset of feature with the highest estimated value of a performance criterion is chosen. The induction algorithm is used to evaluate the data set with the chosen features, on an independent test set.

Almuallim and Dietterich (1991) introduce MIN-FEATURES (if two functions are consistent with the training examples, prefer the function that involves fewer input features) bias to select features in the FOCUS algorithm. They used synthetic data to study the performance of the FOCUS, ID3, and FRINGE algorithms using sample complexity, coverage, and classification accuracy as performance criteria. They increased the number of irrelevant features and showed that FOCUS performed consistently better.

The IDG algorithm (Elomaa and Ukkonen, 1994) takes the positions of examples in the instance space to select features for decision trees. They limit their attention to boundaries separating examples belonging to different classes, while rewarding (penalizing) rules that separate examples from different (same) classes. Eight data sets are used to compare the performance (% accuracy, number of nodes in decision tree, time) of decision trees constructed using the proposed algorithm with ID3 (Quinlan, 1987). Decision trees generated using the proposed algorithm had better accuracy whereas those with ID3 had fewer number of nodes and took more than an order of magnitude less time.

Based on the positions of instances in instance space, the Relief algorithm (Kira and Rendell, 1992) selects features that are statistically relevant to target concept, using a relevancy threshold that is selected by the user. Relief is noise-tolerant and is unaffected by feature interaction. The complexity of relief is $O(pn)$, where n and p are the number of instances and number of features respectively. Relief was studied using two 2-class problems with good results, compared to FOCUS (Almuallim and Dietterich, 1991) and heuristic search (Devijver and Kittler, 1982). Kononenko (1994) extended RELIEF to deal with noisy, incomplete, and multi-class data sets.

Milne (1995) used neural networks to measure the contribution of individual input features to the output of the neural network. A new measure of input features' contribution to output is proposed, and evaluated using data mapping species occurrence in a forest. Using a scatter plot of contribution to output, subsets of features were removed and the remaining feature sets were used as input to neural networks. Setino and Liu (1997) present a similar study using neural networks to select features.

Battiti (1994) developed MIFS to use mutual information for evaluating the information content of each individual feature with respect to the output class. The features thus selected were used as input in neural networks. The author shows that the proposed method is better than those feature selection methods that use linear dependence (e.g., correlations as in principal components analysis) measures. Koller and Sahami (1996) use cross-entropy to minimize the amount of predictive information lost during feature selection. Piramuthu and Shaw (1994) use C4.5 (Quinlan, 1990) to select features used as input in neural networks. Their results showed improvements, over just backpropagation, both in terms of classification accuracy and time taken by neural networks to converge.

The most popular feature selection methods in machine learning literature are variations of sequential forward search (SFS) and sequential backward search (SBS) as described in Devijver and Kittler (1982) and its variants (e.g., Pudil et al., 1994; Quinlan, 1987). SFS (SBS) obtains a chain of nested subsets of features by adding (subtracting) the locally best (worst) feature in the set. These methods are particular cases of the more general 'plus l—take away r' method (Stearns, 1976). Results from previous studies indicate that the performance using forward and backward searches are comparable. In terms of computing resources, forward search has the advantage since fewer number of features are evaluated at each iteration, compared to backward search where the process begins using all the features.

3. Experimental results

In this study, we evaluate SFS with several different distance measures. Both inter-class distance as well as probabilistic distance measures are used. Specifically, the probabilistic distance measures used are the Bhattacharyya measure, the Matusita measure, the divergence measure, the Mahalanobis distance measure, and the Patrick-Fisher measure. The inter-class distance measures used are the Minkowski distance measure, city block distance measure, Euclidean distance measure, the Chebychev distance measure, and the nonlinear (Parzen and hyperspheric kernel) distance measure.

Inter-class distance is taken as the selection criterion for all the inter-class distance-based feature selection methods, where both the between and within class distances are taken into account.

The Euclidean distance between examples of concepts is based on the idea that the greater the distance between the examples from different concepts the better the class separability. The between-class scatter matrix, B, is given as

$$B = \sum_{i=1}^{m} P(\omega_i) v_i v_i^T \tag{1}$$

and the averaged within class scatter matrix, W, is given as

$$W = \sum_{i=1}^{m} P(\omega_i) E\{(x - v_i)(x - v_i)^T\}. \tag{2}$$

The Euclidean distance is calculated from the ratio of the scatter matrices:

$$d = \frac{|W + B|}{|W|}. \tag{3}$$

For the nonlinear (Parzen and hyperspheric kernel) method, we use η (the parameter for the calculus of the radius) $= 0.5$.

The class conditional probability density of pattern x with *a priori* probability of occurrence of normally distributed classes ω_i, $(i = 1, 2)$ is given by

$$p(x|\omega_i) = \frac{1}{\sqrt{((2\pi)^n |\Sigma_i|)}} \times \exp\left[-\frac{1}{2}(x - v_i)^T \Sigma_i^{-1} (x - v_i) \right] \tag{4}$$

where v_i and Σ_i are the mean vector and the covariance matrix of the distribution respectively.

Consider the following matrix with a set of five examples measuring three variables:

$$X = \begin{bmatrix} 4.0 & 2.0 & 0.60 \\ 4.2 & 2.1 & 0.59 \\ 3.9 & 2.0 & 0.58 \\ 4.3 & 2.1 & 0.62 \\ 4.1 & 2.2 & 0.63 \end{bmatrix}.$$

The mean vector for this set of examples, containing the arithmetic averages of the three variables, is

$$v_i = [4.10 \quad 2.08 \quad 0.604]$$

and the covariance matrix, containing the variances of the variables along the main diagonal and the covariances between each pair of variables in the rest of the positions, is

$$\Sigma_i = \begin{bmatrix} 0.025 & 0.0075 & 0.00175 \\ 0.0075 & 0.0070 & 0.00135 \\ 0.00175 & 0.00135 & 0.00043 \end{bmatrix}.$$

Some of the probabilistic distance measures can be expressed in terms of distribution parameters (Fukunaga, 1972).

The Bhattacharyya distance (J_B) is represented as

$$J_B(\omega_1,\omega_2)=\frac{1}{8}(v_2-v_1)^T\left\{\frac{\Sigma_1+\Sigma_2}{2}\right\}^{-1}\times(v_2-v_1)+\frac{1}{2}\ln\frac{\left(\frac{1}{2}|\Sigma_1+\Sigma_2|\right)}{2\sqrt{|\Sigma_1\Sigma_2|}}. \quad (5)$$

Similarly, the divergence distance (J_D) is represented as

$$J_D(\omega_1,\omega_2)=\frac{1}{2}(v_2-v_1)^T\left(\Sigma_1^{-1}+\Sigma_2^{-1}\right)(v_2-v_1)+\frac{1}{2}\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_2+\Sigma_2^{-1}\Sigma_1-2I\right) \quad (6)$$

and the Matusita distance (J_M) is represented as

$$J_M(\omega_1,\omega_2)=\sqrt{(2(1-\exp[-J_B(\omega_1,\omega_2)]))}. \quad (7)$$

When the covariance matrices Σ_1 and Σ_2 are identical, i.e., $\Sigma_1=\Sigma_2=\Sigma$, both the Bhattacharyya distance and the divergence distance simplify even further to become the Mahalanobis distance. The Mahalanobis distance (J_m) is represented as

$$J_m(\omega_1,\omega_2)=8J_B(\omega_1,\omega_2)=(v_2-v_1)^T\Sigma^{-1}(v_2-v_1). \quad (8)$$

We study the performance of these distance measures as feature selectors and as preprocessors for induced decision trees (C4.5, as described in Quinlan (1990)), using five real-world application data. These data sets have been used in previous studies. Since there is no hard and fast rule on the number of features to be used in any given application, using feature selection methods we select about half the number of features that are available in each of the data sets.

3.1. Credit approval data

The credit approval data was used in Quinlan (1987), among others. The data set was *cleaned* to remove examples with missing attribute values. This data is from a large bank. Each of the examples in this data corresponds to a credit card application, with nine discrete and six real attributes. The discrete attributes have anywhere from 2 through 14 possible values. This is a binary classification data, corresponding to positive and negative decisions. There were 690 examples in this data set. We removed the incomplete examples and ended up with 653 examples of which 296 belong to positive class and 357 belong to negative class, where the classes correspond to whether or not credit was approved. This data set is also known to be noisy. We select eight attributes from this data set using the various feature selection methods.

The results using credit approval data in induced decision trees with data preprocessed through the different feature selection methods are given in Table 1. We report here the results from before pruning (BP) as well as after pruning (AP) the induced decision trees, although the results corresponding to AP are of primary importance. Tenfold cross-validation was used—the data set was randomly split into 10 identical samples, and each of these samples were used as the testing (holdout) data set one after another when the other nine were used as the training data set for inducing decision trees. Hence, 10 decision trees were generated with 90% of the data being used to generate the decision trees and the rest for testing the performance of these trees generated. Feature selection was done using 90% (training sample) of the data.

The first column refers to the feature selection method used. Here, "Random" refers to the case where the same number of features (as the feature selection methods) were selected randomly. This was done as a benchmark to compare the feature selection methods using the same number of features. "None" refers to the case where the entire set of features in the data set. That is, no feature selection was done in the "None" case.

The classification accuracies for both the training as well as the testing (holdout) examples are given as percentage correctly classified. Each of the entries in the table are the average of 10 runs (tenfold cross-validation), and the corresponding standard deviation values are given in parentheses.

As can be seen from Table 1, the tree size using different methods are comparable, with minor variations except for the "Random" case. The average tree size BP and AP did not differ appreciably across the different methods except in the "None" and "Random" cases. The tree sizes in the

Table 1
Results using credit approval data set

Distance measure	Tree size		Classification accuracy			
	BP	AP	Training examples		Testing examples	
			BP (%)	AP (%)	BP (%)	AP (%)
Minkowski	11.4 (1.74)	11.4 (1.74)	97.68 (1.04)	97.68 (1.04)	77 (15.5)	77 (15.5)
City Block	11.4 (1.74)	11.4 (1.74)	97.68 (1.04)	97.68 (1.04)	77 (15.5)	77 (15.5)
Euclidean	10.8 (1.66)	9.4 (2.15)	97.91 (1.25)	97.23 (2.45)	71.5 (15.8)	69 (17)
Chebychev	13 (2.19)	12.2 (2.56)	95.85 (2.68)	95.38 (2.92)	60 (14.8)	58 (18.3)
Nonlinear	10.8 (1.4)	10.2 (1.6)	97.92 (0.7)	97.45 (1.63)	80 (20)	82 (18.9)
Bhattacharyya	14.2 (1.83)	14.2 (1.83)	95.61 (1.91)	95.61 (1.91)	59 (14.3)	59 (14.3)
Matusita	14.2 (1.83)	14.2 (1.83)	95.61 (1.91)	95.61 (1.91)	59 (14.3)	59 (14.3)
Divergence	12.6 (2.65)	11.4 (1.96)	96.76 (1.54)	96.3 (1.86)	71.5 (18.2)	71.5 (18.2)
Mahalanobis	10.6 (1.5)	9.6 (1.8)	97.69 (1.45)	97.46 (1.9)	74 (18)	69 (17)
Patrick-Fisher	15.8 (2.86)	14.4 (2.84)	96.3 (1.86)	95.6 (2.43)	61 (12.8)	61 (9.17)
Random	91 (15.66)	30.9 (10.28)	91.73 (0.38)	89.44 (0.97)	64.37 (3.86)	60.39 (3.45)
None	91.8 (9.92)	55 (8.43)	96.76 (0.45)	95.04 (0.86)	81.62 (5.10)	84.23 (3.84)

latter two cases are significantly ($P \ll 0.001$ using two-tailed t-test) different than those in the cases where a feature selection method (other than random) was used. The classification results using training examples are at least as good as those using testing examples. This is expected since the data is noisy, and it is hard for any method to learn to generalize noisy examples. The classification results using the training examples are almost similar, based on the average as well as the standard deviation values for the training examples being small. Also, as expected, the percentage of training examples correctly classified decreased AP. This is because pruning helps remove some of the problems associated with over-fitting noise in the examples by removing some nodes near the leaf-level of the decision tree, thus resulting in reduced classification performance on training examples. The time taken by the feature selection methods was in the order of a few seconds in an IBM/sp2 machine.

Given that the tree sizes and classification accuracy on training examples are comparable across the different methods, let us now consider the last two columns in Table 1. Of these, the last column (classification accuracy on testing examples, using pruned decision trees) is of interest in practice since these are the results using the final decision trees on heretofore unseen examples. Here, some of the methods perform clearly better than some others.

The case where no feature selection was used resulted in the best classification performance [84.23 (3.84)] on testing examples. This could be because of the importance of more variables, than are used in the feature selection methods, for classification purposes. That is, we used only eight of the attributes in the cases using feature selection. It is unlikely that every attribute in the original data set is necessary for the best classification result like in the XOR problem. However, it is likely that at least nine of the variables in the data set are deemed important for improved classification of the examples. The method based on nonlinear distance measure resulted in the second best performance [82 (18.9)]. The differences in classification performence on testing examples between the "None" and nonlinear cases are not statistically significant (using two-tailed t-test). Similarly the difference between the classification performance on holdout sample using random feature selection and the worst performer (Chebychev) is not statistically significant.

Feature selection using the Chebychev measure resulted in the worst performance [58 (18.3)]. The difference between the classification performance on holdout sample using random feature selection and the worst performer (Chebychev) is not statistically significant. Some (e.g., the Minkowski and city block; the Bhattacharyya and Matusita) of the distance measures resulted in similar performance throughout. It should be noted that the

tree size in the case where no feature selection was used is about five times that of the nonlinear case. For this data set, feature selection based on nonlinear distance measure is clearly the method of choice based on both tree size and classification performance on testing examples.

3.2. Loan default data

This data has been used in previous studies (e.g., Abdel-Khalik and El-Sheshai, 1980) to classify a set of firms into those that would default and those that would not default on loan payments. The source of this data is the Index of Corporate Events in the 1973–1975 issues of Disclosure Journal. Sixteen defaulted firms were matched with 16 nondefaulted firms to obtain data for the study. Another set of sixteen examples, all belonging to the nondefault case, were also used. The second data set (of sixteen examples) was used by Abdel-Khalik and El-Sheshai (1980) as the holdout data set. Since we are using tenfold cross-validation in this study, we combined the two data sets together, resulting in a data set with 48 examples from which subsets were formed randomly.

There are 18 variables in this data: (1) net income/total assets, (2) net income/sales, (3) total debt/total assets, (4) cash flow/total debt, (5) long-term debt/net worth, (6) current assets/current liabilities, (7) quick assets/sales, (8) quick assets/current liabilities, (9) working capital/sales, (10) cash at year-end/total debt, (11) earnings trend, (12) sales trend, (13) current ratio trend, (14) trend of LTD/NW, (15) trend of WC/sales, (16) trend of NI/TA, (17) trend of NI/sales, and (18) trend of cash flow/TD. For detailed description of this data, the reader is referred to Abdel-Khalik and El-Sheshai (1980). Using the feature selection methods, we select 10 of the attributes for evaluation.

Table 2 provides results from decision trees generated after preprocessing input through the feature selection methods. The classification performance on training examples (both BP and AP the decision trees) are comparable. Unlike the credit approval data set case, here the average tree size both BP and AP did differ appreciably across the various methods. For example, both BP and AP, among feature selection cases the Chebychev measure resulted in the smallest tree and the Patrick-Fisher measure resulted in the largest tree. As before, there are similar results for both Minkowski and city block as well as Bhattacharyya and Matusita measures.

The classification accuracy on testing examples, AP, does show variations although not as pronounced as in the credit approval data case. Here, the best result [86.5 (4)] corresponds to the Bhattacharyya as well as the Matusita measures. This is followed quite closely by all the other measures

Table 2
Results using loan default data set

Distance measure	Tree size		Classification accuracy			
	BP	AP	Training examples		Testing examples	
			BP (%)	AP (%)	BP (%)	AP (%)
Minkowski	118.4 (8.9)	39 (7.32)	95.93 (0.33)	92.48 (0.76)	81.9 (3.03)	85.1 (4.52)
City Block	118.4 (8.9)	39 (7.32)	95.93 (0.33)	92.48 (0.76)	81.9 (3.03)	85.1 (4.52)
Euclidean	97.8 (4.21)	45.8 (8.54)	95.47 (0.57)	93.25 (1.04)	84.5 (3.34)	86.1 (4.36)
Chebychev	75 (3.9)	25.6 (6.45)	92 (0.63)	90.01 (0.9)	85.8 (3.66)	86.2 (3.98)
Nonlinear	117.8 (10.17)	32.2 (6.71)	95.13 (0.31)	91.73 (0.63)	82.8 (4.38)	86.2 (4.67)
Bhattacharyya	99 (9.51)	44.4 (6.26)	95.3 (0.46)	93.06 (0.44)	85 (3.65)	86.5 (4)
Matusita	99 (9.51)	44.4 (6.26)	95.3 (0.46)	93.06 (0.44)	85 (3.65)	86.5 (4)
Divergence	102.6 (4.88)	35.2 (9.57)	95.37 (0.5)	91.66 (0.95)	82.5 (5.32)	85.3 (4.3)
Mahalanobis	97.8 (4.21)	45.8 (8.54)	95.47 (0.57)	93.25 (1.04)	84.5 (3.34)	86.1 (4.36)
Patrick-Fisher	172.8 (10.75)	91 (12.13)	95.54 (0.65)	90.44 (1.53)	70.3 (5.77)	74.3 (2.72)
Random	87 (1.54)	51.4 (1.02)	90.3 (1.24)	86.3 (1.13)	70.1 (7.65)	72.2 (8.82)
None	9.6 (1.28)	9.4 (1.49)	97.45 (1.25)	97.45 (1.25)	68.5 (14.15)	68.5 (14.15)

except for Patrick-Fisher measure which trails behind the others at [74.3 (2.72)]. The results using Chebychev measure are right behind the best results, unlike in the credit approval data case. Moreover, the results using the nonlinear measure are quite good, just behind those using Chebychev measure, based on classification results on testing examples. The classification performance on testing examples of the "Random" case is statistically significantly different compared to the worst performer (Patrick-Fisher) in this case ($p < 0.5$ using two-tailed t-test).

3.3. Web traffic data

The data used were collected from Silicon Investor's Web site (www.techstocks.com). Specifically, the data comprises data-log of chat sites from six different companies. These include Apple Computers, Compaq Computers, Hasbro, Network Appliance, Seagate and Western Digital. Each had a chat room with varying number of messages. Data were collected from the beginning of July 1998 through early July 1999. The number of messages varied widely, from 130 for Network Appliance to 35,000 for Compaq. The data on each company's stock were also obtained. This include the open, close, high and low price for each company from the beginning of July 1998 through early July 1999. The data on the closing value of four indices of stocks are from their Web sites. The New York Stock Exchange (NYSE) Composite index are from their Web site (www.nyse.com) while the data on the NASDAQ Composite, NASDAQ 100 and NASDAQ Computer indices are from their Web site (www.nasdaq-amex.com). These indices were chosen because four of the stocks are listed on the NYSE while the other two are on NASDAQ.

The independent variables used are daily change in stock price, direction of daily change in stock price, range of movement of daily stock price, change in NYSE Composite Index value, direction of change in NYSE Composite Index value, change in NASDAQ Composite Index value, direction of change in NASDAQ Composite Index value, change in NASDAQ 100 Index value, direction of change in NASDAQ 100 Index value, change in NASDAQ Computer Index value, direction of change in NASDAQ Computer Index value, weekend binary, and daily trading volume. The weekend binary takes on values 0 or 1 depending on whether the day is a day when the stock markets are closed or open. The direction and magnitude variables are discretized. The direction variables take values up, no change, or down. The change variables take values low, medium, and high. We select five features using the feature selection methods.

Table 3 provides results from decision trees generated after preprocessing input through the feature selection methods. The classification performance on training examples (both BP and AP the decision trees) are comparable for all the cases where feature selection methods were used except Patrick-Fisher. Here, the classification performance on testing examples for Patrick-Fisher and "Random" cases are statistically significant ($p < 0.005$ using two-tailed t-test). The differences in tre sizes for these two cases are also statistically significant ($p \ll 0.001$ using two-tailed t-test). However, the classification performance on testing examples for Chebychev and "None" cases are not statistically significant.

3.4. Tam and Kiang (1992) data

This data set was used in the Tam and Kiang (1992) study. Texas banks that failed during 1985–1987 were the primary source of data. Data from a year and two years prior to their failure were used. Data from 59 failed banks were matched with 59 nonfailed banks, which were comparable in terms of asset size, number of branches, age and charter status. Tam and Kiang had also used holdout samples for both the one- and two-year prior cases. The one-year prior case consists of 44 banks, 22 of which belongs to failed and the other 22 to nonfailed banks. The two-year prior case consists of 40 banks, 20 of which belongs to failed and 20 to nonfailed banks. The data describes each of these banks in terms of 19 financial ratios. For a detailed overview of the data set, the reader is referred to Tam and Kiang (1992). For both the data sets, we select 10 of the features using the feature selection methods.

Table 3
Results using web traffic data

Distance measure	Tree size		Classification accuracy			
	BP	AP	Training examples		Testing examples	
			BP (%)	AP (%)	BP (%)	AP (%)
Minkowski	104.6 (25.6)	54.2 (18.7)	82.43 (0.6)	81.38 (0.7)	78.21 (1.4)	78.35 (1.1)
City Block	102.6 (23.9)	52.6 (15.8)	82.16 (0.7)	81.19 (0.6)	77.79 (2.5)	78.79 (1.9)
Euclidean	122.2 (17.8)	34 (7.4)	82.15 (0.4)	80.74 (0.4)	76.97 (2.2)	78.36 (2)
Chebychev	102.6 (23.9)	52.6 (15.8)	82.16 (0.7)	81.19 (0.6)	77.79 (2.5)	78.79 (1.9)
Nonlinear	87.6 (27.4)	36.4 (7.7)	81.56 (0.6)	80.71 (0.5)	76.47 (2.1)	77.12 (2)
Bhattacharyya	74.8 (16.8)	26 (8.7)	81.29 (0.3)	80.34 (0.3)	78.12 (1.7)	78.6 (1.2)
Matusita	74.8 (16.8)	26 (8.7)	81.29 (0.3)	80.34 (0.3)	78.12 (1.7)	78.6 (1.2)
Divergence	74.8 (16.8)	26 (8.7)	81.29 (0.3)	80.34 (0.3)	78.12 (1.7)	78.6 (1.2)
Mahalanobis	94.2 (22.4)	29.6 (11.9)	81.73 (0.6)	80.41 (0.5)	76.73 (2.2)	78.45 (2.3)
Patrick-Fisher	29.8 (6.1)	4.4 (5.8)	73.36 (0.3)	72.53 (0.4)	72.13 (1.3)	71.78 (0.9)
Random	116.4 (17.4)	34.6 (13.5)	76.74 (0.7)	75.22 (0.8)	71.33 (1.7)	68.87 (1.6)
None	349.6 (27.24)	112 (17)	88.54 (0.5)	84.2 (0.5)	75.2 (2.9)	77.69 (2.5)

Table 4 provides results from decision trees generated after preprocessing input through the feature selection methods for the one-year prior case. The classification performance on training examples (both BP and AP the decision trees) are comparable for all the cases where feature selection methods were used. The classification performance on testing examples between the "Random" case and the worst performers using feature selection (Minkowski and City Block methods) is not statistically significant (using two-tailed t-test). However, the tree-sizes between these cases are statistically significantly different.

Table 5 provides results from decision trees generated after preprocessing input through the feature selection methods for the two-year prior case. The classification performance on training examples (both BP and AP the decision trees) are comparable for all the cases where feature selection methods were used except for the Patrick-Fisher case. The classification performance on testing examples between the "Random" case and the worst performer using feature selection (Patrick-Fisher) is not statistically significant (using two-tailed t-test). However, the tree-sizes between these cases are statistically significant.

Table 4
Results using one-year prior data set

Distance measure	Tree size		Classification accuracy			
	BP	AP	Training examples		Testing examples	
			BP (%)	AP (%)	BP (%)	AP (%)
Minkowski	39.8 (5.59)	24.8 (5.03)	94.64 (0.98)	91.91 (1.48)	73.05 (14.23)	75.56 (13.75)
City Block	39.8 (5.59)	24.8 (5.03)	94.64 (0.98)	91.91 (1.48)	73.05 (14.23)	75.56 (13.75)
Euclidean	33.6 (4.99)	17.6 (5.25)	94.38 (2.25)	91.36 (2.73)	74.81 (14.75)	80.99 (13.32)
Chebychev	37.2 (5.53)	23.8 (3.01)	94.85 (1.16)	92.53 (2.15)	79.71 (9.96)	80.93 (8.81)
Nonlinear	37.6 (4.22)	18 (5.35)	94.43 (0.61)	90.47 (2.15)	82.16 (5.08)	80.26 (6.36)
Bhattacharyya	30.6 (5.79)	16.2 (3.91)	93.36 (1)	90.67 (1.27)	83.96 (7.28)	80.23 (7.16)
Matusita	30.6 (5.79)	16.2 (3.91)	93.36 (1)	90.67 (1.27)	83.96 (7.28)	80.23 (7.16)
Divergence	27.2 (6.89)	14 (4.35)	92.6 (1.76)	89.57 (2.24)	78.93 (7.33)	78.29 (9.91)
Mahalanobis	33.6 (4.99)	17.6 (5.25)	94.38 (2.25)	91.36 (2.73)	74.81 (14.75)	80.99 (13.32)
Patrick-Fisher	26.6 (4.09)	11.8 (3.29)	91.44 (1.46)	88.47 (1.91)	79.56 (10.22)	79.63 (8.31)
Random	36.6 (8.15)	26.4 (3.66)	94.58 (0.88)	90.82 (1.81)	70.84 (7.45)	73.89 (8.46)
None	42.8 (2.90)	33.4 (6.38)	96.44 (0.89)	94.11 (2.16)	82.75 (8.09)	78.4 (9.90)

Table 5
Results using two-year prior data set

Distance measure	Tree size		Classification accuracy			
	BP	AP	Training examples		Testing examples	
			BP (%)	AP (%)	BP (%)	AP (%)
Minkowski	44.4 (5.3)	25.2 (6.2)	93.34 (1.2)	89.09 (3.1)	74.13 (12)	72.22 (10.3)
City Block	44.4 (5.3)	25.2 (6.2)	93.34 (1.2)	89.09 (3.1)	74.13 (12)	72.22 (10.3)
Euclidean	37.8 (8.2)	20.2 (3.3)	91.78 (1)	88.51 (1.7)	79.83 (10.4)	81.12 (9.7)
Chebychev	43.6 (6.6)	23.6 (6.9)	91.78 (1.9)	87.95 (2.6)	69.74 (11.9)	68.37 (14.7)
Nonlinear	43.6 (6.6)	23.6 (6.9)	91.78 (1.9)	87.95 (2.6)	69.74 (11.9)	68.37 (14.7)
Bhattacharyya	35.4 (6.9)	16.8 (7.1)	90.28 (1.9)	86.34 (3.3)	75.87 (8.5)	74 (10.1)
Matusita	35.4 (6.9)	16.8 (7.1)	90.28 (1.9)	86.34 (3.3)	75.87 (8.5)	74 (10.1)
Divergence	37 (9.4)	21.6 (6.9)	90.76 (1.8)	87.74 (2.8)	72.17 (10.5)	73.46 (8.2)
Mahalanobis	38.6 (2.8)	22.6 (3.2)	91.7 (1.5)	89.29 (1.9)	79.92 (14)	81.12 (11.3)
Patrick-Fisher	29 (3.8)	11.8 (6)	82.07 (2)	77.27 (3.3)	71.66 (16.9)	67.31 (17.4)
Random	39.6 (4.6)	32.7 (6.8)	91.04 (1.2)	85.92 (2.7)	75.84 (12.8)	66.8 (9.1)
None	41.8 (5.7)	26.4 (5.1)	94.96 (1)	91.41 (2)	73.29 (12.1)	73.38 (11.5)

4. Discussion

We studied the effects feature selection methods have on learning in induced decision trees. Using SFS algorithm, we tested several different distance measures as to their effectiveness for feature selection applications.

Although there is no one "best" distance measure for these applications, it is likely that some methods do perform better in general depending on the characteristics of the data. Given the results from this paper, based on the tree-size and classification accuracy on testing examples, the nonlinear measure is the one of choice in most cases.

The results also show that learning in induced decision trees is sensitive to the input data used. By selecting appropriate features through preprocessing, the performance of induced decision trees can be improved without much effort since most of these preprocessing techniques are not time/computing intensive. This is true for any learning algorithm, since the complexity of the data used directly affects the learning algorithm's performance. Feature selection, when used along with *any* learning system, can help improve performance of these systems even further with minimal additional effort.

By selecting useful features from the data set, we are essentially reducing the number of features needed for these decisions. This in turn translates to reduction in data gathering costs as well as storage and maintenance costs associated with features that are not necessarily useful for the decision problem of interest.

We studied the performance of induced decision trees with data preprocessed by several feature selection methods using five real-world data sets. These are important problems where the stakes are high. Any improvement over the methods currently being used translates to tremendous savings for the institutions involved, both in monetary terms as well as in terms of efficiently using the available raw data to extract useful information.

References

Abdel-Khalik, A.R., El-Sheshai, K.M., 1980. Information choice and utilization in an experiment on default prediction. Journal of Accounting Research, Autumn, 325–342.

Almuallim, H.M., Dietterich, T.G., 1991. Learning with many irrelevant features. In: Proceedings of the Ninth National Conference on Artificial Intelligence. pp. 547–552.

Battiti, R., 1994. Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on Neural Networks 5 (4), 537–550.

Bradley, P.S., Mangasarian, O.L., Street, W.N., 1998. Feature selection in mathematical programming. INFORMS Journal on Computing 10 (2), 209–217.

Cover, T.M., 1974. The best two independent measurements are not the two best. IEEE Transactions on Systems, Man, and Cybernetics (January), 116–117.

Devijver, P.A., Kittler, J., 1982. Pattern Recognition: A Statistical Approach. Prentice-Hall, Englewood Cliffs, NJ.

Elashoff, J.D., Elashoff, R.M., Goldman, G.E., 1967. On the choice of variables in classification problems with dichotomous variables. Biometrika 54, 668–670.

Elomaa, T., Ukkonen, E., 1994. A geometric approach to feature selection. In: Proceedings of the European Conference on Machine Learning. pp. 351–354.

Fukunaga, K., 1972. Introduction to Statistical Pattern Recognition. Academic Press, New York.

Hopkins, C., Routen, T., Watson, T., 1994. Problems with using genetic algorithms for neural network feature selection. 11th European Conference on Artificial Intelligence. pp. 221–225.

John, G.H., Kohavi, R., Pfleger, K., 1994. Irrelevant features and the subset selection problem. In: Proceedings of the Eleventh International Conference on Machine Learning. pp. 121–129.

Kerber, R., Livezy, B., Simoudis, E., 1995. In: Goonatilake, S., Khebbal, S. (Eds.), A Hybrid System for Data Mining. Intelligent Hybrid Systems. John Wiley, New York.

Kira, K., Rendell, L.A., 1992. A practical approach to feature selection. In: Proceedings of the Ninth International Conference on Machine Learning. pp. 249–256.

Kittler, J., 1975. Mathematical methods of feature selection in pattern recognition. International Journal of Man–Machine Studies 7, 609–637.

Kohavi, R., Frasca, B., 1994. Useful feature subsets and rough sets reducts. Third International Workshop on Rough Sets and Soft Computing (RSSC 94).

Kohavi, R., 1995. Wrappers for Performance Enhancement and Oblivious Decision Graphs, Ph.D. Dissertation, Computer Science Department, Stanford University.

Koller, D., Sahami, M., 1996. Toward optimal feature selection. In: Machine Learning: Proceedings of the Thirteenth International Conference.

Kononenko, I., 1994. Estimating attributes: Analysis and extensions of RELIEF. In: Proceedings of the European Conference on Machine Learning. pp. 171–182.

Krivda, C.D., 1995. Data-mining dynamite. BYTE (October), 97–102.

Liu, H., Motoda, H., 1998a. Feature Extraction, Construction, and Selection: A Data Mining Perspective. Kluwer Academic Publishers, Dordrecht.

Liu, H., Motoda, H., 1998b. Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers, Dordrecht.

Modrzejewski, M., 1993. Feature selection using rough sets theory, In: European Conference on Machine Learning. pp. 213–226.

Malki, H.A., Moghaddamjoo, A., 1991. Using the Karhunen–Loe've transformation in the back-propagation training algorithm. IEEE Transactions on Neural Networks 2 (1), 162–165.

Meisel, W.S., 1972. Computer-Oriented Approaches to Pattern Recognition. Academic Press, New York.

Milne, L., 1995. Feature Selection using Neural Networks with Contribution Measures, AI'95 (November), Canberra.

O'Reilly, S., 1995. Why many mining tools cannot dig deep enough. Computer Weekly, 2 (February), 24.

Pawlak, Z., 1982. Rough sets. International Journal of Computer and Information Sciences 11 (5), 341–356.

Piramuthu, S., Shaw, M.J., 1994. On using decision tree as feature selector for feed-forward neural networks, In: International Symposium on Integrating Knowledge and Neural Heuristics. pp. 67–74.

Pudil, P., Ferri, F.J., Novovicova, J., Kittler, J., 1994. Floating search methods for feature selection with nonmonotonic criterion functions. IEEE 12th International Conference on Pattern Recognition. Vol. II, 279–283.

Quinlan, J.R., 1987. Simplifying decision trees. International Journal of Man–Machine Studies 27, 221–234.

Quinlan, J.R., 1990. Decision trees and decision making. IEEE Transactions on Systems, Man and Cybernetics 20 (2), 339–346.

Seshadri, V., Weiss, S.M., Sasisekharan, R., 1995. Feature extraction for massive data mining. In: Proceedings of First International Conference on Knowledge Discovery and Data Mining. pp. 258–262.

Setino, R., Liu, H., 1997. Neural network feature selector. IEEE Transactions on Neural Networks 8 (3), 654–662.

Siedlecki, W., Sklansky, J., 1989. A note on genetic algorithms for large-scale feature selection. Pattern Recognition Letters 10 (5), 335–347.

Stearns, S.D., 1976. On selecting features for pattern classifiers, Third International Conference on Pattern Recognition. pp. 71–75.

Tam, K.Y., Kiang, M.Y., 1992. Managerial applications of neural networks: The case of bank failure predictions. Management Science 38 (7), 926–947.

Toussaint, G.T., 1971. Note on optimal selection of independent binary-valued features for pattern recognition. IEEE Transactions on Information Theory IT 17, 618.

Whitebread, K.R., Jameson, S., 1995. Information discovery in high-volume, frequently changing data. IEEE Expert (October), 51–53.

Yang, J., Honavar, V., 1997. Feature subset selection using a Genetic Algorithm. In: Proceedings of the Genetic Programming Conference (GP'97). pp. 380–385.