Expert Systems with Applications 36 (2009) 5900–5908

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Evolutionary-based feature selection approaches with new criteria for data mining: A case study of credit approval data

Chia-Ming Wang [a], Yin-Fu Huang [b,c,*]

[a] Graduate School of Engineering Science and Technology, National Yunlin University of Science and Technology, 123 University Road, Section 3, Touliou, Yunlin 640, Taiwan, ROC
[b] Graduate School of Computer Science and Information Engineering, National Yunlin University of Science and Technology, 123 University Road, Section 3, Touliou, Yunlin 640, Taiwan, ROC
[c] Department of Computer and Communication Engineering, National Yunlin University of Science and Technology, 123 University Road, Section 3, Touliou, Yunlin 640, Taiwan, ROC

ARTICLE INFO

Keywords:
Data Mining
Evolutionary algorithm
Feature selection
Multi-objective optimization

ABSTRACT

In this paper, the feature selection problem was formulated as a multi-objective optimization problem, and new criteria were proposed to fulfill the goal. Foremost, data were pre-processed with missing value replacement scheme, re-sampling procedure, data type transformation procedure, and min-max normalization procedure. After that a wide variety of classifiers and feature selection methods were conducted and evaluated. Finally, the paper presented comprehensive experiments to show the relative performance of the classification tasks. The experimental results revealed the success of proposed methods in credit approval data. In addition, the numeric results also provide guides in selection of feature selection methods and classifiers in the knowledge discovery process.

1. Introduction

Recently, data mining or knowledge discovery in databases (KDD) has emerged as a very active, evolving area in information technology. Hundreds of novel mining algorithms and new applications such as medicine, business, and engineering have been proposed in the last decade. The aim of data mining is to extract knowledge from data (i.e., to help human finding and interpreting the 'hidden information' in massive raw data). The information and knowledge mined from the large quantities must be meaningful enough to lead to some advantages, usually economic advantages (Witten & Frank, 2005).

A credit scoring technique is the set of decision models and their fundamental techniques assist lenders in the granting of consumer credit (Thomas, 2000). It has been extensively used for credit admission evaluation in recent years. The basic principle of credit scoring is based on the analysis of the past performance of consumers to predict the credit score of those who will be assessed. In fact, the essential operations and philosophy are similar to the knowledge discovery process. Researchers have developed a variety of parametric statistical models such as LDA and logistic regression models (Desai, Crook, & Overstreet, 1996) for credit scoring. Nevertheless, assumptions of the underlying probability distribution are essential part of these methods. Moreover, those methods also assume linear relationships between attributes. These restrictions or shortages decrease the predictive accuracy of the credit scoring models and prevent their success.

In this paper, we applied meta-heuristic search techniques to find approximations of Pareto optimal set for the feature selection problem. Moreover, we proposed two new objectives for this combination optimization problem. Some pre-processing steps were conducted before the knowledge discovery process.

The primary contributions of the paper are as follows:

1. Since the feature selection problem could be considered as a combination optimization problem, the paper proposed new criteria for single/multiple objective evolutionary feature selection. The paper presented comprehensive experiments to show the relative performance of the classification tasks in the knowledge discovery process.
2. The results of an empirical study presented the relative performance of five different feature selection techniques. The results show:
   (a) New criteria with evolutionary algorithm outperform other feature selection methods.
   (b) K-nearest neighbor classifier usually produces poor performance no matter what performance measure is used.

The remainder of this paper is organized as follows. Section 2 described the workflow of the knowledge discovery process. How we preprocess data instances were described precisely in the section.

* Corresponding author. Tel.: +886 5 5342601x4314; fax: +886 5 5312063.
E-mail addresses: wang.chia.ming@gmail.com (C.-M. Wang), huangyf@yuntech.edu.tw (Y.-F. Huang).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.07.026

Section 3 introduced the feature selection problem and proposed solutions. The new objectives for single/multiple objective optimization were proposed in the section. In Section 4, we presented experimental setting and results. Finally, we concluded in Section 5.

2. Learning system

2.1. Workflow

Data pre-processing is always the first step (even the most important one) in the data mining workflow. Without getting to know data carefully in advance, the classification task could be misleading. First, the whole data sets were dealt with missing value replacement scheme. Then, a re-sampling procedure, including up-sampling scheme and down-sampling scheme, was performed for tackling a data imbalance issue. Finally, nominal attributes of data instances were transformed into numeric attributes, then normalized by a min–max normalization procedure, and finally fed into a feature selection module sequentially.

After going through the pre-processing procedures and feature selection module, the whole data is randomly divided into five divisions of equal size. The class in each division is represented in nearly the same proportion as that in the whole data set. Each division is held out in turn and the remaining four-fifths are directly fed into the classifiers. Thus, classifiers are executed 5 times on different training sets. This *k*-fold cross validation procedure could minimize the impact of data dependency and prevent the over-fitting problem (Hsu, Chang, & Lin, 2003). The detail workflow is shown in Fig. 1.

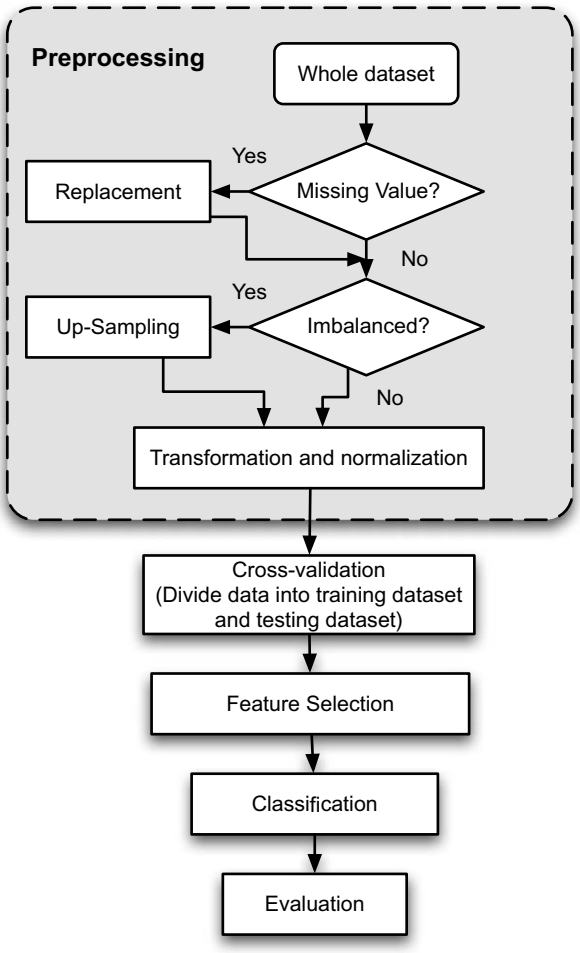

Fig. 1. Workflow of knowledge discovery.

2.2. Pre-processing

In this section, we explain how the data instances are pre-processed. The whole data sets were dealt with missing value replacement scheme, re-sampling procedure, data transformation procedure, and min-max normalization procedure sequentially.

2.2.1. Missing value replacement

Since most data sets encountered in practice contain missing values and most learning schemes lack for ability to handle these data sets, we have replaced missing values with the average or mode of attributes depending on their attribute types; i.e., numerical or categorical ones. Indeed, it seems to be convenient alternatives to remove all of these instances as long as the quantities of data are not too many.

2.2.2. Re-sampling

Recently, the class imbalance problem has been an interesting topic in machine learning and data mining community (Weiss & Provost, 2001). When classes are imbalanced, it would cause seriously negative effects on the classification performance; i.e., the overall error rate (Drummond & Holte, 2003). Most practical classifiers not designed for cost-sensitivity do much better on majority classes since they have a bias towards generality. However, in the worst case, classifiers do nothing on minority classes and predict the entire sample to majority ones. Cost-sensitive learning and re-sampling are two general methods to handle this problem, although there is no consistent winner. Since most algorithms are not cost-sensitive inherently, we adopt a re-sample approach to deal with this problem.

2.2.3. Data transformation and normalization

Some machine learning schemes such as neural network and SVM require that each data instance is represented as a vector of real numbers. Therefore, we have to convert the nominal attributes into numeric data before feeding into classifiers. Instead of using a single-number to represent a nominal attribute, we used *k* numbers to represent all *k* distinct nominal values of an attribute. That is, only one of the *k* numbers is one, and others are all zero. Apparently, this coding uses more numeric attributes to represent one nominal attribute, but it might be more stable than using a single-number if distinct values of an attribute are not too many (Hsu et al., 2003).

In order to prevent attributes with large numeric ranges dominate those with small numeric ranges, data instances are rescaled between 0 and 1 using min–max normalization procedure. The min-max normalization procedure performs a linear transformation of the original input range into a new specified range. The old minimum min_old is mapped to the new minimum min_new (i.e., 0) and max_old is mapped to max_new (i.e., 1), as shown in Eq. (1).

$$\text{New\_value} = \frac{\text{original\_value} - \text{min\_old}}{\text{max\_old} - \text{min\_old}}(\text{max\_new} - \text{min\_new}) + \text{min\_new} \quad (1)$$

3. Feature selection

The feature selection module in the knowledge discovery process aims to select the most relevant features. There are three regular approaches for feature selection – filter-based, wrapper-based and embedded-based ones (Huang, 2003). The filter approach

selects relevant features before the classification algorithm is applied. Therefore, this approach is a general feature selection method independent of classification algorithms. On the other hand, the wrapper approach includes a target classifier as a black box for performance evaluation. In other words, a computation-intensive evaluator is performed many times on candidate feature subsets to choose relevant features. The embedded approach is the inherent ability of a classification algorithm; i.e., feature selection is occurred naturally as a part of learners. Since the embedded-based approach is algorithm-specific, it is not an adequate one for our requirement. Moreover, even if we could decide a suitable classification algorithm and parameters for the wrapped approach, the classification algorithm should be further modified and the computation loading is heavy. Thus, we adopt the filter model to select the most relevant features in the paper.

3.1. Feature ranking with chi-squared statistics

This filter ranks features individually by measuring their chi-squared statistics with respect to the classes (Cantu-Paz, Newsam, & Kamath, 2004). For a continuous attribute, the range of the attribute should be discretized into several intervals first. In the discretization, we adopt the entropy-based technique to divide these continuous attributes. Subsequently, the chi-squared statistics of an individual attribute is computed based on the frequency counts tabulated in a contingency table. Without loss of generality, Table 1 shows as an example of a contingency table of binary classes for ith feature divided into four intervals.

The contingency table has rows for classes and columns for possible intervals of a feature. Each entry in the table denotes a frequency count. The chi-squared statistics of an attribute is defined as

$$\chi_i^2 = \sum_j \sum_k \frac{f_{jk} - e_{jk}}{e_{jk}} \tag{2}$$

The expected frequency is calculated as below

$$e_{jk} = \frac{f_{j+} \times f_{+k}}{N} \tag{3}$$

where row sum f_{j+} represents the support count for classes, and column sum f_{+k} represents the support count for intervals.

More precisely, row sum f_{1+} is the marginal frequency of the true class in Table 1, regardless of information about intervals, typically obtained by summing the joint probability f_{jk} over all intervals.

3.2. Feature ranking with relief

Relief developed by Kira and Rendell (1992) is able to estimate the quality of attributes. The basic idea of relief is to evaluate attributes with respect to how well their values discriminate among the instances that are near to each other. In other words, each attribute is individually assigned a weight according to its relevance to the class label, and a subset of features with high weights is selected. For continuous attributes, the difference between the values of two attributes is calculated according to diff function; i.e., squared difference. The pseudo code of relief is shown in Fig. 2.

Table 1
A 4-interval contingency table for ith feature

	I_1	I_2	I_3	I_4	
True	f_{11}	f_{12}	f_{13}	f_{14}	f_{1+}
False	f_{21}	f_{22}	f_{23}	f_{24}	f_{2+}
	f_{+1}	f_{+2}	f_{+3}	f_{+4}	N

3.3. Evolutionary feature selection with relative correlations

Here, we proposed a new filter-based feature selection algorithm, which is based on a stochastic evolutionary strategy and relative correlations of the feature set. The correlation between two variables reflects the degree to which the variables are related. The basic idea is to find out the best attribute subset with the property of lower intra-correlation within the feature set, but higher inter-correlation between the set and the corresponding class, thereby making the set discriminatory. High correlations (either positive or negative) within the feature set are less discriminative for class discrimination. However, high correlations between the set and the corresponding class indicate such feature set may be the relevant one for later classification task. We adopt the well-known Pearson's correlation coefficient (Montgomery & Runger, 2006) as the based measure for relative correlation calculations. The Pearson's correlation coefficient between two variables is defined as follows:

$$\mathrm{corr}(\mathbf{x},\mathbf{y}) = \frac{\frac{1}{n-1}\sum_{k=1}^{n}(x_k-\bar{x})(y_k-\bar{y})}{\sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(x_k-\bar{x})^2} * \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(y_k-\bar{y})^2}} \tag{4}$$

where n is the number of observations for variable x and x-bar is the average of variable x.

The Pearson's correlation between attributes also reflects the degree of the linear relationship between two variables, which ranges from +1 to −1. While the correlation coefficient is +1, it indicates a perfect positive linear relationship between two variables and vise verse. If the correlation is 0, then there is no linear relationship between them. Here, we proposed a new correlation measure among variables of the selected feature subset.

Definition 1. The overall correlation between the selected feature subset and the corresponding class (i.e., inter-correlation) is defined as follows:

$$RT(\mathbf{S},y) = \frac{1}{|\mathbf{S}|}\sum_{i=1}^{|\mathbf{S}|}|\mathrm{corr}(\mathbf{x}_i,y)| \tag{5}$$

where $|\mathbf{S}|$ is the cardinality of the selected feature subset $\mathbf{S}$, and y is the output class. The overall absolute correlation of each (feature, class) pair is averaged by $|\mathbf{S}|$.

Definition 2. The overall correlation within the selected feature subset (i.e., intra-correlation) is defined as follows:

$$RI(\mathbf{S}) = \frac{1}{C(|\mathbf{S}|,2)}\sum_{i=1}^{|\mathbf{S}|}\sum_{j=i+1}^{|\mathbf{S}|}|\mathrm{corr}(\mathbf{x}_i,\mathbf{x}_j)| \tag{6}$$

where $C(|\mathbf{S}|,2)$ is the number of 2-combinations (each of size 2) from the selected feature subset $\mathbf{S}$. In order to reduce the effect of the set size, the overall pairwise correlations within set $\mathbf{S}$ is divided by $C(|\mathbf{S}|,2)$.

Definition 3. The relative overall correlation combining both inter-correlation and intra-correlation is defined as follows:

$$RC(\mathbf{S},y) = \frac{RT(\mathbf{S},y)}{RI(\mathbf{S})} = \frac{1}{C(|\mathbf{S}|,2)}\sum_{i=1}^{|\mathbf{S}|}\sum_{j=i+1}^{|\mathbf{S}|}|\mathrm{corr}(\mathbf{x}_i,\mathbf{x}_j)| \tag{7}$$

As mentioned before, since the desired feature subset is the one with higher intra-correlation and lower intra-correlation, the RC value should be as large as possible.

Procedure Relief ($\mathbf{T}$, S_{max})

Input: $\mathbf{T}$ (sequence of N label examples), S_{max}

$\{\mathbf{T} = \left(\mathbf{x}^i, y_i\right)_{i=1}^{N}$, where $\mathbf{x}^i \in \mathbf{X} \subset \mathbf{R}^d$ is the vector of the i^{th} data point, and

$y_i \in \{-1, 1\}$ is its class label$\}$

Output: $\mathbf{W}$ (weight of attributes)

Begin

1. Initial the weight $\mathbf{W}(i)$ to zero for all attributes, where $i = 1, 2, .., d$
2. For each S (number of iteration), do the following:
   (a) Randomly select an instance t from $\mathbf{T}$
   (b) Find the nearest neighbor h of t from the same class and the nearest neighbor m of t from the different class
   (c) For each i (number of attribute), **do**

$$\mathbf{W}(i)^{new} = \mathbf{W}(i)^{old} - \mathrm{diff}(t^i, h^i) + \mathrm{diff}(t^i, m^i)$$

Fig. 2. Procedure relief.

Here, we describe the proposed feature selection algorithm – EAFS. Since the basic structure of EAFS is based on an evolutionary algorithm, we have to define the genome structure of an individual (or solution) first. Intuitively, the original feature set without the corresponding class is encoded as an n-bit binary string, where n is the cardinality of the original feature set. Consequently, zero or one indicates the corresponding feature is selected or not. Besides, a fitness function should be given in EAFS; i.e., the function used to calculate the quality of the candidate solutions represented by chromosomes. Here, we use the relative correlation defined in Eq. (7) as the fitness function. The detailed process of EAFS is shown in Fig. 3.

3.4. Multi-objective evolutionary feature selection with relative correlations

The primary shortage of the single-objective optimization is that it could not find out the right relationship between objectives. However, the Pareto approach could be a solution when it is difficult to combine objective functions into a single-objective function. The most widely adopted techniques for finding Pareto optimal solutions are the weighted methods and constraint methods. In recent years, evolutionary algorithms were widely used to solve this problem. The non-dominated sorting genetic algorithm II (NSGA-II), which was introduced by Deb, Agrawal, Pratab, and Meyarivan (2000) is one of the representations. In the paper, we adopt NSGA-II, and inter-correlation and intra-correlation defined in Section 3.3 to solve the feature selection problem.

First, the population is sorted into several ranks using non-dominated sorting. Each individual is assigned a fitness value according to the ranking levels, as shown in Fig. 4. The lower (frontier) the level is, the better the fitness value is. Then, the population is reproduced into double size after genetic operator crossover and mutation are applied. NSGA-II also incorporates an elitism scheme to maintain the solutions. That is, individuals with higher crowding distances are assigned better fitness values. Finally, the procedures are repeated until the stopping criterion is met.

For detailed discussions, literature reviews, and recent developments on multi-objective evolutionary algorithms, see Kalyanmoy and Kalyanmoy (2001).

4. Experiments

In the section, we introduced the data sets used in this work first of all. Second, classifiers and experimental settings were explained. Finally, experimental results were presented.

4.1. Data sets

Two data sets were used in the experiments; i.e., Australian and Germany credit data sets obtained from the UCI Repository of Machine Learning Databases (Asuncion & Newman, 2007).

The Australian credit approval data set originated from Quilan contains 690 instances, where 307 instances are creditworthy applicants and 383 instances are not creditworthy. Each instance contains 6 nominal, 8 numeric attributes, and 1 class attribute (accepted or rejected). The data set has a good mixture of attribute types: continuous, nominal with small numbers of values, and nominal with large numbers of values, plus a few missing values. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data.

The German credit scoring data are more unbalanced, and it consists of 700 instances of creditworthy applicants and 300 instances whose credits should not be extended. Each instance contains 7 nominal, 13 numeric attributes, and 1 class attribute (accepted or

Procedure EAFS ($Pr_{crossover}$, $Pr_{mutation}$, M_{max}, N, S)

Input: $Pr_{crossover}$ (Probability of crossover), $Pr_{mutation}$ (Probability of mutation), M_{max} (size of population), N (proportion of elitism), S_{max} (max generations)

Output: L (k-bit chromosome)

Begin

1. Randomly generate and initial population of chromosomes of size M_{max}.
2. For each S (number of generation), do the following:
   (d) Calculate the fitness of each individual chromosome according to Equation (5).
   (e) Rank the individual chromosomes according to RC. Pick up the first chromosomes as offspring chromosomes.
   (f) Select two chromosomes for mating from the current population. Parent chromosomes are selected with a probability related to their fitness.
   (g) **If** Rand[0, 1] < $Pr_{crossover}$
   then crossover the pair at a randomly chosen bit
   else change each bit with probability $Pr_{mutation}$.
   (h) Place the created offspring chromosomes in the new population.
   (i) Repeat Step (d) until the size of the new chromosomes population becomes equal to the size of the initial population, N.
   (j) Replace the parent chromosome population with the offspring population.
3. Rank the individual chromosomes according to RC. Return the highest fitness chromosome.

Fig. 3. Algorithm EAFS.

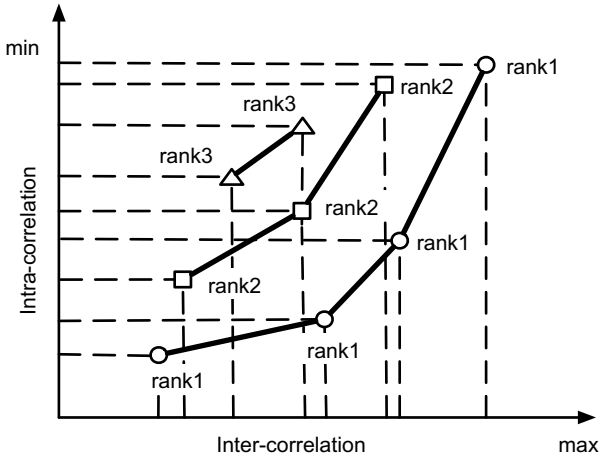

Fig. 4. Non-dominated sort.

rejected). The data sets before and after pre-processing are summarized in Tables 2 and 3.

4.2. Classifiers

In the paper, we compared the relative performance of SVM with several other classifiers, such as decision stump, naive Bayes, nearest neighbor (k = 3, 5), back-propagation neural networks, C4.5 decision tree, perceptron, and linear discriminant analysis (LDA). All the performance results of classifiers were obtained through five-fold cross validation to minimize the impacts of data dependency and prevent the over-fitting problem.

4.2.1. Support vector machines

Support vector machines (SVMs) are a type of new and promising learning machines, much like the famous perceptron algorithm (Cortes & Vapnik, 1995). The main characteristics of SVMs is to

Table 2
Data summary before pre-processing

Data sets	# Classes	# Instances	Nominal features	Numeric features
Australian	2	307/383	6	8
Germany	2	700/300	7	13

Table 3
Data summary after pre-processing

Data sets	# Classes	# Instances	Nominal features	Numeric features
Australian	2	307/383	0	42
Germany	2	700/300	0	60

classify unlabeled input samples by using linear or nonlinear kernel functions that implicitly map input spaces into high-dimensional feature spaces, as shown in Fig. 5. The superior classification performances and good generalization capabilities of SVMs have attracted much attention in the past ten years. They also have been applied successfully to a wide variety of application domains including pattern recognition, text categorization, and fraud detection in recent years (Cristianini & Shawe-Taylor, 2000; Hsu & Lin, 2002).

Given training vectors $x_k \in R^n$, $k = 1, \ldots, m$ in two classes, and a vector labels $y \in R^m$ such that $y_k \in \{1, -1\}$, SVM solves a quadratic problem

$$\min_{\{\mathbf{w},b,\xi\}} \quad \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{k=1}^{m}\xi_k$$

$$\text{subject to} \quad y_k(\mathbf{w}^{\mathrm{T}}\phi(x_k)) + b \geqslant 1 - \xi_k, \quad \xi_k \geqslant 0, k = 1, \ldots, m \qquad (8)$$

where training data are mapped to a high-dimensional space by the function ϕ, and C is a penalty parameter of the training error (Hsu et al., 2003).

4.2.2. Parameters Optimization of SVMs

The RBF kernel is used in the experiments

$$k(x, x') = \exp(-\gamma\|x - x'\|^2) \qquad (9)$$

With the RBF kernel, there are two parameters to be determined in the SVM model; i.e., C and γ. In the paper, we conducted a uniform design (UD) methodology proposed by Huang, Lee, Lin, and Huang (2004) to select the parameters. The procedure is similar to the conventional exhaustive grid method. However, it can dramatically cut down the number of parameter trials and also provide reliable solutions. The key idea is to use a suitable UD table to accommodate the number of parameters and levels. Then, the run order of experiments could be determined randomly by the UD table. Finally, each parameter combination in the UD is evaluated by performance estimators. The automatic UD-based model selection function called 'hibiscus' in the SSVM toolbox is available at http://dmlab.csie.ntust.edu.tw/downloads/.

4.2.3. Back-propagation neural networks

In the paper, a three-layer feed-forward network consisting of input units, hidden neurons, and only one output neuron, is optimized to classify the credit data. The number of input units is the same as the number of input attributes of the credit scoring data, and the number of hidden neurons is half of the number of input attributes. All of the input units are connected to each hidden unit, and every hidden unit is connected to the output unit.

All weights are randomly initialized to a number near zero, and then updated by the back-propagation algorithm. The back-propagation algorithm contains two phases: forward phase and backward phase. In the forward phase, we compute the output values of each layer unit using the weights on the arcs. In the backward phase, we update the weights on the arcs by a gradient descent method to minimize the squared error between the network values and the target values.

4.3. General heuristics

For the evolutionary algorithm described in Sections 3.3 and 3.4, the specific parameters for the data set are as follows: popula-

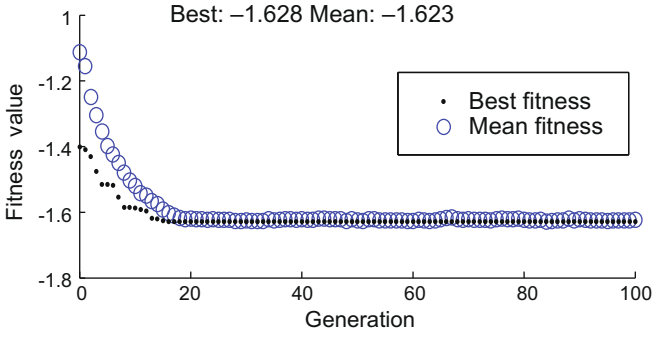

Fig. 6. Learning curve of algorithm EAFS.

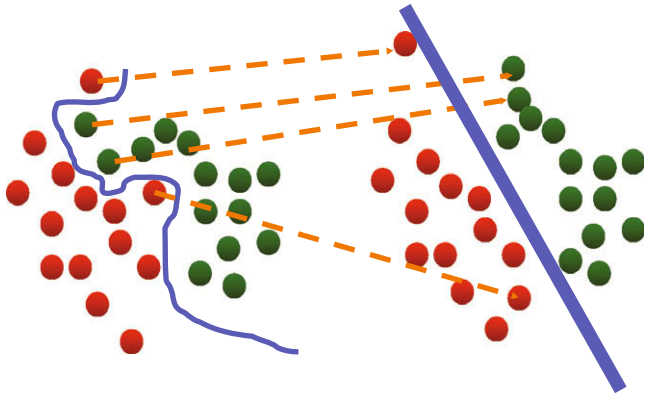

Fig. 5. Schematic example of SVMs.

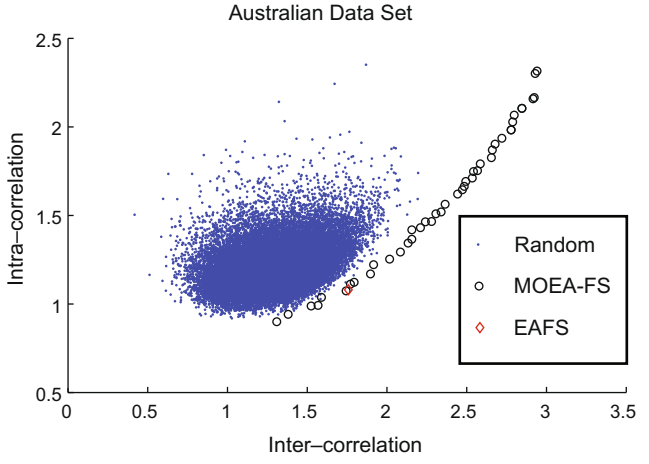

Fig. 7. Pareto front of Australian data set.

tion size is 50, reproduction rate is 0.9, crossover rate is 0.9, mutation rate is 0.01, and maximum number of generations is 100 and 1000, respectively. For the back-propagation neural network, we set the maximum run, learning rate and momentum to 1000, 0.9, and 0.2, respectively. For the other classifiers, we choose their default settings.

4.4. Performance evaluation measures

Any single performance estimator suffers the risk of being fitted if we compare many classifiers based on the estimators (Lin & Li, 2005). Thus, we carefully used five measures to evaluate the performance, which are defined as follows:

$$error = \frac{fp + fn}{tp + tn + fp + fn} \tag{10}$$

$$precision = \frac{fp}{tp + fp} \tag{11}$$

$$recall = \frac{fp}{tp + fn} \tag{12}$$

$$f_1 = \frac{2 \times recall \times precision}{recall + precision} \tag{13}$$

where tp is the number of true positives, fp is the number of the false positives, tn is the number of true negatives, and fn is the number of false negatives, respectively.

Error is one of the most used empirical measures that estimate the overall misclassified instances (misclassified as true and misclassified as false) over all instances. *Precision* is a function of the correctly classified examples (true positives) and the misclassified examples (false positives). *Recall* is a function of true positives and false negatives. F_1 measure is an evenly balanced precision and recall. The last three measures distinguish the correct classification of different classes.

Additionally, the area under the ROC curve or simply AUC is also used in this work. The ROC (receiver operating characteristics) curve shows a classifier's performance across the entire range of class distributions and error costs (Ling, Huang, & Zhang, 2003). Besides, AUC provides a single-number summary for the performance of a learning algorithm.

Table 4
Error of Australian data set

	χ^2	EAFS	MOEA	None	Relief
BP	0.161	0.049	**0.009**	0.19	0.028
DS	0.145	0.085	**0.004**	0.145	0.031
DT	0.155	0.155	**0.004**	0.155	0.023
LDA	0.141	0.053	**0.004**	0.138	0.031
3NN	0.205	**0.097**	**0.097**	0.220	0.156
5NN	0.193	0.127	**0.099**	0.212	0.162
NB	0.206	0.045	**0.010**	0.206	0.081
Preceptron	0.266	0.058	**0.013**	0.168	0.068
SVM	0.199	0.133	**0.028**	0.188	0.151

4.5. Experimental results

Although we run all procedures for Australian and Germany credit scoring data sets, respectively, we only present some results here for the limit of paragraph spacing.

As shown in Fig. 6, we plotted the learning curve of the best and average values of the fitness function across 1000 generations of the evolutionary algorithm described in Fig. 3. The Australian data

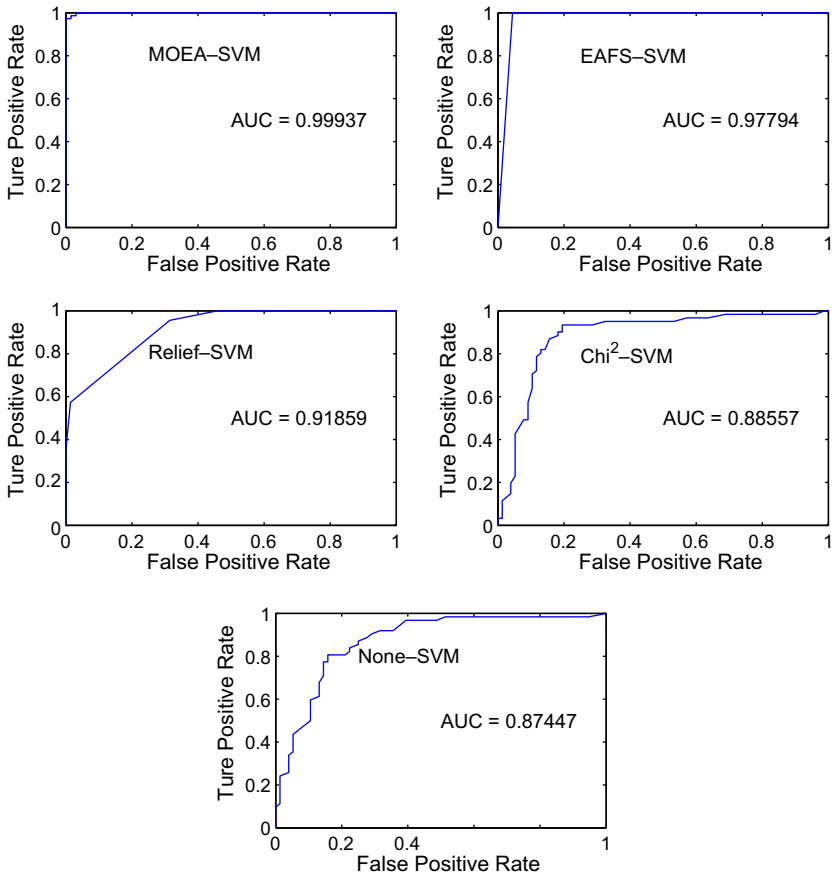

Fig. 8. ROC curve of Australian data set.

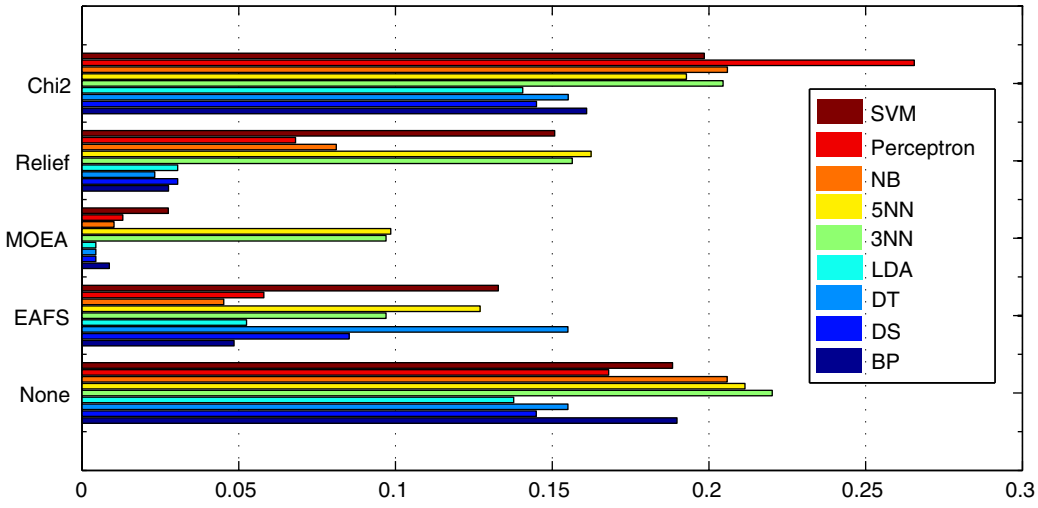

Fig. 9. Error of Australian data set.

set is analyzed in the following figures. Since the class distribution is nearly balanced, the re-sampling procedure is not activated. The fitness value is decreased rapidly at the beginning of the run, but then as the population converges on the nearly optimal solution, it decreases more slowly (after run 20) and settles down finally. Here, we used a 'minus' relative correlation (see Eq. (7)) as the objective function in EAFS.

As shown in Fig. 7, the Pareto front is obtained with three feature selection techniques. The random feature selection is a trivial method and used merely as a reference. Obviously, MOEA can find a good diversity of solution distributions, and also converges on better results than the random search. Besides, although the single-objective EAFS has found a good solution near the Pareto front, it might have difficulties in finding a good spread of solutions near the Pareto optimal front.

As shown in Fig. 8, we plotted the ROC curves for the Australian data set, which reveal the performances of different SVMs without regard to class distributions or error costs. Clearly, two evolutionary feature selection methods outperform the others in the run. From Table 4 to Table 8, the feature selection methods are evaluated on different classifiers and measures. The winning method of each classifier is marked in bold face. The measures in the tables are better when they have higher values, except Table 4 (i.e., the overall error estimator). The multi-objective feature selection method wins 27 runs out of 36. Obviously, it is the best one among these methods. By the way, the single-objective feature selection method also has good performances in some runs.

From the perspective of classifies, although there is no consistent winner, the nearest neighbor classifiers usually reveal the worst results. As shown in Fig. 9 reproduced from Table 4, the horizontal histogram reveals the relative performance of the classifiers with different feature selection methods. There, multi-objective feature selection method significantly outperforms the other methods under 95% confidence interval paired t-test (Tables 5–7).

Table 5
AUC of Australian data set

	χ^2	EAFS	MOEA	None	Relief
BP	0.903	0.993	**0.999**	0.889	0.994
DS	0.862	0.980	**0.996**	0.862	0.970
DT	0.854	0.971	**0.996**	0.858	0.975
LDA	0.925	0.993	**0.100**	0.920	0.994
3NN	**0.561**	0.546	0.556	0.546	0.541
5NN	**0.561**	0.546	0.556	0.546	0.541
NB	0.888	0.982	**0.988**	0.890	0.978
Preceptron	0.847	0.986	**0.990**	0.890	0.988
SVM	0.850	**0.984**	0.992	0.881	0.971

Table 6
Precision of Australian data set

	χ^2	EAFS	MOEA	None	Relief
BP	0.829	0.963	**0.990**	0.786	0.968
DS	0.788	0.964	**1.000**	0.787	0.945
DT	0.815	0.964	**1.000**	0.838	0.963
LDA	0.799	0.964	**1.000**	0.802	0.945
3NN	0.797	**0.922**	0.913	0.775	0.876
5NN	0.823	**0.920**	0.909	0.801	0.880
NB	0.858	0.954	**0.995**	0.868	0.959
Preceptron	0.666	0.963	0.984	0.774	**0.988**
SVM	0.786	0.960	**0.970**	0.808	0.906

Table 7
Recall of Australian data set

	χ^2	EAFS	MOEA	None	Relief
BP	0.804	0.968	**0.995**	0.788	0.977
DS	0.925	**0.997**	0.992	0.925	0.997
DT	0.847	0.988	**0.992**	0.811	0.991
LDA	0.919	**0.997**	0.992	0.919	**0.997**
3NN	0.736	**0.911**	**0.904**	0.713	0.798
5NN	0.727	0.900	**0.904**	0.700	0.781
NB	0.645	0.945	**0.986**	0.635	0.874
Preceptron	0.909	0.966	**0.992**	0.880	0.874
SVM	0.769	0.963	0.981	0.759	0.800

Table 8
F1 measure of Australian data set

	χ^2	EAFS	MOEA	None	Relief
BP	0.817	0.966	**0.992**	0.787	0.972
DS	0.851	0.980	**0.996**	0.850	0.970
DT	0.829	0.976	**0.996**	0.822	0.977
LDA	0.854	0.980	**0.996**	0.856	0.970
3NN	0.764	**0.916**	0.908	0.743	0.835
5NN	0.771	**0.909**	0.906	0.747	0.826
NB	0.736	0.949	**0.990**	0.733	0.914
Preceptron	0.760	0.964	**0.988**	0.823	0.927
SVM	0.775	0.961	**0.975**	0.780	0.799

5. Conclusions

In the paper, we proposed new criteria for single and multi-objective evolutionary feature selection algorithms. Comprehensive experiments were conducted to show the relative performance of the classification tasks. It was observed that the evolutionary-based feature selection with new criteria outperforms other techniques in finding a large and important part of the Pareto front in the feature selection problem.

Since data pre-processing and feature selection are important steps in the knowledge discovery process, the further work will apply these techniques with other classifiers to larges scale problems. Moreover, new multi-objective evolutionary algorithms should be considered.

References

Asuncion, A., & Newman, D. J. (2007). UCI machine learning repository. [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

Cantu-Paz, E., Newsam, S., & Kamath, C. (2004). *Feature selection in scientific applications. KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM Press.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press.

Kalyanmoy, D., & Kalyanmoy, D. (2001). *Multi-objective optimization using evolutionary algorithms*. New York, NY, USA: Wiley.

Deb, K., Agrawal, S., Pratab, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, & H.-P. Schwefel (Eds.). *Proceedings of the parallel problem solving from nature VI conference* (pp. 849–858). Paris, France: Springer. Lecture Notes in Computer Science No. 1917.

Desai, V. S., Crook, J. N., & Overstreet, G. A. Jr., (1996). A comparison of neural network and linear scoring models in credit union environment. *European Journal of Operational Research*, 24–35.

Drummond, C., & Holte, R. (2003). C4.5, Class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II (within ICML)*.

Hsu, C. W., Chang, C. C., & Lin, C. J., 2003. A practical guide to support vector classification. *Technical report*. Taipei: Department of Computer Science and Information Engineering, National Taiwan University.

Hsu, C.-W., & Lin, C.-J. (2002). A simple decomposition method for support vector machines. *Machine Learning, 46*, 291–314.

Huang, C.-M., Lee, Y-J., Lin, D. K. J., & Huang, S.-Y. (2004). Model selection for support vector machines via uniform design. A special issue on Machine Learning and Robust Data Mining of Computational Statistics and Data Analysis.

Huang, S. H. (2003). Dimensionality reduction in automatic knowledge acquisition: A simple greedy search approach. *IEEE Transactions on Knowledge and Data Engineering, 15*(6), 1364–1373.

Kira, K., & Rendell, L. A. (1992). The feature selection problem: Traditional methods and a new algorithm. *AAAI*, 129–134.

Lin, H.-T., & Li, L., 2005. Analysis of SAGE results with combined learning techniques. In P. Berka, & B. Crémilleux (Eds.), *Proceedings of the ECML/PKDD 2005 discovery challenge* (pp. 102–113).

Ling, C., Huang, J., & Zhang, H., 2003. AUC: A statistically consistent and more discriminating measure than accuracy. In *Proceedings of international joint conferences on artificial intelligence*.

Montgomery, D. C., & Runger, G. C. (2006). *Applied statistics and probability for engineers*. Wiley.

Thomas, L. C. (2000). A survey of credit and behavioral scoring: forecasting financial risk of lending to consumers. *International Journal of Forecasting, 16*(2), 149–172.

Weiss, G., & Provost, F. (2001). The effect of class distribution on classifier learning. *Technical report ML-TR 43*. Department of Computer Science, Rutgers University.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques* (2nd ed.). Morgan Kaufman.