Journal of the Operational Research Society (2005), 1–10

www.palgrave-journals.com/jors

Data mining feature selection for credit scoring models

Y Liu* and M Schumann

University of Goettingen, Germany

The features used may have an important effect on the performance of credit scoring models. The process of choosing the best set of features for credit scoring models is usually unsystematic and dominated by somewhat arbitrary trial. This paper presents an empirical study of four machine learning feature selection methods. These methods provide an automatic data mining technique for reducing the feature space. The study illustrates how four feature selection methods—'ReliefF', 'Correlation-based', 'Consistency-based' and 'Wrapper' algorithms help to improve three aspects of the performance of scoring models: model simplicity, model speed and model accuracy. The experiments are conducted on real data sets using four classification algorithms—'model tree (M5)', 'neural network (multi-layer perceptron with back-propagation)', 'logistic regression', and '*k*-nearest-neighbours'.

Journal of the Operational Research Society advance online publication, 20 April 2005
doi:10.1057/palgrave.jors.2601976

Keywords: insurance; risk; credit scoring; classification algorithms; feature selection

Introduction

One challenge researchers face when they use classification algorithms to build credit scoring models is which features to select. Because the real-world data used for scoring models might come from different sources and be collected for a general task, it is likely that the original data contain not only many observations, but often also a large number of features. Some of the features may be irrelevant to credit risk; some of them may be redundant due to their high intercorrelation. With so many irrelevant and redundant features, most of the classification algorithms suffer from extensive computation time, possible decrease in model accuracy and decrease in scoring interpretation.

To select the most predictive independent features and reduce redundancy, some scholars and practitioners have applied the following methods. For example, some used univariate analysis to evaluate the effect of each independent feature on the class feature, some used statistical correlation analysis to detect the correlation between different independent features and eliminate highly correlated ones. In addition, the process of feature selection is sometimes incorporated into the classification algorithms, for example, using stepwise statistical procedures in discriminant analysis and regression techniques, and selecting the most suitable features while building decision trees.

Since there is no economic theory to denote which features are relevant and which are not relevant to creditworthiness, the process of choosing the best set of features in practice is unsystematic and dominated by arbitrary trial.[1] Contrary to this unsystematic way, the method of feature selection using automatic data mining is defined as a formal process. Many machine learning researchers have developed various feature selection methods. Whether these methods are efficient for the feature selection problem in credit scoring models is rarely discussed.

This paper presents an empirical study of four machine learning feature selection methods to show their performance in a real-world credit data analysis problem. We begin by describing the used classification algorithms and the feature selection methods. Then, the credit analysis problem and the sampling of the data are introduced. After that, we present the process of feature selection and discuss the results of the experiment. Finally, the conclusion and related future work are provided.

Introduction of the classification algorithms

The four classification algorithms included in this paper represent three broad areas of machine learning methods and one regression algorithm. M5 is a variation of decision-tree methods. *k*-Nearest-neighbours method is one example of instance-based methods. The multi-layer perceptron neural network with back-propagation is a frequently used neural network method for credit scoring models. Logistic regression is a widely used classifier in the credit scoring

**Correspondence: Y Liu, Business School of Jilin University, 10 Qianwei Road, Changchun, PR China.*
E-mail: yliuddff@hotmail.com

industry nowadays. These classification algorithms are introduced below:

Model tree (M5)

The M5 model tree is a numeric prediction algorithm.[2] M5 algorithm generates a conventional decision-tree structure with linear regression models at the leaves, which would give a smoother score distribution instead of discrete class labels.

The basic tree is generated by a recursive partition method similar to C4.5. Then, a regression function is built for each node of the constructed tree using the standard regression algorithm. The regression functions are simplified by minimizing an estimate of the expected error, which is multiplied by a factor $(n+v)/(n-v)$:

$$E_{\text{error}} = \frac{\sum_{i=1}^{n} |\text{difference between predicted value and actual value}|}{n} \times \frac{n+v}{n-v}$$

Here n is the number of training cases that reach that node, v is the number of terms (including the constant and the independent variables) in the linear regression function at that node. Terms in the linear model are dropped one by one, greedily, so long as doing so decreases E_{error}. Thus, due to the factor $(n+v)/(n-v)$, the linear model may be simplified to minimize the E_{error}. Finally, once the final simplified linear model is placed for each node, the tree is pruned back from the leaves. If the E_{error} of a node is smaller than the E_{error} of the subtree below, the subtree is replaced by this single node.

A parameter of M5 algorithm is the pruning factor F, which decides the extent of the pruning. Trees with different size are generated by varying this parameter.

Multi-layer perceptron neural network with back-propagation (MLP)

MLP consists of the input, hidden and output layers of interconnected nodes. The nodes in the network are all sigmoid. Through multiple passes of training with the examples, the weights of the nodes are modified, based on the error rates of the resulting outputs.

Only one hidden layer is used in the network in the following experiments. The number of hidden nodes (H) is the only parameter to be determined. Network models have other parameters: learning rate (set to be 0.3) and momentum (set to be 0.2). If the network diverges from the answer, the network will be automatically reset with a lower learning rate and be trained again.

k-Nearest-neighbours (k-NN)

The k-NN algorithm in this study follows the method described in Aha *et al.*[3] In this method, a set of training examples is saved. A Euclidean distance metric is used to measure the similarity between each training example and a new example.

In this simple k-NN algorithm, features will not be weighted and they are treated equivalently. The algorithm finds the k examples nearest to the new example. The new example is assigned to the class to which the majority of the k neighbour examples belong. Neighbours will be weighed by the inverse of their distance when voting. The number of neighbours (parameter k) is the important factor. A suitable k is to be empirically determined for the data set used in order to smooth the influence of the noise in the data set.

Logistic Regression (LR)

Logistic regression can predict the probability (P) that an example X belongs to one of two predefined classes. Suppose example $X=(x_1, x_2, \ldots, x_k)$, as in linear regression, logistic regression gives each x_i a coefficient w_i which measures the contribution of each x_i to variations in P. First, a logistic transformation of P is defined as

$$\text{logit}(P) = \ln(P/(1-P))$$

where P can only range from 0 to 1, while logit (P) ranges from $-\infty$ to ∞. Logit (P) is then matched by a linear function of the feature variables:

$$\text{logit}(P) = \ln(P/(1-P)) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_kx_k$$

Parameters w_i are usually estimated by the maximum likelihood method.

Introduction of the feature selection methods

Feature selection using automatic data mining is defined as 'the process of finding a best subset of features, from the original set of features in a given data set, optimal according to the defined goal and criterion of feature selection (a feature goodness criterion)'.[4] Many researchers have developed various feature selection algorithms using different evaluation criterions and searching strategies.

Some feature selection methods use a measure to evaluate the goodness of individual features. For example, information measures, distance measures and dependence measures.[5] Features are ranked according to their values on this measure. One can simply choose the first X features as the selected feature subset. X is decided according to some domain knowledge or a user-specified threshold value. Another group of feature selection methods evaluates the goodness of a group of features. The aim is try to find good or poor feature subsets, not good or poor features. Searching for good feature subsets requires extensive computation time. Therefore, some search strategies are applied to decrease the number of subsets to be evaluated, for example, best first search and Beam search, forward/backward hill climbing search, and genetic search, etc.[5]

According to the nature of the measures used to evaluate features, feature selection methods can be divided into 'filter algorithms' and 'wrapper algorithms'.[6] Wrapper algorithms evaluate features with the classification accuracy provided by a target classification algorithm. Filter algorithms, on the other hand, are independent of any learning algorithms and use a particular measure that reflects the characteristics of the data set to evaluate features. A target classifier is included in the feature selection process of the wrapper approach. This leads to, on the one hand, the improved classification accuracy of the wrapped classifier and on the other hand, an increase in the computation time. If the wrapped classification algorithm is time-consuming, or if the train data set is large, the application of the wrapper approach may be unrealistic due to the enormous computation time required. Moreover, because the derived feature subset is biased to the wrapped classification algorithm, good performance may not occur when the feature subset is used to build models with other classification algorithms.

The study in this paper includes four feature selection algorithms (three filter algorithms and one wrapper algorithm). They are representative of the methods developed in recent years. In a study[7] that compared various feature selection algorithms using sixteen data sets from the UCI collection, these four algorithms give a better performance than two other classic algorithms: the information gain algorithm and the principal components analysis. The four algorithms are introduced below:

'ReliefF' feature selection algorithm (REF)

'Relief' algorithm is a feature ranking algorithm first proposed by Kira and Rendell[8] and was enhanced to 'ReliefF' by Kononenko.[9] The key idea of 'Relief' is to rank the quality of features according to how well their values distinguish between the cases that are near to each other. It is reasonable to expect that a useful feature should have different values between cases from different classes and have the same value for cases from the same class. The feature evaluation measure is denoted as '$\boldsymbol{M}_{\mathbf{REF}}$', which is calculated by the following algorithm:

```
Set $\boldsymbol{M}_{\mathbf{REF}}$ (feature A) $=$ 0;
for $i = 1$ to $m$ do: ($m$ is a user-specified number)
Begin
    randomly sample a case R;
    find its nearest neighbour from the same class (case S);
    find its nearest neighbour from a different class (case D);
    $\boldsymbol{M}_{\mathbf{REF}}$ (feature A) $=$ $\boldsymbol{M}_{\mathbf{REF}}$ (feature A)$-$diff (feature A, R, S)$/m+$diff (feature A, R, D)$/m$;
end;
```

where diff (feature A, case 1, case 2) is the normalized difference between the values of feature A for two cases. For categorical features, the difference is either 1 (the values of two cases are different) or 0 (the values of two cases are equal).

The following experiments use the enhanced method 'ReliefF', which smoothes the influence of noise in the data by averaging the contribution of k nearest neighbours of each sampled case. Two parameters need to be set when calculating '$\boldsymbol{M}_{\mathbf{REF}}$': m (the number of the sampled cases) and k (the number of nearest neighbours). The larger m implies more reliable approximation.[9] Therefore, in the experiments, m is set to be the number of all train cases, k is set to be 30. (Preliminary experiments with $k = 20$, 30 and 40 showed that the results of feature selection in this study are not sensitive with respect to k).

The correlation-based feature selection algorithm (CFS)

The correlation-based feature selection algorithm[10] uses a correlation-based measure to evaluate the worth of feature subsets. The hypothesis on which this method is based is: good feature subsets contain features highly correlated with the class, yet uncorrelated with each other.

Suppose a feature subset S contains k features, the evaluation measure for S is calculated as

$$\boldsymbol{M}_{\mathbf{REF}} = \frac{k\overline{r_{\mathrm{cf}}}}{\sqrt{k + k(k-1)\overline{r_{\mathrm{ff}}}}}$$

where $\overline{r_{\mathrm{ff}}}$ is calculated by averaging all of feature–feature correlation r_{ff} (the correlation between each pair of features in S), and $\overline{r_{\mathrm{cf}}}$ is calculated by averaging all of feature–class correlation r_{cf} (the correlation between each feature in S and the class feature). This measure assigns high values to subsets containing features that are highly correlated with the class and have a low intercorrelation with each other. In such a way, the subsets that contain irrelevant features (with low feature–class correlation $\overline{r_{\mathrm{cf}}}$) and redundant features (with high feature–feature correlation $\overline{r_{\mathrm{ff}}}$) are evaluated as bad subsets of features.

The correlation between two features is calculated with 'Symmetrical Uncertainty':

$$\begin{aligned} r_{\mathrm{AB}} &= \text{Symmetrical Uncertainty} \\ &= 2 * \frac{\text{Entropy}(\text{A}) - \text{Entropy}(\text{A}|\text{B})}{\text{Entropy}(\text{A}) + \text{Entropy}(\text{B})} \\ &= 2 * \frac{\text{Entropy}(\text{B}) - \text{Entropy}(\text{B}|\text{A})}{\text{Entropy}(\text{A}) + \text{Entropy}(\text{B})} \end{aligned}$$

In fact, ‘Symmetrical uncertainty’ is the normalized ‘information gain’. The numerator is the ‘information gain’, which is a symmetrical measure—that is, the amount of information gained about A after observing B is equal to the amount of information gained about B after observing A. Symmetry is a desirable property for a measure of feature–feature intercorrelation. In order to compensate for the bias of ‘information gain’ toward features with more values and to ensure they are comparable, ‘information gain’ is normalized by the entropy of features. If A and B are categorical features with respective ranges R_A and R_B, the following equations give the entropy of B before and after observing A.

$$\text{Entropy (B)} = -\sum_{b\in R_B} p(b)\log(p(b))$$

$$\text{Entropy (B|A)} = -\sum_{a\in R_A} p(a)\sum_{b\in R_B} p(b|a)\log(p(b|a))$$

The consistency-based feature selection algorithm (CON)

The consistency-based feature selection methods try to find the smallest set of features that can distinguish classes as if with the full set. In other words, the reduced feature subset retains the discriminating power of the data defined by the original features. This method can help remove both redundant and irrelevant features.[11]

The goodness of feature subsets is measured by a consistency rate:

$\boldsymbol{M}_{\mathbf{CON}}$ = Consistency rate = 1−inconsistency rate

An inconsistency rate over the data set for a given feature subset S is calculated as follows:

(1) Two patterns (a pattern is a set of values for S) are considered inconsistent if they have the same values for all features in S, but belong to different classes. For example, pattern A (0, 0, 1) belongs to class c_1 and pattern B (0, 0, 1) belongs to class c_2. They are inconsistent if $c_1 \neq c_2$.

(2) The inconsistency count for a pattern is the number of times it appears in the data set minus the number of times it appears in the majority class; for example, let us assume there are n pattern A in the data set, among which n_1 belong to c_1 and n_2 belong to c_2, where $n_1 + n_2 = n$. The inconsistency count of A is n-max $\{n_1, n_2\}$.

(3) The inconsistency rate of S is the sum of all the inconsistency counts for all possible patterns of S divided by the total number of patterns.

In fact, the full set of features has the largest value of ‘$\boldsymbol{M}_{\mathbf{CON}}$’ among all of the feature subsets. If the search is exhaustive, the consistency-based method can find the smallest feature subset that has the same value of ‘$\boldsymbol{M}_{\mathbf{CON}}$’ as the full set of features.

The wrapper feature selection algorithm (WRP)

In wrapper algorithms, the evaluation measure for the goodness of feature subsets is the classification accuracy provided by a target learning algorithm. The classification accuracy is estimated with a test data set in the wrapper algorithm in this paper:

$\boldsymbol{M}_{\mathbf{WRP}}$ = classification accuracy = 1−test error rate

The wrapper algorithm searches for the feature subset that generates the lowest error rate in the test data set.

If a decrease of the classification error rate is the foremost concern in building a scoring model, the wrapper feature selection approach is the appropriate method. The wrapper approach can generally give better results in model accuracy when using the target learning algorithm. However, due to extensive computation time required this approach cannot be used for large data sets and some time-consuming classification algorithms. In the empirical study section of this paper, the model tree algorithm M5, logistic regression and the k-NN algorithm are conducted using the wrapper approach, while the low speed of the neural network MLP algorithm prevents the use of the wrapper approach.

Description of the problem and sampling of the data

The data were obtained from a German credit insurance company. The kind of product provided by this insurance company can be described in this way: A company supplies goods or services to its clients. The clients do not pay for the received goods or services immediately but will pay later. The supplier asks for the insurance company to insure the payment of its clients. The granted insurance is related to the creditworthiness of the clients of the suppliers. These clients are called the risk partners of the insurance company.

To analyse the default probability of each risk partner, the insurance company’s current risk assessment system collects credit information about risk partners from different sources. One source of information is banks. A bank provides bank reports, which contain information of companies’ accounts in the bank as well as broad evaluations of their financial status and creditworthiness, etc. The bank reports are in free text. The free text has been mapped into feature vectors. For example, the text ‘The situation of company A’s development is expansive/positive/constant/stagnant/declining/strongly declining’ is mapped into a categorical feature named ‘Development’, which contains six values. Table 1 gives examples of features included in the data set. The data set includes 72 features, which are all categorical.

Since the actual payment behaviours of the available examples in the data set are not known, they are classified by

Table 1 Examples of features in the database

Name of feature	*Meaning of feature*
Development	The situation of the company's development
Reputation	Whether the company has a good reputation
Property	Whether the company has real property
Mort-property	The mortgage situation of real property
Account	The situation of account behaviour
Overdraft	The situation of overdrafts
Payment	The payment behaviour of debts
Tight-finance	Whether the financial status seems tight
……	……

Table 2 The ratings of companies given by the current system

A	Very good	Companies that have above-average good creditworthiness and very low default probability.
B	Rather good	Companies that have good or satisfied creditworthiness and low default probability.
C	Problematic	Companies that are problematic and have high risk of default.
D	Very bad	Companies that have very high credit risk, and the insurance applications associated with them must be refused.

the ratings given by the current risk assessment system of the insurance company. The four ratings A, B, C, D are assigned based on the credit experts' knowledge (see Table 2).

Available examples in the data set are classified into two classes, 'good' (with rating A/B) and 'bad' (with rating C/D). There are 38 283 examples in total, among them 34 562 are 'good' and 3721 are 'bad'.

Owing to the low prevalence of 'bad' examples in the data set, a random selection will result in a sample with a low proportion of 'bad' examples. It was argued by other researchers and also validated by preliminary experiments with this data set that low numbers of 'bad' examples lead to inaccurate parameter estimations in the neural network, model tree and *k*-nearest-neighbours models. Consequently, the models trained with such samples perform badly. A good result can be achieved by using balanced selection of samples. 'Balanced' means that the train sample includes the same number of 'good' and 'bad' examples. Figure 1 illustrates the selecting of the balanced samples, all of the 'bad' examples are selected and the same number of 'good' examples is randomly selected in order to ensure the selected examples are representative. The selected examples are randomly divided into three samples: Train Sample, Test Sample 1 and Test Sample 2.

Models during the feature selection are trained with the Train Sample, and tested on the Test Sample 1. To increase the number of train examples, the final models after feature selection are built on the Large Train Sample, which consists of examples from both the Train Sample and the Test Sample 1. Test Sample 2 is used to evaluate and compare the final models' performances. Test Sample 2 is unseen during the feature selection and the model building process in order to prevent overfitting and ensure the validity of the final evaluation.

The process of the feature selection

The search strategy

The CFS, CON and WRP methods measure the goodness of feature subsets rather than each single feature. An exhaustive search for the data set with 72 features is unrealistic due to the enormous computation time required. Therefore, heuristic search methods need to be used. Different search

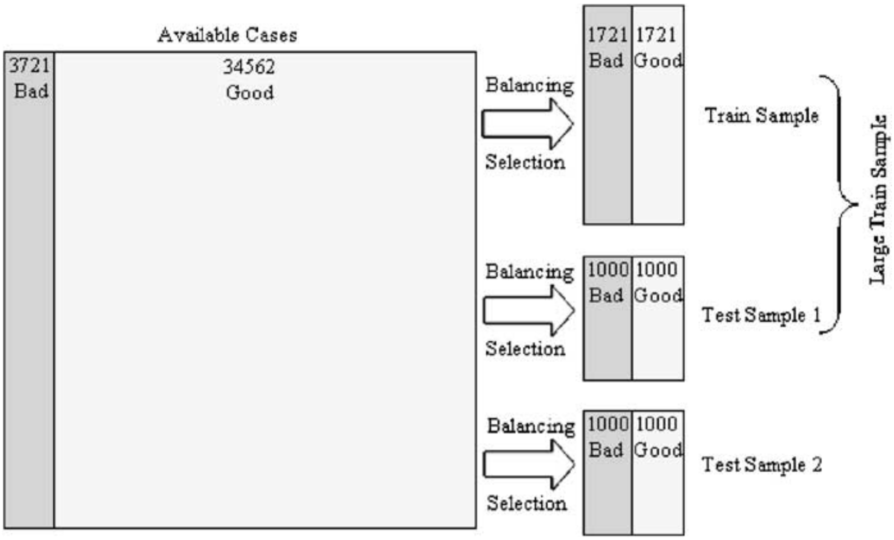

Figure 1 The three samples for the experiments.

methods may lead to different results. Greedy hill climbing search strategies such as forward selection and backward elimination are often applied to search the feature subset space in a reasonable time. Although simple, these searches often yield good results[12] compared to more sophisticated search strategies. In order to compare the four feature selection methods, a search method, which is similar to 'forward hill climbing', presented by Hall and Holmes[7] is used in the following experiments.

Forward hill climbing search starts with an empty set and evaluates each feature to find the best one. Then it combines each of the remaining features with the best one and evaluates them to find the best pair of features. This process continues until no additional feature can improve the evaluation of the feature subset. Its termination condition can be altered in order to generate a ranked list of features: even though the adding of any of the features cannot increase the evaluation measure, the search continues, and the feature that gives the least decrease in the evaluation measure is selected. The process continues until all features are selected. The adding order of each feature is recorded. Finally, a ranked features list is obtained according to their incremental improvements to the evaluation measure.

Features can be ranked according to '$\boldsymbol{M}_{\mathbf{REF}}$'; through using the search method described above, features can also be ranked according to the other three evaluation measures '$\boldsymbol{M}_{\mathbf{CFS}}$', '$\boldsymbol{M}_{\mathbf{CON}}$' and '$\boldsymbol{M}_{\mathbf{WRP}}$'.

The determination of model parameters

To select the first *X* features from the ranked features list, 72 models are trained by adding features one by one. A learning curve will show the changing of models' accuracy. Because the models will be built on different subsets of features, the determination of models' parameters is problematic. The optimal model parameter for one feature subset may not be optimal for other feature subsets. Considering that several hundreds of models need to be built in the experiments, if parameters are tuned for each of the models, then the time spent on parameter adjustments is unrealistically large. For example, choosing the parameter H with 10-fold cross validation for MLP-BP models takes from more than ten minutes to more than 10 hours, depending on the number of features used.

To solve this problem, some preliminary experiments were carried out. Parameters are tuned for the models that use the first 10, 30, 50 and 70 features in the ranked list. These experiments' results shown in Figure 2 illustrate the change of model error rates with different k when different feature subsets are included in the *k*-NN models. The error rates in the figure are the 10-fold cross validation error rates with the Train Sample. It shows that k should be set to different values (1, 3 or 5) for models using 10, 20, 30, 50, 70 features. But the small differences in error rates when *k* is set to 1, 3 or 5 implies that not much is lost if the parameters are not chosen strictly according to the minimal error rates.

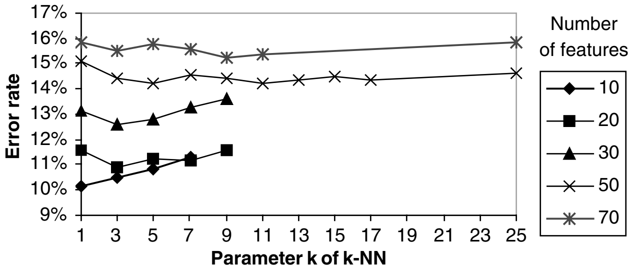

Figure 2 The selection of parameter *k* for *k*-NN models.

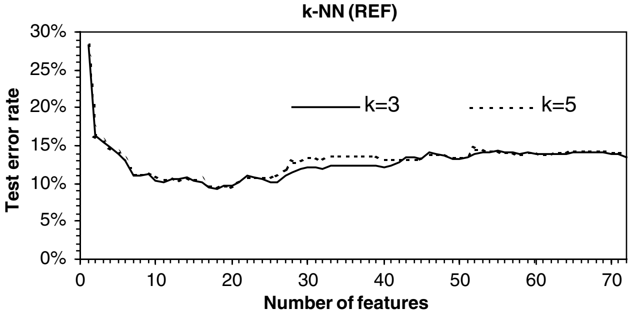

Figure 3 The learning curves for *k*-NN models with $k=3$ and $k=5$.

These preliminary experiments simplify the setting of parameters for models. *k* is set to 3, no matter how many features are used. To show the reasonability of this simplification, the learning curves (with REF method) are drawn for the *k*-NN models using $k=3$ and $k=5$, respectively (see Figure 3).

Figure 3 shows that the forms of learning curves are not significantly affected with $k=3$ or 5. The final selection of features based on the learning curves is reliable even without the fine tuning of parameters for each model individually. Therefore, in the following experiments, the models' parameters are set to be the same for models at the different points in the learning curves. Other experiments on M5 and MLP showed the same results; parameter *F* of M5 algorithm and parameter *H* of MLP algorithm are set to be fixed values for all models: $F=1$, $H=7$.

Although the models may not reach their best classification accuracies due to a fixed parameter, this simplification is justifiable because the aim of the learning curve is not to find the model with the optimal classification accuracy but rather to reduce the number of features. After the feature selection, the building of each final model will be based on the individual tuning of parameters.

The learning curves of the feature selection

Learning curves were drawn, which demonstrate the changing of the test error rates on Test Sample 1 when

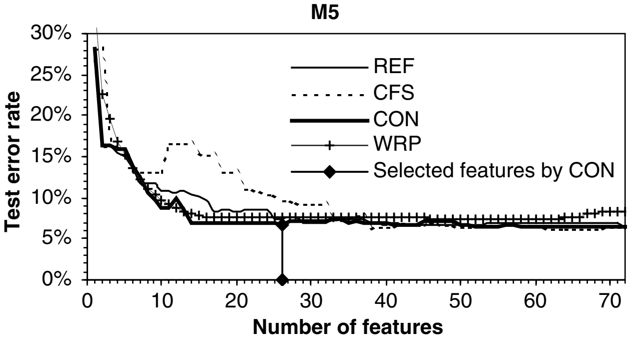

Figure 4 The learning curves of feature selection (M5).

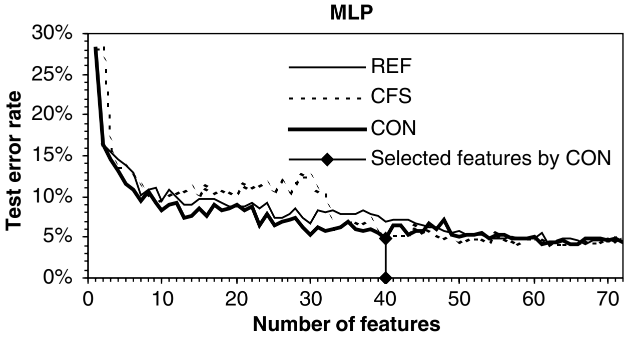

Figure 5 The learning curves of feature selection (MLP).

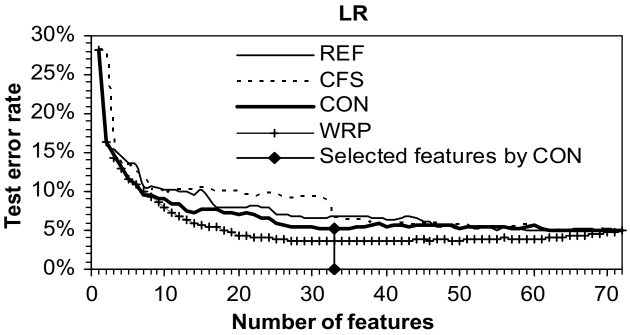

Figure 6 The learning curves of feature selection (LR).

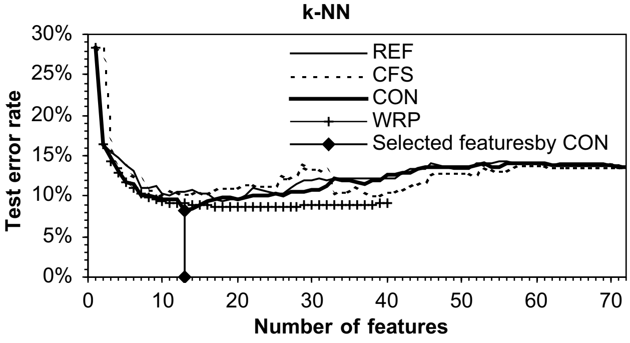

Figure 7 The learning curves of feature selection (*k*-NN).

adding features one by one (see Figures 4–7). The learning curve for the *k*-NN with WRP method was not completed. Owing to the large computation time required and the fact that adding further features will obviously increase the test error rate, the points after 40 features in the learning curve are ignored.

The final selected features are determined by observing the learning curves. The principle is to use fewer features to reach the lower or the same level of the test error rate. The numbers of selected features by each feature selection method are discussed in next section. Here the selected features by CON are illustrated in Figures 4–7 as examples.

The comparison of results

The final models were trained with the selected features. The Large Train Sample was used as the training data set. Parameters were tuned according to the 10-fold cross validation error rates. Models' performances were tested on Test Sample 2. The effects of feature selection can be shown by comparing the models' performances before and after feature selection. Three aspects were considered:

- Model simplicity: the number of used features.
- Model speed: the time of training models with M5 and MLP algorithms. Since the *k*-NN model has little training time, its model speed denotes the time of testing models with Test Sample 2.
- Model accuracy: the test error rates on Test Sample 2.

The performances of the models without feature selection (all 72 features were used as train data) are shown in Table 3.

Figure 8 shows the number of selected features by the four feature selection methods (REF, CFS, CON, and WRP). The reduction of features was most significant in the *k*-NN models, and a minimal 13 features were selected by the CON method. The reduction of features for M5 and LR models was also remarkable, where the least number of features (26 for M5, 33 for LR) were selected by the CON method. The significant reduction of features for MLP models was also achieved by CON (40 features), while REF and CFS could not reduce the number of features significantly for MLP models. In general, the CON and WRP method were better in reducing the number of features.

Table 3 Models before feature selection

Models	*No. of features*	*Test error rate on the Test Sample 2 (%)*	*Model speed (s)*
M5	72	7.15	655
MLP	72	4.2	7244
LR	72	4.95	325
k-NN	72	12.75	807

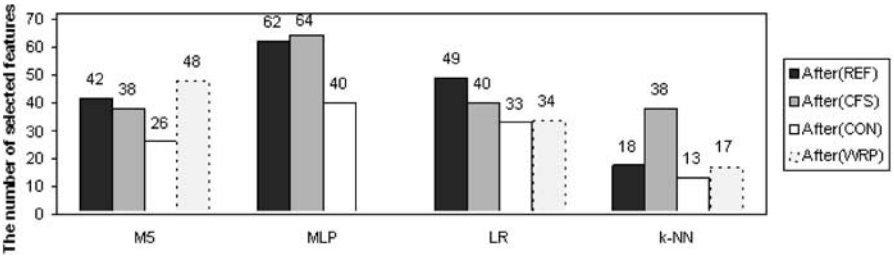

Figure 8 The number of selected features.

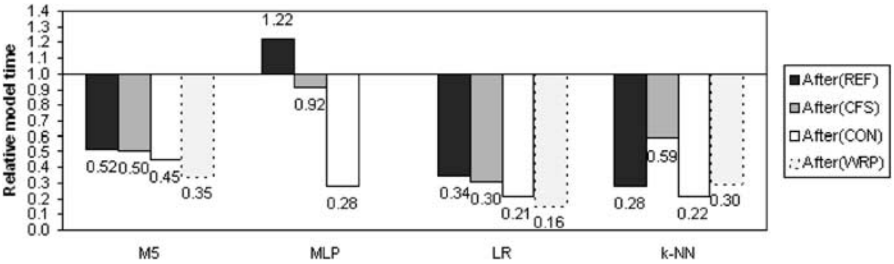

Figure 9 The change in the model time after feature selection.

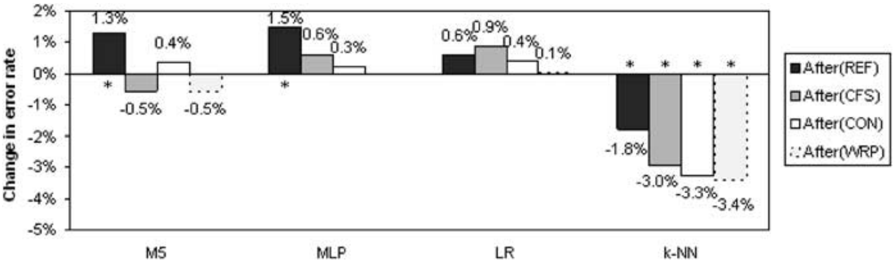

Figure 10 The change in model error rates after feature selection.

The model speeds are shown with their relative values compared with the models without feature selection. The models with all 72 features are used as the basic models in the comparison. Their performances are assumed as 1. Being divided by the values of the basic models, the speeds of each model after feature selection are expressed as relative values (see Figure 9). The improvements in model speed are significant for M5, LR and *k*-NN models when using every feature selection method. A considerable improvement in MLP models' speed happened only in the CON method—28% of the original training time was needed. Since the training of the MLP model converged slower when using the features selected by the REF method, it took more time than before.

Figure 10 shows the differences in error rates between the models before and after feature selection. Significance tests were carried out using 2 as the Z value (critical value). The sign '*' denotes that the difference between error rates before and after feature selection was significant. The changes in error rates were small for the M5, LR and MLP models after feature selection (no significant change except M5 (REF) and MLP (REF)), while the decreases in error rates for *k*-NN models were significant.

As shown in Table 4, in general, for the M5 and LR models, except REF all other feature selection methods can reduce the number of features significantly and at the same time keep the models' accuracy. For MLP models, only the feature selection method CON can reduce the number of features significantly and at the same time keep the same level of model accuracy. For *k*-NN models, all feature selection methods can reduce the number of features and improve the model accuracy significantly. Among the four methods, CON and WRP are better both in the reduction of the number of features and in the improvement of model accuracy.

Conclusion and outlook

The empirical study of the feature selection methods for the credit scoring model in this paper illustrates how different feature selection methods perform on one real data set.

Table 4 The comparison of the results across all the experiments

	REF	*CFS*	*CON*	*WRP*
M5	(1) 42 (2) 52% (3) −	(1) 38 (2) 50% (3) 0	(1) 26 (2) 45% (3) 0	(1) 48 (2) 35% (3) 0
MLP	(1) 62 (2) 122% (3) −	(1) 64 (2) 92% (3) 0	(1) 40 (2) 28% (3) 0	
LR	(1) 49 (2) 34% (3) 0	(1) 40 (2) 30% (3) 0	(1) 33 (2) 21% (3) 0	(1) 34 (2) 16% (3) 0
k-NN	(1) 18 (2) 28% (3) +	(1) 38 (2) 59% (3) +	(1) 13 (2) 22% (3) +	(1) 17 (2) 30% (3) +

(1) Change in simplicity (no. of features).
(2) Change in speed (percentage of the original model time).
(3) Change in accuracy (+: increase −: decrease 0: no change).

In conclusion, we summarize the following as the main results:

- Among the four feature selection methods, the consistency-based and the wrapper feature selection methods perform relatively better. This has been illustrated in the learning curves (see Figures 4–7). The learning curves of 'CON' and 'WRP' are lower than the other two curves in the earlier range (from 0 to about 33 features), which means, models of curves of 'CON' and 'WRP' reach the lower error rates earlier (in other words, with fewer features) than models of the other two curves. It is obvious that the feature lists generated by the methods of 'CON' and 'WRP' are superior to the other two methods.
- After feature selection, improvements in model accuracy are shown for the *k*-NN algorithm, but no improvement is shown for other algorithms. This result may be caused by the nature of the algorithms. The model trees method can select the useful features during the building of the tree structure. The neural networks are trained to assign a small weight to the irrelevant features by learning from data. The logistic regression algorithm weighs the used features differently according to their predictability. This implies that these three algorithms are not seriously affected by including some redundant or irrelevant features. In contrast, since the simple *k*-NN algorithm treats all features equivalently, the presence of irrelevant and redundant variables is always a problem. Therefore, the *k*-NN models get the largest improvements in classification accuracy from feature selection.
- Apart from classification accuracy, other aspects (eg simplicity, speed) are important factors when choosing a model in practice. The reduction in the number of features decreases the training time and simplifies the final models. Sometimes in practice, it is necessary and deserving to reduce the feature space even if it means a small sacrifice in accuracy, especially when a large number of features are present.

To reach a more generalizable conclusion, experiments need to be conducted on other data sets; some related work will be studied in further research, in which other classification algorithms and feature selection methods may be included in the comparison study. In addition, it would be meaningful to show the impact of the studied feature selection methods on other classification criteria, for example, the area under the receiver operating characteristic curve.

Although the conclusions in the paper need to be validated with other data sets in further research, a general conclusion can be drawn from this study: the automated data mining feature selection technique provides an effective method for selecting the most predictable features from many presented features. We are sure that in the future it will play an important role in building credit scoring models.

Acknowledgements—The algorithms in this study are implemented by the free data mining tool 'Weka'.[13]

References

1 Hand DJ and Henley WE (1997). Statistical classification methods in consumer credit scoring: a review. *J R Stat Soc Seri A* **160**: 523–541.

2 Quinlan JR (1992). Learning with continuous classes. In: Adams A and Sterling L (eds). *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*. World Scientific, Singapore, pp 343–348.

3 Aha D, Kibler D and Albert M (1991). Instance-based learning algorithms. *Mach Learn* **6**: 37–66.

4 Cios KJ, Pedrycz W and Swiniarski RW (1998). *Data Mining Methods for Knowledge Discovery*. Kluwer Academic Publishers: Boston.

5 Liu H and Motoda H (1998). *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers: London.

6 Kohavi R and John GH (1997). Wrappers for feature subset selection. *Artif Intell* **97**: 273–324.

7 Hall MA and Holmes G (2000). *Benchmarking attribute selection techniques for data mining*. Working paper 00/10, Department of Computer Science, University of Waikato, Hamilton, New Zealand.

8 Kira K and Rendell LA (1992). A practical approach to feature selection. In: Sleeman D and Edwards P (eds). *Proceedings of the 9th International Conference on Machine Learning*. Morgan Kaufmann, Aberdeen, Scotland, pp 249–256.

9 Kononenko I (1994). Estimating attributes: analysis and extensions of RELIEF. In: De Raedt L and Bergadano F (eds). *Proceedings of the European Conference on Machine Learning*. Springer, Berlin, pp 171–182.

10 Hall MA (2000). Correlation-based feature selection for discrete and numeric class machine learning. In: Langley P (ed).

Proceedings of the 17th International Conference on Machine Learning. Morgan Kaufmann, San Francisco, CA, pp 359–366.

11 Dash M, Liu H and Motoda H (2000). Consistency based feature selection. In: Terano T, Liu H and Chen ALP (eds). *Knowledge Discovery and Data Mining: Current Issues and New Applications, the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, Berlin, pp 98–109.

12 Hall MA and Smith LA (1997). Feature subset selection: a correlation based filter approach. In: Kasabov N *et al* (eds). *Proceedings of the 4th International Conference on Neural Information Processing and Intelligent Information Systems*. Springer, Singapore, pp 855–858.

13 Witten IH and Frank E (2000). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgen Kaufmann Publishers: San Francisco, CA.

Received October 2003;
accepted December 2004 after one revision