Engineering Applications of Artificial Intelligence 26 (2013) 848–862

Contents lists available at SciVerse ScienceDirect

Engineering Applications of Artificial Intelligence

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

Orthogonal support vector machine for credit scoring

Lu Han [a,b,*], Liyan Han [a], Hongwei Zhao [b]

[a] School of Economics and Management, Beihang University, Beijing 100191, China
[b] PBC School of Finance, Tsinghua University, Beijing 100083, China

ARTICLE INFO

Article history:
Received 8 December 2011
Received in revised form
25 September 2012
Accepted 8 October 2012
Available online 17 November 2012

Keywords:
Dimension curse
Orthogonal dimension reduction
Support vector machine
Logistic regression
Principal component analysis
Credit scoring

ABSTRACT

The most commonly used techniques for credit scoring is logistic regression, and more recent research has proposed that the support vector machine is a more effective method. However, both logistic regression and support vector machine suffers from curse of dimension. In this paper, we introduce a new way to address this problem which is defined as orthogonal dimension reduction. We discuss the related properties of this method in detail and test it against other common statistical approaches—principal component analysis and hybridizing logistic regression to better solve and evaluate the data. With experiments on German data set, there is also an interesting phenomenon with respect to the use of support vector machine, which we define as 'Dimensional interference', and discuss in general. Based on the results of cross-validation, it can be found that through the use of logistic regression filtering the dummy variables and orthogonal extracting feature, the support vector machine not only reduces complexity and accelerates convergence, but also achieves better performance.

1. Introduction

Credit risk based on the characteristics of the debtor is often divided into sovereign, corporate, retail, etc. Retail debt is centered on customer credit, which includes short-term and intermediate-term credit to finance the purchase of commodities and services for consumption or to refinance debt incurred for such purposes. Retail credit is characterized by three points: first, large amounts with small scale. At present in China, retail loans can account for a quarter of the total debt, with a speed of growth approaching 10%; second, the potential risk is high but the information is scattered and complicated. In the loan application form there are thousands of variables to describe and, even worse, is that different organizations always use different variables; and third, the efficiency of business processing requires highly developed decision-making techniques as competition is getting more and more intense. These characteristics determine the banks need to implement risk management evaluation methods based on quantitative analysis. A good credit risk evaluation tool can help to grant credit to more creditworthy applicants and thus increases profit. Moreover, it can deny credit for the noncredit worthy applicants and thus decreases losses.

Currently, credit scoring has become the primary method to develop a credit risk assessment tool. It is a method to evaluate the credit risk of loan applicants with their corresponding credit score that is obtained from a credit scoring model (Altman, 1998). A credit score is a number that can represent the creditworthiness of an applicant and it is based on the analysis of an applicant's characteristics from the application file using the credit scoring model. The credit scoring model (Thomas et al., 2002) is developed on the basis of historical data about the applicant's performance on previously made loans with the use of some quantitative techniques, such as statistics analysis, mathematical programming, artificial intelligence and data mining. A well-designed model should have higher classification accuracy to classify the new applicants or existing customers as good or bad and the model is the core of credit scoring.

The most popular methods adopted in credit scoring are statistical methods. The statistical principle discriminating different groups in a population can be traced back to 1936 in Fisher (1936) publication which used a linear model to calculate the distance between two classes as the decision factor. It is known as the Fisher's discrimination model. In 1977, Martin (1977) first introduced the logistic regression method to the bank crisis early warning classification. Martin chose to use data between 1970 and 1976, with 105 bankrupt companies and 2058 non-bankrupt companies in the matching sample, and analyzed the bankruptcy probability interval distribution, with two types of errors and the relationship between the split points, he then found that size, capital structure, and performance were key indexes for the judgment. Martin determined that the accuracy rate of the overall classification could reach 96.12%. Logistic regression analysis had

* Corresponding author. Tel.: +86 1861 166 7963.
E-mail addresses: hanluivy@126.com (L. Han), hanly@buaa.edu.cn (L.Y. Han), hongwei_zhao@yeah.net (H.W. Zhao).

0952-1976/$ - see front matter Crown Copyright © 2012 Published by Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2012.10.005

significant improvements over discriminant analysis with respect to the problem of classification. Martin also noted that logistic regression could overcome many of the issues with discriminant analysis, including the assumption that variables must be normally distributed. Wiginton (1980), was one of the first researchers to report credit scoring results with the logistic regression model. Although the result was not very impressive, the model was simple and could be illustrated easily. Then, at that point the logistic regression model had become the main approach for the practical credit scoring application. In 1997, Hand and Henley (1997) summarized statistical methods in credit scoring. These methods are relatively easy to implement and are able to generate straightforward results that can be readily interpreted. Nonetheless, as commonly known, there are also quite a few limitations associated with the applications of these statistical methods. First of all, they have the fatal problem called 'Curse of dimension' which suggests that if there are numerous variables to apply, because of multicollinearity between variables, the results are always erroneous and misleading. Therefore, before applying statistical methods, the process entailed tremendous data preprocessing efforts through variable selection. This strategy usually requires domain expert knowledge and an in-depth understanding of the data. In addition, all the statistical models are based on a hypothesis condition. In a real world application, a hypothesis such as that the dependent variable should follow logic normal distribution and so on, may not hold. Most importantly, based on these algorithms, these statistical models have difficulty in the automation of modeling processes and lack robustness. When environmental or population changes occur, the static models usually fail to adapt and need to be rebuilt again.

In response to the concern for classification accuracy in retail loans applications, researchers discovered the application of the support vector machine (SVM). The support vector machines (SVM) approach was first proposed by Cortes and Vapnik (1995). The main idea of SVM is to minimize the upper bound of the generalization error. SVM usually maps the input variables into a high-dimensional feature space through some nonlinear mapping. In that space, an optimal separating hyper plane, which is one that separates the data with the maximal margin, is constructed by solving a constrained quadratic optimization problem. Suykens et al. (2002) constructed the least squares support vector machine(LS-SVM) and used it for the credit rating of banks and reported the experimental results compared with ordinary least squares (OLS), ordinary logistic regression (OLR) and the multilayer perceptron (MLP). The result showed that the accuracy of the LS-SVM classifier was better than the other three methods. Schebesch and Stecking (2005) used a type of standard SVM proposed by Vanik with a linear and radial basis function (RBF) kernel for dividing credit applicants into subsets of 'typical' and 'critical' patterns which can be used for rejecting applicants. Schebesch and Stecking concluded these types of SVM should be widely used because of their performance. Gestel et al. (2003) discussed a benchmark study of seventeen different classification techniques on eight different real-life credit datasets. They used SVM and LS-SVM with linear and RBF kernels and adopted a grid search mechanism to tune the hyper parameters in their study. The experimental results indicated that six different methods were the best in terms of classification accuracy among the eight datasets — linear regression, logistic regression, linear programming, classification tree, neural networks and SVM. In addition, the experiments showed that the SVM classifiers can overall yield the best performance. Yang (2007) experimented with several kernel learning methods to apply adaptive credit scoring, and found that the results can be very impressive when using the SVM. Nevertheless the existing research findings have all focused on batch learning and the selection of parameters, as seen in the work of Yu et al. (2006,2008) which shows SVM's advantages in solving high dimensional problems. However, there are two obvious drawbacks to SVM (Min and Lee, 2005). One is that when the variables are not 'meaningful' and 'huge', SVM requires a long time to train and the hyper plane is not accurate, which we also define as curse of dimension. The drawback is a fatal flaw, although this method has good robustness and can always achieve higher accuracy, when applied to samples, SVM lacks the capability to explain its results. That is, the results obtained from SVM classifiers are not intuitive to humans and are difficult to illustrate comparing with logistic regression. This is a common problem that all machine learning methods are facing. Though the results with these methods have strong advantages in accuracy, the non-parameter results often lack of statistical theory, and so which cannot be directly corresponding to the realistic economic significance. Just as in regression analysis, regression coefficient directly represents the influence of independent variable acting on dependent variable, but in support vector machine (SVM), the relationship between independent variable and dependent variable cannot be explained directly. So this limits these methods in practical application, and at the same time this also is a cause for over fitting phenomenon.

Dimension curse (Anderson, 1962) can be defined as this phenomenon: as the number of variables increase, more and more variables will have multicollinearity, which can be described as when the correlation coefficient gets large, and is in a high dimensional space, the distribution of the sample points will become sparse. Statistical methods will prove to be erroneous with multicollinearity, and SVM will need a large amount of support vectors to construct hyper plane. Now, to solve the curse of dimensionality, researchers often use two methods to reduce variables. One method is feature selection, another is feature extraction. Feature selection is to select important variables closely related with the target in order to reduce the model's dimensions; feature extraction is to construct new variables which are not linearly dependent through structure transformation. The drawback of feature selection is in reducing information and the advantage is that it is easy to explain. Feature extraction is just the opposite. Many scholars have performed a lot of work to reduce dimensions. Sugiyama (2007) tried feature selection to reduce dimensions in Fisher discriminant analysis. Bellman (1961) is the first to note the curse of dimension in kernel classifiers. He stipulated that owing to the large amounts of data from public financial statements that can be used for bankruptcy predictions, the large scale of input data makes Kernel classifiers infeasible due to the curse of dimensionality. Consequently, one needs to transform the input data space to a suitable low dimensional subspace that optimally represents the data structure. In the studies of Huang (2009), he discussed the use of a nonlinear graph as a type of method for feature selection to reduce dimension. Han and Han (2010) have tried logistic regression to select meaningful variables for neural networks. The other methods regarding dimensionality reduction, linear algorithms such as principal component analysis and linear discriminant analysis, are the two most widely used methods, which can be found in the works of Gutierrez et al. (2010) and Hua et al. (2007).

Just based on the studies above, we want to improve the accuracy of credit scoring through dimension reduction. Our novel contribution is that we give these researchers in the field of application using logistic regression and support vector machine a new way to address dimension curse that we defined as 'Orthogonal dimension reduction' (ORD). Based on the experience of statistics, we compare the traditional way to address dimension curse—hybridizing with logistic regression (HLR) (Fukunaga, 1990) on behalf of feature selection and principal

component analysis (PCA) (Jolliffe, 2002) on behalf of feature extraction. Then, we fit these helpful features chosen by ORD, HLR and PCA in logistic regression and support vector machine to evaluate the accuracy of credit scoring. Furthermore, we compare these results with the original methods without reducing dimension. Finally, we acquire cross-validation to make the results sensitive.

The structure of the rest in this paper is as follows: the next section puts forward the prior research of logistic regression and support vector machine. Section 3 briefly summarizes the previous methods of reducing dimension. Section 4 describes our method of orthogonal dimension reduction and its main principles in detail. Section 5 is about experiment design, including data and variable description, data pre-processing, evolutionary learning for SVM, features selection with HLR, features extraction with PCA and ODR, cross-validation design, and accuracy criterion. Experimental studies using the original methods and the methods hybridizing dimension reduction are presented in Section 6. The Final section discusses the interesting results and gives some remarks.

2. Prior research

Let $X=(x_1,x_2,\cdots x_n)^T$ be a set of n random variables which describe the information from a customer's application form and credit reference bureau. The actual value of the variables for a particular applicant i is denoted by $X_i=(x_{1i},x_{2i},\cdots x_{ni})^T$. All samples denoted by $S=(X_i,y_i),i=1,2,\cdots,N$, where N is the number of samples, X_i is the attribute vector of the *ith* customer, and y_i is its corresponding observed result of timely repayment. If the customer is good, $y_i=1$, else $y_i=-1$. Let $I=\{i|y_i=1,i\in N,(x_i,y_i)\in S\}$ is on behalf of good customers, $J=\{i|y_i=-1,i\in N,(x_i,y_i)\in S\}$ is on behalf of bad ones.

Though in practice a credit scoring result needs the score of each applicant, in fact our greatest concern is the accuracy of the distinction between categories. Thus, the credit scoring problem can be described simply as making a classification of good or bad for a certain customer using the attribute characteristics of a certain customer. That is, using the attribute vector X_k, one can judge the credit status. The typical credit risk modeling techniques, which were tested in this paper, are briefly described below.

2.1. Logistic regression

Just as linear regression, logistic regression assumes that the sum of the weighted input variables is linearly correlated to the natural log of the odds that the outcome event will happen. It can be described as (1):

$$\log(p/1-p)=\beta_1x_1+\beta_2x_2+\ldots+\beta_kx_k+e=\beta^TX_k+e \tag{1}$$

where $\beta=(\beta_1,\beta_2,\ldots\ldots,\beta_k)$ is the vector of the coefficients of the model, the maximum likelihood method can be applied to compute the estimate of $\beta_i\ \{i=1,2\cdots k\}$. We refer to $p/(1-p)$ as odds-ratio and assume the regression model in (1) is obtained, the estimated probability of no default is as follows:

$$p=\frac{e^{\beta^Tx}}{1+e^{\beta^Tx}} \tag{2}$$

Linear regression is based on the idea of using vector X to explain y logistic regression is the same, using X to explain natural log of the odds, so just like linear regression, it has good interpretations in statistical sense. But logistic regression can overcome the flaw of linear regression, which is that the right side of the model could take any value from $-\infty$ to $+\infty$ but the left side can only take values between 0 and 1.

The method has two shortcomings: one is that it can only explain the intrinsically linear relationship, and cannot address non-linear effects in practice. Researchers always explain any non-linear effects with variable combinations and this requires several repetitions of a trial-and-error process. In addition, the method is sensitive to redundancy or collinearity in the input variables to guarantee the basic assumption of e, which is $e_i\sim NID(0,\sigma^2)$, which therefore requires that y obey logic normal distribution. If this condition is not satisfied, this method will give erroneous estimates of the coefficients and is not valid for statistical interpretation.

2.2. Support vector machine

The main idea of support vector machine is to minimize the upper bound of the generalization error not the empirical error. Without loss of generality, in a two-dimensional space, if these scoring samples are linear separable, The upper bound can be constructed by $(wx)+b=1$ and $(wx)+b=-1$, so a decision function can be created to specify whether a given application belongs to either I or J. Its definition is as follow: $f(x)=sign((wx)+b)$. While the vector w defines the boundary, in order to get the two upper bound separated as far as possible, the optimal hyperplane can be obtained as a solution to the optimization problem:

$$\begin{aligned}&\max\tfrac{2}{\|w\|^2}\\&s.t.\\&y_i((w\cdot x_i)+b)\geq 1\ \ i=1,2,\cdots n\end{aligned} \tag{3}$$

which could be written as (4):

$$\begin{aligned}&\min\tfrac{1}{2}\|w\|^2\\&s.t.\\&y_i((w\cdot x_i)+b)\geq 1\ \ i=1,2,\cdots n\end{aligned} \tag{4}$$

So an optimal separating hyperplane which is one that separates the data with the maximal margin is constructed by solving a constrained quadratic optimization problem whose solution has an expansion in terms of a subset of training patterns that lie closet to the boundary, and this subset of patterns are called as support vector (SV). Fig. 1 shows such a hyperplane that separate two classes to the boundary.

In many practical situation, the training samples cannot be linear separable. There is a need to use soft margin and C penalty parameters, this is formulized as the following constraint optimization problem:

$$\begin{aligned}&\min J(w,b,\xi_k)=\tfrac{1}{2}\|w\|^2+C\textstyle\sum_k\xi_k\\&s.t.\\&y_i((w\cdot x_i)+b)\geq 1\ \ i=1,2,\ldots n\xi_k\geq 0\end{aligned} \tag{5}$$

where C is the corresponding penalty parameters indicating a tradeoff between large margin and a small number of margin

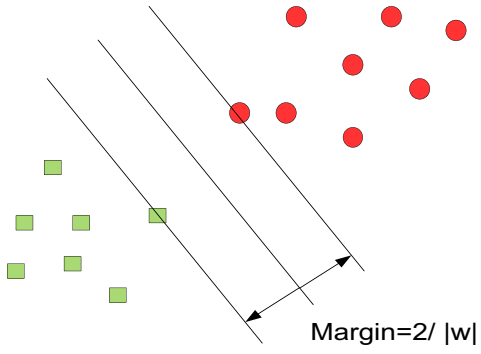

Fig. 1. Separating hyperplane for two separable classes.

failures: few errors are permitted for high C, while low C allows a higher proportion of errors in the solution. The solution to this optimization problem can be given by the saddle point of the Lagrange function with Lagrange multipliers α_i, and then the problem can be transformed into its dual form:

$$\begin{aligned} &\max J(\alpha) = -\tfrac{1}{2}\textstyle\sum_k\sum_l \alpha_k\alpha_l y_k y_l \langle x_k \cdot x_l \rangle + \sum_k \alpha_k \\ &s.t. \\ &\textstyle\sum_k \alpha_k y_k = 0,\ \begin{cases} y_k = 1, k \in I \\ y_k = -1, k \in J \end{cases} 0 \le \alpha_k \le C,\ \forall k \end{aligned} \tag{6}$$

In cases where the linear boundary in input spaces is not be able to separate the two classes accurately, a hyperplane is created that allows linear separation in the higher dimension by the use of transformation function $\varphi(\cdot)$ which can map the input space into a higher dimensional feature space (z-space), then the objective function can be rewritten as:

$$\max J(\alpha) = -\frac{1}{2}\sum_k\sum_l \alpha_k\alpha_l y_k y_l \langle \varphi(x_k)\varphi(x_l) \rangle + \sum_k \alpha_k \tag{7}$$

But using this way to transformation is relatively computation-intensive. And we can find $\varphi(\cdot)$ only uses for inner product, therefore a kernel can be used to perform this transformation and then inner product can be replaced by kernel function which is given by Mercer's theorem. The kernel function is defined as (8):

$$K(x,y) = \varphi(x)\varphi(y) \tag{8}$$

The most common kernel functions are listed below.

(1) Linear kernel function: $K(x_k,x) = x_k^T x$
(2) Polynomial function: $K(x_k,x) = \left(x_k^T x + 1\right)^d$
(3) Gaussian function: $K(x_k,x) = \exp\left(-\|x_k - x\|^2/\sigma^2\right)$
(4) Radial basis kernel: $K(x_k,x) = \exp\left(-\|x_k - x\|^2/2\sigma^2\right)$

Then the objective function can be rewritten as:

$$\begin{aligned} &\max J(\alpha) = -\tfrac{1}{2}\textstyle\sum_k\sum_l \alpha_k\alpha_l y_k y_l K(x_k,x_l) + \sum_k \alpha_k \\ &s.t. \\ &\textstyle\sum_k \alpha_k y_k = 0, \begin{cases} y_k = 1, k \in I \\ y_k = -1, k \in J \end{cases} 0 \le \alpha_k \le C, \forall k \end{aligned} \tag{9}$$

In some perspectives, the support vector machines and artificial neural networks are similar; this can be illustrated in the following Fig. 2. SVM can be seen as a type of networks which uses the kernel function as an activation function, and the optimization program as a threshold function. So to control the generalization capability of SVM, there are a few parameters such as C and kernel parameters that need to be trained. But there is not a most effective method up to now, a number of approaches have been presented including Genetic algorithms (Jack and Nandi, 2002), Artificial Immunization Algorithm (Yuan and Chu, 2007), Particle Swarm Optimization (Samanta and Nataraj, 2009) and so on. In our work, the parameters choice is not the focus of our research, we give a general method in Section 4.

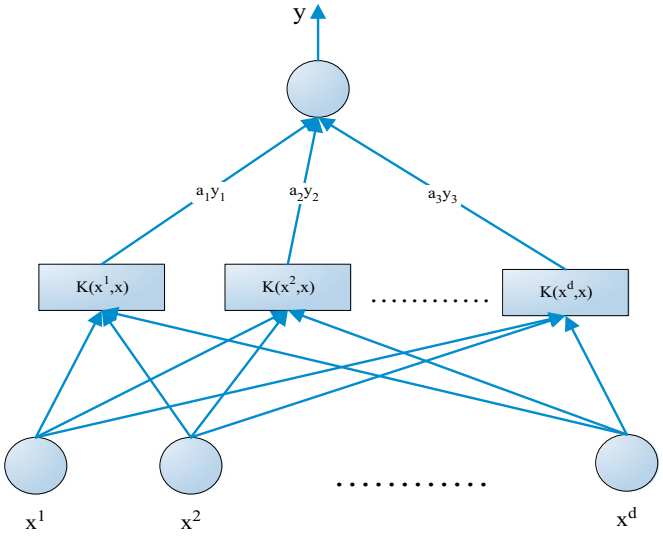

Fig. 2. Structure of SVM.

3. Reducing dimension

3.1. Feature selection

The task of feature selection is to examine characteristics which are contained in the input variables, and then delete those that are irrelevant to the target variables. There are many statistical methods such as variance analysis (ANOVA) and correlation analysis. However, they are all based on the conditions of the experimental data, for credit scoring, the most widely used method is hybridizing with logistic regression to do feature selection.

Hybridizing with logistic regression is based on the statistics' meaning. In these regression models, through the variance analysis, we can find the variable which can give the largest contribution to the variation of target variables, and we think these variables have the closest relationship to the target variable. As (1), $\hat{y}_i$ is the regression result, $\bar{y}$ is the mean of observation, y_i is an observation. The total fluctuations of data can be described as $S_T = \Sigma(y_i - \bar{y})^2$, $S_R = \Sigma(\hat{y}_i - \bar{y})^2$ measures explanatory power of $E(y)$ decided by X, and $S_e = \Sigma(y_i - \hat{y}_i)^2$ measures the difference between result of regression and observation. So $S_T = S_R + S_e$ and it can be proved $S_e/\sigma^2 \sim \chi^2(n-2)$; if $E(y)$ decided by X is true, $S_R/\sigma^2 \sim \chi^2(1)$; and S_e is independent with S_R. Therefore, through ANOVA we can get the useful variable.

3.2. Feature extraction

The most widely used method for feature extraction is principal component analysis. The principle of component analysis is to keep as much information as possible of the original variables, and to achieve dimension reduction, through the use of comprehensive new variables.

Let $u = E(X)$, $\Sigma = \mathrm{cov}(X) = E(xx')$ is the covariance matrix. To Apply linear transformation with X, we can construct the new variables Z as follow:

$$\begin{cases} z_1 = u_{11}x_1 + u_{12}x_2 + \cdots + u_{1n}x_n \\ z_2 = u_{21}x_1 + u_{22}x_2 + \cdots + u_{2n}x_n \\ \cdots\cdots \\ z_n = u_{n1}x_1 + u_{n2}x_2 + \cdots + u_{nn}x_n \end{cases} \tag{10}$$

PCA tries to sequentially find the projection $u_1, u_2, \ldots u_n$ (where $\|u_i\| = 1$) such that the variance of the projected data $z_i (i = 1,2,\ldots,n)$ is maximized:

$$var(z_i) = \mathrm{cov}(u_i'X) = u_i'\Sigma u_i \tag{11}$$

If Σ is estimated by its MLE, which is the sample covariance matrix S defined as (12):

$$S = \frac{1}{N}\sum_{k=1}^{N} x_k x_k' \tag{12}$$

Let $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n$ is eigenvalue of Σ and $r_1, r_2, \cdots r_n$ is the corresponding eigenvectors, based on spectrum decomposition, the *ith* principal component can be written as (13), with $\mathrm{var}(z_i) = r_i'\Sigma r_i = \lambda_i$ and $\mathrm{cov}(z_i, z_j) = r_i'\Sigma r_j = 0$

$$z_i = r_{1i}x_1 + r_{2i}x_2 + \cdots + r_{ni}x_n (i = 1,2,\ldots,n) \tag{13}$$

4. Orthogonal dimension reduction

In this section we give a new method to do feature extraction. The step of transform is described as (14):

$$\begin{aligned} &z_1 = x_1 \\ &z_2 = x_2 - \frac{x_2' z_1}{x_1' z_1} z_1 \\ &z_3 = x_3 - \frac{x_3' z_1}{x_1' z_1} z_1 - \frac{x_3' z_2}{x_2' z_2} z_2 \\ &\cdots\cdots \\ &z_s = x_s - \sum_{i=1}^{s-1} \frac{x_s' z_i}{x_i' z_i} z_i \end{aligned} \tag{14}$$

Theorem 1. : Any group of vectors $(x_1, \cdots, x_s)$ can transform into orthogonal vectors $(z_1, \cdots, z_s)$ by the process as (14).

The process in two dimensions space can be explained as Fig. 3. The proof can be found in the Ref. Jain and Gunawardena (2003).

From Theorem 1, we know that for any group of variables with the rank $s \le n$ to apply the orthogonal transform, and we get $z_1, z_2, \cdots, z_n$. Among them, there must be s variables orthogonal with each other, and the other $(p–s)$ variables are zero vector. Therefore, through this transform, it normally reduces the dimensions of the original high dimensional system.

Theorem 2. : If the variables $x_1, x_2, \cdots, x_n$ are standardized, after orthogonal transform, the variance $z_1, z_2, \cdots, z_s$ can get is $Var(z_k) = Var(x_k) - \Sigma_{j=1}^{k-1} r^2(z_j, x_k)$

Proof.

$$\begin{aligned} &\because var(x_k) = 1 \\ &\therefore \tfrac{1}{n}\|x_k\|^2 = 1 \\ &\therefore \|x_k\|^2 = n \\ &\because x_k = z_k + \sum_{j=1}^{k-1} \frac{x_k^T z_j}{\|z_j\|^2} z_j \\ &\langle z_k \cdot z_j \rangle = 0 \\ &\therefore \tfrac{1}{n}\|x_k\|^2 = \tfrac{1}{n}\|z_k\|^2 + \tfrac{1}{n}\sum_{j=1}^{k-1}\left[\frac{x_k^T z_j}{\|z_j\|^2}\right]^2 \|z_j\|^2 \\ &\therefore \tfrac{1}{n}\|x_k\|^2 = \tfrac{1}{n}\|z_k\|^2 + \sum_{j=1}^{k-1} r^2(z_j, x_k) \\ &\therefore \mathrm{var}(z_k) = \mathrm{var}(x_k) - \sum_{j=1}^{k-1} r^2(z_j, x_k) \end{aligned}$$

Information in data collection can be measured by the total variance of variables. From Theorem 2, we know that the reductions of these variables are redundancy information that can be measured by the correlation coefficient.

In short, through the orthogonal transform process we can reduce dimensions to realize feature extraction from two aspects: firstly, through the transform, one will find the orthogonal basis of a high dimensional space, which will help find whether a variable is a linear combination of other variables or not. If it is, the transform will turn it into 0 vectors. Thus, it is helpful to reduce the number of variables; furthermore, to those which are not 0 vectors, because of the subtraction of the correlation coefficient, the new feature can obtain the core original variables.

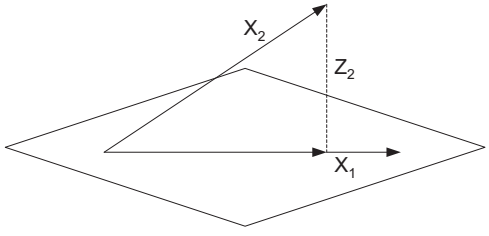

Fig. 3. Orthogonal transform.

The step to apply orthogonal dimension reduction is summarized below:

Step 1. Standardize the group of $x_1, x_2, \cdots, x_n$

Step 2. Choose z_1, which has the maximum correlation coefficient squares with the other variables, might as well written as $z_1 = x_1$

Step 3. Let $z_j^1 = x_j - (x_j' z_1 / x_1' z_1) z_1$, $j = 2, 3, \cdots, p$ calculate variance $\mathrm{var}(z_j^1)$ with the vectors $z_j^1, j = 2, \cdots, n$ which are not 0 vectors, find the one has the largest variance $\mathrm{var}(z_2) = \max \mathrm{var}(z_j^1)$, $j = 2, \cdots, n$ as z_2, written as $z_2 = z_2^1$

Step 4. Let $z_j^2 = x_j - ((x_j' z_1)/(z_1' z_1)) z_1 - ((x_j' z_2)/(z_2' z_2)) z_2$, $j = 3, 4, \ldots, m$, choose z_3 with the with the vectors which are not 0 vectors and $\mathrm{var}(z_3) = \max \mathrm{var}(z_j^2)$, $= 2, \cdots, m$

Step 5. Repeat the step until $TNP = (\Sigma_{i=1}^{s} Var(z_s) / \Sigma_{j=1}^{q} Var(z_j)) \ge \delta$, δ is settled with situations.

Step 6. End. The first s vectors are the features extracted from orthogonal dimension reduction (ODR).

5. Experiments design

This section is structured in six subsections. Firstly, we have a brief description of the dataset used in the experiments. And then we discuss the process of data pre-processing for modeling. In the third part, we discuss the methods for training parameters in SVM. The methods reducing dimension which are introduced in Sections 3 and 4 will be tested in the next subsections. Then, to test the robustness of the models, we design two cross-validation methodologies that are illustrated in the fifth subsection. The final subsection defines evaluation criteria.

5.1. Dataset description

The credit dataset used in these experiments is German credit dataset, which is provided by Professor Dr. Hans Hofmann of the University of Hamburg and is obtained from UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/databases/statlog/german/). The total number of instances is 1000 including 700 creditworthy cases and 300 default cases. There is no missing data.

For each applicant, 20 kinds of attribute are available; the variable names of these attributes used in the models are listed below with short names in brackets. There are 13 categorical attributes including status of existing checking account (checking), credit history (history), purpose for the credit (purpose), saving account (savings), present employment since (employed), personal status and sex (marital), other debtors/guarantors (coapp), property style (property), other installment plans (other), housing situation (housing), job status (job), telephone status (telephone), foreign worker or not (foreign) and 7 numerical attributes including duration in month (duration), credit amount (amount), installment rate in percentage of disposable income (installp), present residence since (resident), age in years (age), number of existing credits at this bank (existcr), number of people being liable to provide maintenance for (depends). Tables 1 and 2 below show the basic statistics and information of these attributes. Table 3 shows correlation matrix.

From these statistics of these attributes, we can see some attributes have relatively concentrated distribution, for example foreign and coapp, the modes get more than 90%. With the numerical attributes, the variable amount is more 'big' than others in the amount level. And because of concentrated level of categorical

Table 1
Statistics of categorical attributes.

Variable	Level	Missing	Mode	Mode (%)
Foreign	2	0	1	96.3
Coapp	3	0	1	90.7
Other	3	0	3	81.4
Housing	3	0	2	71.3
Job	4	0	3	63
Savings	5	0	1	60.3
Telephon	2	0	1	59.6
Marital	4	0	3	54.8
History	5	0	2	53
Checking	4	0	4	39.4
Employed	5	0	3	33.9
Property	4	0	3	33.2
Purpose	10	0	3	28

Table 2
Statistics of numerical attributes.

Variable	Means	STD	Missing	Min	Median	Max
Age	35.55	11.38	0	19	33	75
Amount	3271.26	2822.74	0	250	2319	18424
Depends	1.16	0.36	0	1	1	2
Duration	20.9	12.06	0	4	18	72
Existcr	1.41	0.58	0	1	1	4
Installp	2.97	1.12	0	1	3	4
Resident	2.85	1.1	0	1	3	4

they always have relevance with others which can be found in Table 3. So there may need some pre-processing before modeling.

5.2. Data pre-processing

To address the categorical attributes, the typical method is to code them with their levels as dummy variables. Then, it will reduce relevance. In our models we use the same method, so all of the categorical attributes are used with their levels. For example, foreign has two levels, so there will be two variables in the model—one is foreigh_0, and another is foreign_1. Other variables are the same. However, it can be seen that the way to address categorical attributes will cause the number of total variables to increase significantly.

To address the numerical attributes, there is no need to do normalization with the logistic regression model (Liu et al., 2012). Though the variable coefficient estimates β_i vary because of the unit, the correlation coefficient estimate R^2 and the model's results are not influenced, therefore it has no effect on the choice of variables. Because there is no proof as to the necessity of normalization with SVM, we cannot give a logistic reason for normalization, but in practice, we always normalize the numerical attributes in the models so that the results cannot be affected by the unit. To better demonstrate we also try the experiments with dimensionless numerical attributes and the ones which are not normalized, respectively, to compare in the models of SVM. The results will be illustrated in the fifth section. The method we use for normalization is $(x_i - mean)/std$, this method can transform any distribution of the variable into a standard normal distribution, which well achieves the variable distribution that the statistical models require. Fig. 4 shows the distributions of two main numerical variables in the data set: amount and age. It can be found that they are very different from normal distribution.

5.3. Select parameters for SVM

As introduced in the second part above, although SVM is a powerful learning method for classification problems, its performance is sensitive not only to the algorithm that solves the quadratic programming problem, but also to the parameters set in the SVM. In the process of using SVM, the first issue is how to discover the best parameter of SVM for a specified problem.

An easy and reliable approach is to determine a parameter range, and then make an exhaustive grid search over the parameter space to find the best setting (Rojas and Nandi 2006). In the grid search method, each point in the grid is defined by the grid range [(C_min, sigma_min), (C_max, sigma_max)] and a unit grid size (_C, _sigma) is evaluated by the objective function F. The point with the smallest F value corresponds to the optimal parameters. In the experiments, we used 750 samples and gird search to select t C and kernel parameters of SVM. To address the nonlinear effect, we choose a linear kernel function to compare with the logistic regression.

5.4. Reduce dimension

These categorical attributes in the previous section are all contained in dummy variables which avoid multi-linearity. However, the number of variables will be increasing. To reduce these types of variables, they must fit the feature selection. We apply logistic regression to this problem and set 15% as the significance level. In this paper, we refer to this method as HLG for short. Because logistic regression can give the model more meaningful interpretation for three reasons. First, logistic regression selects the most relevant variables to the target variables into the model. Second, in statistics logistic regression can use the Wald test to decide whether adding variables improves the unconstrained model. Third, by fitting variable selection with logistic regression we can exclude the multiple correlations among variables. Then, we follow these steps: firstly, we use all the dummy variables as inputs, which have been prepared with the methods introduced in Section 5.2. And secondly we address the variable selection process in logistic regression. The steps of selecting variables using the forward logistic regression are summarized in Table 4.

For these numerical attributes, after standardization, we experiment with PCA and ORD to compare.

The process of PCA is introduced in Section 3.2. The results of PCA are shown in Tables 5 and 6.

From Table 5, the first three components have contained 60% variance. In order to make comparable with ODR, we can reduce seven dimensions to three what can be obtained from the eigenvectors of correlation matrix. For example, $z_1 = 0.66du + 0.72am - 0.22in + 0.06re + 0.01age + 0.02ex + 0.03de$.

For ODR, the correlation matrix of numerical attributes is shown in Table 7 below. $\mathrm{Max}\Sigma_{i=1}^{7} r^2(x_j, x_i) = \Sigma_{i=1}^{7} r^2(x_2, x_i)$ so we choose $z_1 = x_2$, and then apply ODR with the other six variables, the variance of the six variables $z_i^1, i = 1,3,4,5,6,7$ listed in Table 8.

$\mathrm{Max}\mathrm{var}(z_i^1) = \mathrm{var}(z_6^1)$ so let $z_2 = z_6^1$, repeat this process till all the variables have realized orthogonalization.

Calculate net information percentage $NP = \Sigma_i Var(z_i)/\Sigma_i Var(x_i)$ and total net information percentage $TNP = \Sigma_{i=1}^{s} Var(z_s)/\Sigma_{j=1}^{q} Var(z_j)$, the results listed in Table 9 and the original variables are labeled in brackets. Fig. 5 shows the accumulative contribution of z_i.

From Fig. 5, we can see z_1, z_2, z_3 have 74% information of all, and has 66% variance which is smaller than the sum of our z_i. This means by ODR we can only choose the orthogonal variables z_1, z_2, z_3 instead of the seven numerical attributes.

5.5. Cross-validation

As the best model is tailored to fit one sub-sample, the model often estimates the true error rate too optimistically. Therefore, to get a true estimate of the error rate, we applied two types of cross-validation methodologies which were suggested by Zhang et al. (1999). First, as these typically did, the cross-validation

Table 3
Correlation matrix.

H0: Rho=0 Prob > |r|

	Checking	Duration	History	Purpose	Amount	Savings	Employed	Installp	Marital	Coapp	Resident	Property
Checking	1	−0.07201 0.0228	0.19219 <0.0001	0.02878 0.3632	−0.0427 0.1772	0.22287 <0.0001	0.10634 0.0008	−0.00528 0.8676	0.04326 0.1716	−0.12774 <0.0001	−0.04223 0.182	−0.03226 0.3081
Duration		1	−0.07719 0.0146	0.14749 <0.0001	0.62498 <0.0001	0.04766 0.132	0.05738 0.0697	0.07475 0.0181	0.01479 0.6404	−0.02449 0.4392	0.03407 0.2818	0.30397 <0.0001
History			1	−0.09034 0.0043	−0.05991 0.0583	0.03906 0.2172	0.13823 <0.0001	0.04437 0.1609	0.04217 0.1827	−0.04068 0.1987	0.0632 0.0457	−0.05378 0.0892
Purpose				1	0.06847 0.0304	−0.01868 0.5551	0.01601 0.613	0.04837 0.1264	0.00016 0.9961	−0.01761 0.5781	−0.03822 0.2272	0.01097 0.7291
Amount					1	0.06463 0.041	−0.00837 0.7916	−0.27132 <0.0001	−0.01609 0.6113	−0.02783 0.3793	0.02893 0.3608	0.3116 <0.0001
Savings						1	0.12095 0.0001	0.02199 0.4873	0.01735 0.5837	−0.10507 0.0009	0.09142 0.0038	0.01895 0.5495
Employed							1	0.12616 <0.0001	0.11128 0.0004	−0.00812 0.7977	0.24508 <0.0001	0.08719 0.0058
Installp								1	0.11931 0.0002	−0.0114 0.7189	0.0493 0.1192	0.05339 0.0915
Marital									1	0.05063 0.1096	−0.02727 0.389	−0.00694 0.8265
Coapp										1	−0.02568 0.4173	−0.15545 <0.0001
Resident											1	0.14723 <0.0001
Property												1
Age												
Other												
Housing												
Existcr												
Job												
Depends												
Telephone												
Foreign												

H0: Rho=0 Prob > |r|

	Age	Other	Housing	Existcr	Job	Depends	Telephone	Foreign
Checking	0.05975 0.0589	0.04684 0.1388	0.02242 0.4788	0.07601 0.0162	0.04066 0.1989	−0.01415 0.655	0.0663 0.0361	−0.02676 0.398
Duration	−0.03614 0.2536	−0.05488 0.0828	0.15705 <0.0001	−0.01128 0.7216	0.21091 <0.0001	−0.02383 0.4515	0.16472 <0.0001	−0.1382 <0.0001
History	0.14709 <0.0001	0.12197 0.0001	0.06209 0.0496	0.43707 <0.0001	0.01035 0.7437	0.01155 0.7153	0.05237 0.0979	0.01387 0.6613
Purpose	0.00131 0.967	−0.09661 0.0022	0.01839 0.5613	0.05494 0.0825	0.00808 0.7985	−0.03258 0.3034	0.07837 0.0132	−0.09972 0.0016
Amount	0.03272 0.3013	−0.04601 0.146	0.13563 <0.0001	0.02079 0.5113	0.28539 <0.0001	0.01714 0.5882	0.277 <0.0001	−0.05005 0.1137
Savings	0.08425 0.0077	0.00191 0.9519	0.00651 0.8372	−0.02164 0.4942	0.01171 0.7115	0.02751 0.3848	0.08721 0.0058	0.00709 0.8227
Employed	0.25623 <0.0001	−0.04015 0.2046	0.11113 0.0004	0.12579 <0.0001	0.10122 0.0013	0.09719 0.0021	0.06052 0.0557	−0.02723 0.3897
Installp	0.05827 0.0655	−0.00098 0.9752	0.0894 0.0047	0.02167 0.4937	0.09776 0.002	−0.07121 0.0243	0.01441 0.6489	−0.09002 0.0044
Marital	0.00778 0.8058	−0.03676 0.2454	0.09958 0.0016	0.06467 0.0409	−0.01196 0.7057	0.12216 0.0001	0.02727 0.3889	0.06562 0.038
Coapp	−0.02987 0.3453	−0.05902 0.0621	−0.06589 0.0372	−0.02545 0.4215	−0.05796 0.0669	0.0204 0.5193	−0.07503 0.0176	0.118 0.0002
Resident	0.26642 <0.0001	0.00209 0.9474	0.01194 0.7061	0.08963 0.0046	0.01265 0.6894	0.04264 0.1778	0.09536 0.0025	−0.0541 0.0873
Property	0.07261 0.0217	−0.09003 0.0044	0.34522 <0.0001	−0.00777 0.8063	0.27615 <0.0001	0.01187 0.7077	0.1968 <0.0001	−0.13246 <0.0001
Age	1	−0.04235 0.1809	0.30142 <0.0001	0.14925 <0.0001	0.01567 0.6206	0.1182 0.0002	0.14526 <0.0001	−0.00615 0.846
Other		1	−0.0723 0.0222	−0.04844 0.1258	−0.00478 0.8801	−0.07689 0.015	−0.01936 0.5409	0.01521 0.6309

Table 3 (*continued*)

H0: Rho=0 Prob > |r|

	Age	Other	Housing	Existcr	Job	Depends	Telephone	Foreign
Housing			1	0.04859 0.1246	0.10719 0.0007	0.11451 0.0003	0.10241 0.0012	−0.06358 0.0444
Existcr				1	−0.02632 0.4057	0.10967 0.0005	0.06555 0.0382	−0.00972 0.7589
Job					1	−0.09356 0.0031	0.38302 <0.0001	−0.10094 0.0014
Depends						1	−0.01475 0.6412	0.07707 0.0148
Telephone							1	−0.1074 0.0007
Foreign								1

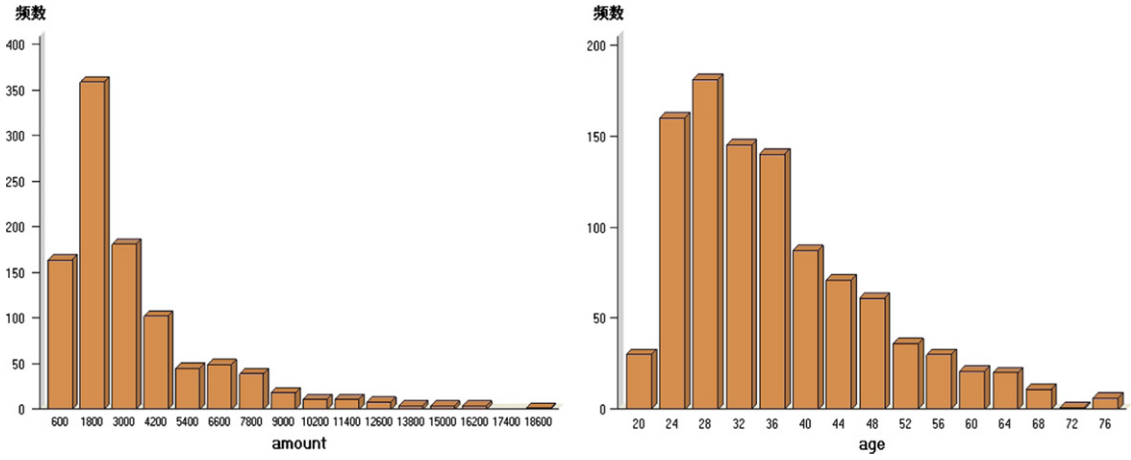

Fig. 4. Distributions of amount and age.

Table 4
Logistic regression reducing categorical variables.

Steps	Improvement			Model			OA (%)	Variables
	x^2	*df*	*Sig.*	x^2	*df*	*Sig.*		
1	131.336	3	0	131.336	3	0	70.00	checking_4
2	38.497	1	0	169.833	4	0	73.40	coapp_1
3	29.311	4	0	199.144	8	0	74.80	history_2
4	33.509	9	0	232.653	17	0	76.20	purpose_3
5	18.753	4	0.001	251.406	21	0	76.20	savings_1
6	11.133	2	0.004	262.539	23	0	76.60%	employed_3
7	6.488	1	0.011	269.027	24	0	77.40	housing_2
8	7.015	1	0.008	286.944	28	0	77.50%	job_3
9	8.561	2	0.014	295.505	30	0	78.00%	other_3

methodology is employed to test the effect of sampling variation on the model performance. To test the robustness of the models, one should apply a simple validation technique, by dividing the data set into a training sample and a validating sample, with a small scale which evaluates the predictive effectiveness of the fitted model. Second, to study the overall predictive capability of the classification models for unknown populations, one should use the whole data set as a large test set, if the data set for unknown population is not available.

To implement the first cross-validation methodology, we divide the data sample into four mutually exclusive equal sub-samples. Each sub-sample has the same rate for the bad customers and the good ones. We train the logistic regression and the SVM with three sub-samples, and validate the models with the fourth remaining sub-sample. Therefore, out-of-train prediction of validation gives us a relative true classification rate of all the observations in the data set with its averages. Secondly, to test the overall predictive capability of the unknown population comprehensively, we use the entire data sample, by using the entire data set as the test sample; we can reduce the sampling variation in the test design. Finally, we apply statistical tests to test these

Table 5
Eigenvalues and eigenvectors of the correlation matrix.

	Eigenvalue	Difference	Proportion	Cumulative
1	1.66	0.25	0.24	0.24
2	1.41	0.29	0.20	0.44
3	1.12	0.18	0.16	0.60
4	0.94	0.07	0.13	0.73
5	0.87	0.15	0.12	0.86
6	0.72	0.44	0.10	0.96
7	0.28		0.04	1.00

Table 6
Eigenvectors of the correlation matrix.

	Eigenvectors						
	z1	z2	z3	z4	z5	z6	z7
Duration	0.66	−0.04	0.32	0.20	0.14	0.03	−0.64
Amount	0.72	−0.01	−0.06	−0.03	−0.05	−0.08	0.69
Installp	−0.22	0.12	0.75	0.43	0.29	−0.01	0.33
Residence	0.06	0.54	0.22	−0.49	0.00	0.64	0.02
Age	0.01	0.62	0.06	−0.24	0.04	−0.73	−0.09
Existcr	0.02	0.44	−0.14	0.59	−0.65	0.13	−0.02
Depends	0.03	0.33	−0.52	0.35	0.69	0.17	0.01

models for accuracy. We use a paired-t test to test the difference between the means of the original method and the method hybridizing reduction of dimension in experimental and illustrate the application in Section 6.

5.6. The evaluation criteria

The problem of credit scoring mainly focuses on the accuracy of classification. As the criterion of accuracy, the goal is to judge the good ones from the bad. Let the number of creditworthy cases classified as good be GG and classified as bad be GB; denote the number of default cases classified as good with BG and as bad with BB. Then, the evaluation criteria measure the accuracy of the classification, which is defined as follows:

$$\text{Good credit accuracy (GCA)} = \frac{GG}{GG+GB}\times 100\%$$

$$\text{Bad credit accuracy (BCA)} = \frac{BB}{BG+BB}\times 100\%$$

$$\text{Overall accuracy (OA)} = \frac{GG+BB}{GG+GB+BG+BB}\times 100\% \quad (15)$$

Defined by these three indicators, one can see GCA is the specificity, which determines the ability to identify good clients; BCA is the sensitivity for the model that shows the ability to identify bad customers. At the same time, OA gives the total efficiency of the model and reflects prediction accuracy of the model and can compare with others.

In our study, Type I error occurs when a bad credit is classified as good credit, which equals 1-BCA. And Type II error occurs when a good credit is classified as a bad credit, which equals 1-GCA. For credit scoring, Type I error is more critical than Type II error. Note that all these measures with Type I error and Type II error are mostly obtained using a 0.5 probability threshold for the classification. However, the use of arbitrary cut-off probabilities makes the computed error rates difficult to interpret and the use of a relevant pay-off function and prior probabilities to determine the optional model could lead to some types of bias on the results. Bradley (1997) gave a way to judge the efficiency with ROC. The receiver operating characteristics (ROC) graph is useful for organizing classifiers and visualizing their performance. The ROC graph is a two-dimensional graph in which the BCA (true positives) rate is plotted on the Y axis and the 1-GCA (false positives) rate is plotted on the X axis. The ROC graph depicts relative trade-offs between benefits (true positives) and costs (false positives).

Most classifiers naturally yield an instance probability or score, a numeric value that represents the degree to which an instance is a member of a class. Such a ranking or scoring classifier can be used with a threshold to produce a discrete (binary) classifier. Each threshold value produces a different point in ROC space, and if we join all these points we obtain an ROC curve. Additionally, as the production process of ROC, it can be seen that the diagonal line $y=x$ represents the strategy of randomly guessing a class. To compare classifiers we want to reduce ROC performance into a single scalar value representing expected performance. A common method is to calculate the area under the ROC curve, abbreviated as AUC. Because the AUC is a portion of the area of the unit square, its value will always be between 0 and 1. However, because random guessing produces the diagonal line between (0, 0) and (1, 1), which has an area of 0.5, no realistic classifier should have an AUC less than 0.5. In Fawcett's (2006) study, he compared popular machine learning algorithms using AUC and found that AUC is a highly effective way to measure the results of models, and it can exhibit several desirable properties compared to accuracy. For example, AUC has increased sensitivity in analysis of variance tests, is independent to the decision threshold, and is invariant to a priori class probability distributions. Moreover, the AUC measure is more sensitive to the errors on the positive class, because it has important statistical meaning: it is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance, considering also all possible thresholds. It is very meaningful because in the credit scoring problem, cost will increase if one judges a bad credit as a worthy one.

Because of these reasons, the AUC has been selected as the main evaluation criteria in this paper, which can be seen in the ROC pictures, and the BCA and GCA are also be used for checking the accuracy of these models as a reference.

6. Experiments results

The experimental results presented in this paper are structured in three subsections. The first subsection describes the performance difference of original methods between the one with attributes normalization and the one without. The next section shows performance of original methods and methods hybridizing with reducing dimension. Finally, the comparison of accuracy and ROC is shown in details in the third subsection. The process of experiments is shown in Fig. 6 as below, data set is described by round said, variables are with elliptic sections, and the methods and models are with rectangular.

6.1. Performance difference with attributes normalization

In this section, we focus on the original methods to discuss the effect of attributes normalization. We use the test data set (1000

Table 8
Variance of Z.

Var (z)	$z1$ Duration	$z3$ Installp	$z4$ Residence	$z5$ Age	$z6$ Existcr	$z7$ Depends
	0.587915	0.401513	0.7569	0.000126	0.971213	0.8464

Table 7
Correlation matrix of numerical attributes.

	Duration	**Amount**	**Installp**	**Residence**	**Age**	**Existcr**	**Depends**
Duration	1.00	0.39	0.01	0.00	0.00	0.00	0.00
Amount	0.39	1.00	0.07	0.00	0.00	0.00	0.00
Installp	0.01	0.07	1.00	0.00	0.00	0.00	0.01
Residence	0.00	0.00	0.00	1.00	0.07	0.01	0.00
Age	0.00	0.00	0.00	0.07	1.00	0.02	0.01
Existcr	0.00	0.00	0.00	0.01	0.02	1.00	0.01
Depends	0.00	0.00	0.01	0.00	0.01	0.01	1.00
Total	1.40	1.47	1.09	1.09	1.11	1.04	1.03

observations and 20 variables) for this experiment. The method for normalization is introduced in Section 5.

Table 10 lists the results of these analysis models with and without normalization. From the results, we can see that the models with normalization have the same results as the models without normalization in LG. However, there is a very interesting thing with SVM. Without normalization the results using SVM are very bad and the main error occurs in the prediction of the good credit. We also perform some similar experiments with numerical variables reduced by PCA and ODR to test the results and these results are the same: SVM cannot do the right job without normalization in the good prediction. We propose that this phenomenon may be caused by dimension curse, because different dimension units can get different variance scales, so in the high space, the distribution of data will be sparse, and the plane will not be sensitive to some specific points. Until now, however, we have found no relevant information introduced, so we define it as 'Dimensional Interference'. Thus, in the following experiments, we use only the models with normalization.

6.2. Performance difference with reducing dimensions

Based on the methods for reducing dimensions introduced in 4.D and just as the introduction of cross-validation design above in Section 5.5, we have four times experiments with the train data set of 750 observations tagged as D1, and then apply four times validation with the validation data set of 250 observations tagged as D2 left by the train data set. Each of these data sets uses equal proportions of the bad ones. The entire data set is used as the test data set tagged as D3. The results are the average of these data sets for four experiments. The cross-validation results of these models with normalization are summarized in Table 10.

From Table 11, there are several interesting findings. First of all is that with these models, without reducing dimensions, we can see all of these models do not get satisfactory accuracy in the prediction for the bad ones, though OA is really high with the train data set. Second, the SVM model is a little better and with the expense of the good customer forecast. There is an over-fitting problem with logistic regression and SVM. Because logistic regression is the method based on the variance of variables, it lacks robustness, from Table 10, we can find it is a very good model for the train data set, but it is not the same good to the validation data set and the test data set. Also there is the same problem with SVM. Furthermore, with reducing dimensions, it can be found that there are improvements with SVM in all the data sets.

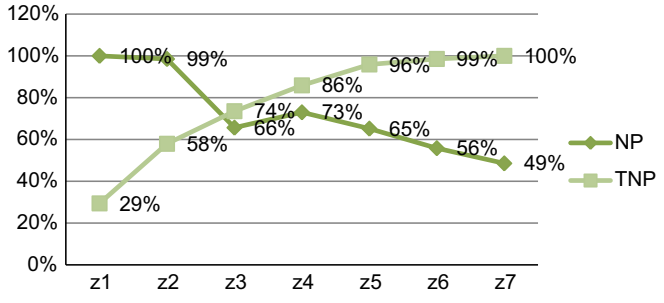

Fig. 5. Accumulative contribution of Z.

6.3. Comparison of models

To conduct a comparison of these models, we design four groups for comparative test.

First, compare the performance between LG and SVM. We conduct experiments separately, and the results are shown in the following Tables 12–14.

From these tables, we can find the performance of SVM is better than logistic regression, especially for these reducing dimension models. This point is consistent with recent research.

Second, analyze the effect of dimension reduction for the logistic regression model. We use the logistic regression model and the other two reducing dimension logistic regression models to apply comparative analysis. The results of the comparison can be found in Tables 15 and 16.

As shown in Tables 15 and 16, on average, the overall performance of reducing dimension models for prediction is not better than the original models at a 5% rejection level. This may be due to the limited variables in the model, so that the linear relationship is not obvious. In these models, it is still difficult to say that the model with reducing dimension is significantly improved.

Furthermore, we compare these results between each model to analyze the effect of dimension reduction SVM. A comparison of results is listed in Table 17.

From Table 17, we can find that some models have the same accuracy, such as HLG–SVM and ODR–SVM; these are in italics. From the accuracy of BCA, we can determine that the HLG–ODR–SVM is the best one. HLG–SVM and PCA–SVM are the same at the accuracy of BCA, and what is more is that HLG–SVM has more accuracy than ODR–SVM and it is statistically significant. For GCA, there is not much obvious improvement with these models. We still can find that PCA–SVM is not a good model for the prediction accuracy because other models, except HLG–PCA–SVM, can outperform it at the 5% significance level. For OA, there is an increase for SVM and PCA–SVM, so we can infer that dimension reduction caused the overall increase in accuracy, which mainly comes from the increase in BCA.

This boost in BCA is mainly dues to reduction of redundant variables and getting the major characteristic which increased the model's predictive power; while due to the reduction of variables, the amount of information about customers is also reduced, so it leads to some difficulty in upgrading the good ones.

Finally, find the best model for credit scoring. Through the first three discussions on the comparison, we know that SVM can outperform LG and that reducing dimension does not improve LG much but improves SVM a lot. Therefore, to further compare the accuracy of the model with any distribution, we provide the ROC graphs with these reducing dimension SVMs (PCA–SVM, ODR–SVM, HLG–SVM, HLG–PCA–SVM, HLG–ODR–SVM) in Fig. 7, which describes the ROC of the validation dataset and in Fig. 8 ,which is

Table 9
Contribution of *Z*.

	*z*1 (amount)	*z*2 (existcr)	*z*3 (duration)	*z*4 (installp)	*z*5 (residence)	*z*6 (depends)	*z*7 (age)
VAR (*z*)	1	0.97	0.53	0.42	0.34	0.09	0.05
NP (*z*)	100%	99%	66%	73%	65%	56%	49%
TNP (*z*)	29%	58%	74%	86%	96%	99%	100%

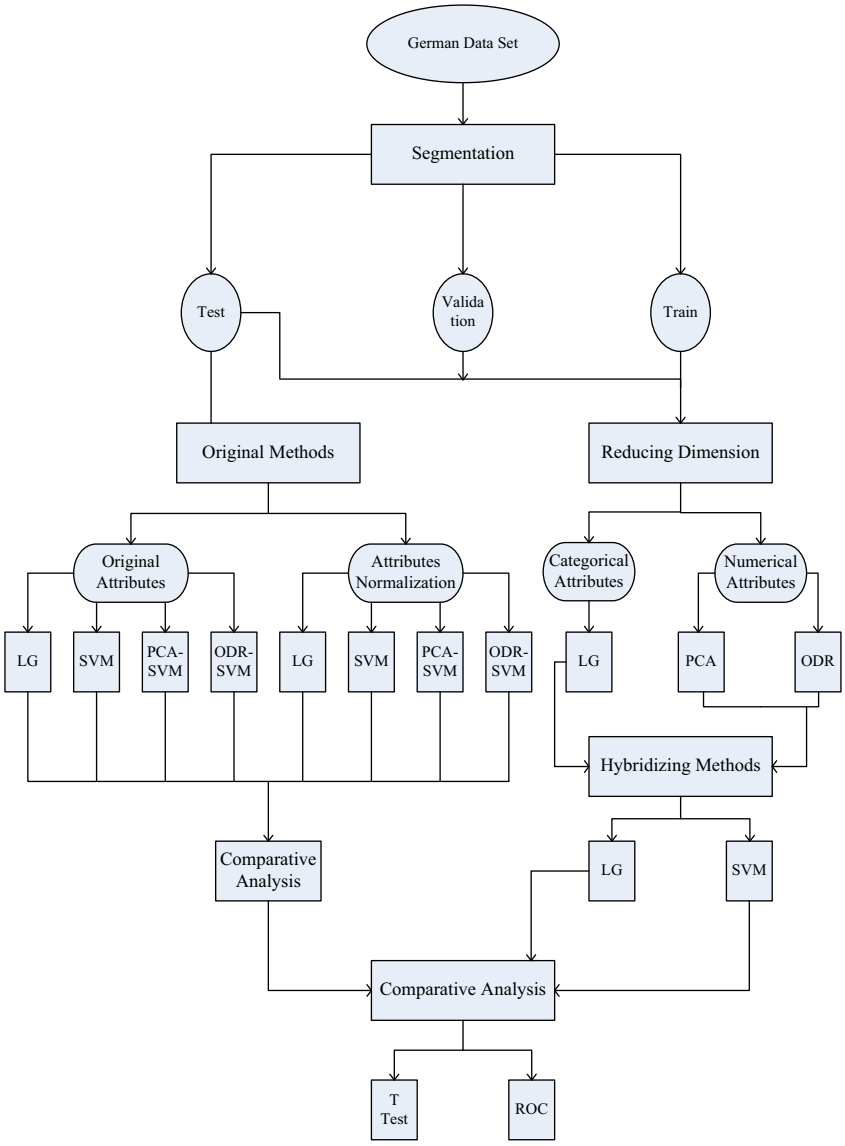

Fig. 6. The process of experiments.

Table 10
Classification of different models with/without normalization.

		With normalization			
Goal	Result	LG	SVM	PCA–SVM	ODR–SVM
Bad	Bad	160	184	151	161
Bad	Good	140	116	149	139
Good	Good	624	637	614	623
Good	Bad	76	63	86	77
	OA	78.4%	82.1%	76.5%	78.4%
		Without normalization			
Goal	Result	LG	SVM	PCA–SVM	ODR–SVM
Bad	Bad	160	116	149	176
Bad	Good	140	184	151	124
Good	Good	624	181	233	217
Good	Bad	76	519	467	483
	OA	78.4%	**29.7**%	**38.2**%	**39.3**%

Table 11
Cross-validation results of these models.

Comparative Analysis																										
		Without reducing dimensions						With reducing dimensions																		
								Feature extraction										Feature selection								
		LG			SVM			PCA–LG			ORD–LG			PCA–SVM			ODR–SVM			HLG–SVM			HLG–PCA–SVM			HLG–ODR–SVM		
		D1	D2	D3	D1	D2	D3	D1	D2	D3	D1	D2	D3	D1	D2	D3	D1	D2	D3	D1	D2	D3	D1	D2	D3	D1	D2	D3
Bad	Bad	110	31	127	113	32	129	114	32	139	115	34	138	139	45	174	141	47	178	140	44	177	142	48	183	147	52	194
Bad	Good	115	44	173	112	43	171	111	43	161	110	41	162	86	30	126	84	28	122	85	31	123	83	27	117	78	23	106
Good	Good	426	102	402	411	134	503	428	112	416	411	107	415	399	130	502	433	138	496	432	139	511	422	132	505	425	136	508
Good	Bad	99	73	298	114	41	197	97	63	284	114	68	285	126	45	198	92	37	204	93	36	189	103	43	195	100	39	192
	BCA	49%	41%	42%	50%	43%	43%	51%	43%	46%	51%	45%	46%	62%	60%	58%	63%	63%	59%	62%	59%	59%	63%	64%	61%	65%	69%	65%
	GCA	81%	58%	57%	78%	77%	72%	82%	64%	59%	78%	61%	59%	76%	74%	72%	82%	79%	71%	82%	79%	73%	80%	75%	72%	81%	78%	73%
	OA	71%	53%	53%	70%	66%	63%	72%	58%	56%	70%	56%	55%	72%	70%	68%	77%	74%	67%	76%	73%	69%	75%	72%	69%	76%	75%	70%

the ROC of the test dataset. Therefore, AUG can be seen clearly and we can choose the best model more easily.

With the ROC graph, we can see that no model can be completely superior to other models in any case. The AUGs between PCA–SVM, ODR–SVM and HLG–PCA–SVM are almost the same, a little weaker for HLG–SVM and a little stronger in HLG–ODR–SVM. Upon the above comparison, therefore, HLG–ODR–SVM is most effective method in this credit scoring problem.

We can summarize the advantages of orthogonal dimension reduction models. First of all, reducing dimension has better accuracy in BCA, which is the key index for credit scoring. Then, ODR reduces the less correlated variables so enables rapidity of convergence of the SVM models. Lastly, using the features extracted by orthogonal dimension reduction hybridizing logistic regression gives the modeler better explanations.

7. Conclusions

Most recently, researchers have found support vector machine can provide better performance in the prediction of credit scoring. However, support vector machine is a black-box method and lacks rules for selecting good input variables. Similar to other artificial intelligence methods, they face the problem of 'garbage in, garbage out'. Thus SVM is usually troubled with dimension curse. Focusing on this, we introduce orthogonal feature extraction techniques with logistic regression and support vector machine, which has better interpretability for the input variables, reduces the dimension and accelerates convergence.

Our research follows the next processes. Because SVM is sensitive to initial condition and training algorithms, we use a grid search to select parameters for the SVM. Then, we design the process of reducing dimension with PCA, ODR and HLG to reduce the redundant variables, thus this resolves the problem of high multicollinearity to a certain degree. Finally, we experiment on these methods using the filtered features with the same training methods to test the effectiveness.

We have also found out in our experiments that SVM cannot work well with the numerical variables which are not normalized. We call this phenomenon as 'Dimensional interference', as described in Section 6.1 in detail. That may be caused by the distribution of variables, which stops the kernel function effectively transform it to the high dimension. So in the high dimension space, it still cannot be separated by line. However, there are few references that have discussed this. It still needs more theoretical study.

To assess the performance of these experiments, we experiment with two types of cross-validation—a small data set and a complete data set, and use a paired-t test to test the means for different accuracy between original models and the ones with reducing dimension. This way can validate the robustness of the models and give statistical significance for the improvement in accuracy. Because Type I error is more serious in credit scoring and cannot be arbitrarily cut off, we also use the area under the ROC graph to judge the effectiveness between different models. We find that no model can be completely superior to other models in any case, but SVMs with orthogonal feature extraction techniques have improved the prediction of BCA and OA. Obviously, PCA is not a good choice for SVMs reducing dimension and in total the AUG shows there is a little success with the orthogonal feature extraction SVM hybridizing logistic regression.

Based on those assessments there is also an interesting finding that the lift in SVMs mostly comes from the lift in

Table 12
Pairwise comparsion between LG and SVM.

mean		Bad loan		Good loan		Overall	
		LG	SVM	LG	SVM	LG	SVM
		0.44	0.45	0.65	0.76	0.59	0.66
t-Value	Pooled	−1.03		−3.39		−3.06	
	Satterthwaite	−1.03		−3.39		−3.06	
	Cochran	−1.03		−3.39		−3.06	
pr > |t|	Pooled	0.31		<0.05		<0.05	
	Satterthwaite	0.31		<0.05		<0.05	
	Cochran	0.31		<0.05		<0.05	

Table 13
Pairwise comparsion between PCA–LG and PCA–SVM.

mean		Bad loan		Good loan		Overall	
		PCA–LG	PCA–SVM	PCA–LG	PCA–SVM	PCA–LG	PCA–SVM
		0.47	0.6	0.68	0.74	0.62	0.7
t-Value	Pooled	−13.55		−2.12		−4.1	
	Satterthwaite	−13.55		−2.12		−4.1	
	Cochran	−13.55		−2.12		−4.1	
pr > |t|	Pooled	<0.05		<0.05		<0.05	
	Satterthwaite	<0.05		0.05		<0.05	
	Cochran	<0.05		0.05		<0.05	

Table 14
Pairwise comparsion between ORD–LG and ODR–SVM.

mean		Bad loan		Good loan		Overall	
		ODR–LG	ODR–SVM	ODR–LG	ODR–SVM	ODR–LG	ODR–SVM
		0.47	0.62	0.66	0.77	0.6	0.73
t-Value	Pooled	−14.91		2.02		−5.75	
	Satterthwaite	−14.91		2.02		−5.75	
	Cochran	−14.91		2.02		−5.75	
pr > |t|	Pooled	<0.05		0.05		<0.05	
	Satterthwaite	<0.05		0.05		<0.05	
	Cochran	<0.05		0.05		<0.05	

Table 15
Pairwise comparsion between LG and PCA–LG.

mean		Bad loan		Good loan		Overall	
		LG	PCA–LG	LG	PCA–LG	LG	PCA–LG
		0.44	0.47	0.65	0.68	0.59	0.62
t-Value	Pooled	−0.78		−0.29		−0.38	
	Satterthwaite	−0.78		−0.29		−0.38	
	Cochran	−0.78		−0.29		−0.38	
pr > |t|	Pooled	0.48		0.78		0.72	
	Satterthwaite	0.48		0.79		0.72	
	Cochran	0.52		0.81		0.73	

Table 16
Pairwise comparsion between LG and ODR–LG.

mean		Bad loan		Good loan		Overall	
		LG	ODR–LG	LG	ODR–LG	LG	ODR–LG
		0.44	0.47	0.65	0.66	0.59	0.6
t-Value	Pooled	−1.07		−0.07		−0.17	
	Satterthwaite	−1.07		−0.07		−0.17	
	Cochran	−1.07		−0.07		−0.17	
pr > |t|	Pooled	0.35		0.95		0.87	
	Satterthwaite	0.35		0.95		0.87	
	Cochran	0.4		0.95		0.87	

Table 17
Pairwise comparsion of svm models.

BCA Mean		SVM (mean=0.45)					PCA–SVM(mean=0.60)				ODR–SVM (mean=0.62)			HLG–SVM (mean=0.60)		HLG–PCA–SVM (mean=0.63)
		PCA–SVM	ODR–SVM	HLG–SVM	HLG–PCA–SVM	HLG–ODR–SVM	ODR–SVM	HLG–SVM	HLG–PCA–SVM	HLG–ODR–SVM	HLG–SVM	HLG–PCA–SVM	HLG–ODR–SVM	HLG–PCA–SVM	HLG–ODR–SVM	HLG–ODR–SVM
		0.60	0.62	0.60	0.63	0.66	0.62	0.60	0.63	0.66	0.60	0.63	0.66	0.63	0.66	0.66
t-Value	Pooled	−13.21	−14.25	−13.55	−16.30	−18.33	−2.22	0.00	−4.30	−8.42	2.35	−1.47	−5.80	−4.69	−8.91	−5.38
	Satterthwaite	−13.21	−14.25	−13.55	−16.30	−18.33	−2.22	0.00	−4.30	−8.42	2.35	−1.47	−5.80	−4.69	−8.91	−5.38
	Cochran	−13.21	−14.25	−13.55	−16.30	−18.33	−2.22	0.00	−4.30	−8.42	2.35	−1.47	−5.80	−4.69	−8.91	-5.38
pr > |t|	Pooled	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**	**0.0371**	*1.0000*	**0.0003**	**<0.0001**	**0.0284**	0.1565	**<0.0001**	**0.0001**	**<0.0001**	**<0.0001**
	Satterthwaite	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**	**0.0377**	*1.0000*	**0.0003**	**<0.0001**	**0.0292**	0.1587	**<0.0001**	**0.0001**	**<0.0001**	**<0.0001**
	Cochran	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**	**0.0489**	*1.0000*	**0.0012**	**<0.0001**	**0.0388**	0.1704	**<0.0001**	**0.0001**	**<0.0001**	**0.0002**

GCA Mean		SVM (mean=0.76)					PCA–SVM (mean=0.74)				ODR–SVM (mean=0.77)			HLG–SVM (mean=0.78)		HLG–PCA–SVM (mean=0.76)
		PCA–SVM	ODR–SVM	HLG–SVM	HLG–PCA–SVM	HLG–ODR–SVM	ODR–SVM	HLG–SVM	HLG–PCA–SVM	HLG–ODR–SVM	HLG–SVM	HLG–PCA–SVM	HLG–ODR–SVM	HLG–PCA–SVM	HLG–ODR–SVM	HLG–ODR–SVM
		0.74	0.77	0.78	0.76	0.77	0.77	0.78	0.76	0.77	0.78	0.76	0.77	0.77	0.77	0.77
t-Value	Pooled	1.79	−1.04	−1.69	0.00	−1.31	−2.25	−3.25	−1.50	−3.00	−0.37	0.97	0.00	1.55	0.44	−1.18
	Satterthwaite	1.79	−1.04	−1.69	0.00	−1.31	−2.25	−3.25	−1.50	−3.00	−0.37	0.97	0.00	1.55	0.44	−1.18
	Cochran	1.79	−1.04	−1.69	0.00	−1.31	−2.25	−3.25	−1.50	−3.00	−0.37	0.97	0.00	1.55	0.44	−1.18
pr > |t|	Pooled	0.0882	0.3121	0.1002	*1.0000*	0.2031	**0.0351**	**0.0037**	0.1475	**0.0066**	0.7143	0.3424	*1.0000*	0.1351	0.6619	0.2489
	Satterthwaite	0.0886	0.3133	0.1006	*1.0000*	0.2043	**0.0422**	**0.0054**	0.1526	**0.0084**	0.7145	0.3435	*1.0000*	0.1353	0.6620	0.2489
	Cochran	0.1002	0.3214	0.1221	*1.0000*	0.2186	**0.0462**	**0.0077**	0.1614	**0.0120**	0.7178	0.3527	*1.0000*	0.1491	0.6662	0.2612

OA Mean		SVM (mean=0.66)					PCA–SVM (mean=0.70)				ODR–SVM (mean=0.73)			HLG–SVM (mean=0.73)		HLG–PCA–SVM (mean=0.72)
		PCA–SVM	ODR–SVM	HLG–SVM	HLG–PCA–SVM	HLG–ODR–SVM	ODR–SVM	HLG–SVM	HLG–PCA–SVM	HLG–ODR–SVM	HLG–SVM	HLG–PCA–SVM	HLG–ODR–SVM	HLG–PCA–SVM	HLG–ODR–SVM	HLG–ODR–SVM
		0.70	0.73	0.73	0.72	0.74	0.73	0.73	0.72	0.74	0.73	0.72	0.74	0.72	0.74	0.74
t-Value	Pooled	−3.69	−4.14	−5.18	−4.98	−6.26	−1.97	−2.68	−2.25	−3.93	0.00	0.46	−0.67	0.59	−0.85	−1.54
	Satterthwaite	−3.69	−4.14	−5.18	−4.98	−6.26	−1.97	−2.68	−2.25	−3.93	0.00	0.46	−0.67	0.59	−0.85	−1.54
	Cochran	−3.69	−4.14	−5.18	−4.98	−6.26	−1.97	−2.68	−2.25	−3.93	0.00	0.46	−0.67	0.59	−0.85	−1.54
pr > |t|	Pooled	**0.0010**	**0.0004**	**<0.0001**	**<0.0001**	**<0.0001**	0.0623	**0.0137**	**0.0346**	**0.0007**	*1.0000*	0.6532	0.5093	0.5636	0.4027	0.1379
	Satterthwaite	**0.0020**	**0.0005**	**<0.0001**	**<0.0001**	**<0.0001**	0.0694	**0.0156**	**0.0361**	**0.0009**	*1.0000*	0.6542	0.5106	0.5636	0.4028	0.1380
	Cochran	**0.0040**	**0.0017**	**0.0003**	**0.0004**	**<0.0001**	0.0752	**0.0214**	**0.0456**	**0.0023**	*1.0000*	0.6576	0.5162	0.5695	0.4117	0.1519

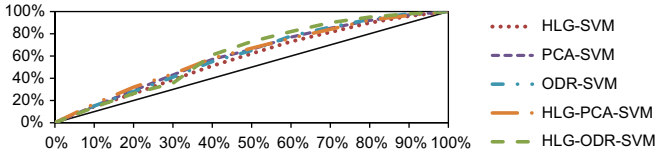

Fig. 7. The ROC graph for validation dataset.

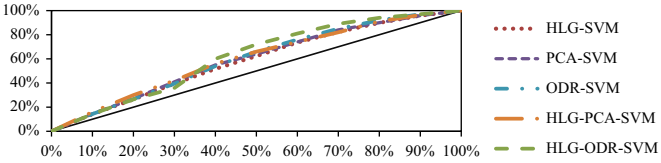

Fig. 8. The ROC graph for test dataset.

BCA. For a more elaborate model, such as HLG–PCA–SVM and HLG–ODR–SVM, there always is a decline in GCA, though it is not statistically significant, which is understandable because of the reduction of the variable: the model has a power to better distinguish core features, which may give more results for the bad ones, but it will lose some information and lower the specificity for a degree, which decreases the accuracy for some good cases. Fortunately, GCA, of all the models, is quite impressive.

Finally, to summarize, this study provides a new way – orthogonal dimension reduction – to address dimension curse, and it has an impressive effect in SVM for credit scoring, which is quite distinct from these methods based on techniques improvement. We discuss the related properties of this method in detail and test other common statistical approaches – principal component analysis and hybridizing logistic regression – to better solve and evaluate the problem. Moreover, in our opinion, for other applications such as pattern recognition, this method can also be used. The results potentially need further discussion, but we propose that it will prove highly helpful because reducing dimension methods can lead better interpretations for the variables selection at a minimum.

Acknowledgment

R.B.G. thanks Project supported by the National Natural Science Foundation of China (Grant nos. 70831001, 70821061 and 71232003).

References

Altman, E.I., 1998. Credit risk measurement: developments over the last 20 years. J. Bank. Finan. 21, 1721–1742.
Anderson, T.W., 1962. An Introduction to Multivariate Statistical Analysis. Wiley, New York.
Bellman, R., 1961. Adaptive Control Processes: A Guided Tour. Princeton University Press, Princeton.
Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Patt. Recog. 30, 1145–1159.
Cortes, C., Vapnik, V., 1995. Support vector networks. Mach. Learn. 20, 273–297.
Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugen. 7, 179–188.
Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition. Academic Press, Boston.
Fawcett, T., 2006. An introduce to ROC analysis. Patt. Recog. 27, 861–874.
Gestel, T.V., Baesens, B., Garcia, J., et al., 2003. A support vector machine approach to credit scoring. Bank. Finan. 2, 73–82.
Gutierrez, P.A., Vargas, M.J.S., Sanz, S.S., et al., 2010. Hybridizing logistic regression with product unit and RBF networks for accurate detection and prediction of banking crises. Omega 38, 333–344.
Hand, D.J., Henley, W.E., 1997. Statistical classification methods in consumer credit scoring. J. R. Stat. Soc. Ser. A 160, 523–541.
Huang, S.C., 2009. Integrating nonlinear graph based dimensionality reduction schemes with SVMs for credit rating forecasting. Expert Syst. Appl. 36 (4), 7515–7518.
Han, L., Han, L.Y., Zhao, H.W., Combined model of empirical study for credit risk management, In: Proceedings of 2010 2nd IEEE International Conference on Information and Financial Engineering, ICIFE 2010: pp. 485–489.
Hua, Z., Wang, Y., Xu, X., et al., 2007. Predicting corporate financial distress based on integration of support vector machine and logistic regression. Expert Syst. Appl. 33 (2), 434–440.
Jolliffe, I.T., 2002. Principal Component Analysis. Springer-Verlag, Berlin.
Jack, L.B., Nandi, A.K., 2002. Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms. Mech. Syst. Signal Process. 16 (2–3), 373–390.
Jain, S.K., Gunawardena, A.D., 2003. Linear Algebra: An Interactive Approach. China Machine Press, Beijing.
Liu, L., Zechman, E.M., Mahinthakumar, G., et al., 2012. Coupling of logistic regression analysis and local search methods for characterization of water distribution system contaminant source. Eng. Appl. Artif. Intell. 25, 309–316.
Martin, D., 1977. Early warning of bank failure: a logistic regression approach. J. Bank. Finan. 1 (3), 249–276.
Min, J.H., Lee, Y.C., 2005. Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. Expert Syst. Appl. 28, 603–614.
Rojas, A., Nandi, A., 2006. Practical scheme for fast detection and classification of rolling-element bearing faults using support vector machines. Mech. Syst. Signal Process. 20 (7), 1523–1536.
Suykens, J.A.K., Gestel, T.V., Brabanter, J.D., et al., 2002. Least Squares Support Vector Machines. World Scientific, Singapore.
Schebesch, K., Stecking, R., 2005. Support vector machines for classifying and describing credit applicants: detecting typical and critical regions. J. Oper. Res. Soc. 56 (9), 1082–1088.
Sugiyama, M., 2007. Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis. J. Mach. Learn. Res. 8, 1027–1061.
Samanta, B., Nataraj, C., 2009. Use of particle swarm optimization for machinery fault detection. Eng. Appl. Artif. Intell. 22 (2), 308–316.
Thomas, L.C., Edelman, D.B., Crook, J.N., 2002. Credit Scoring and its Applications. Society of Industrial and Applied Mathematics, Philadelphia.
Wiginton, J.C., 1980. A note on the comparison of logic and discriminant models of customer credit behavior. J. Finan. Quant. Anal. 15, 757–770.
Yang, Y.X., 2007. Adaptive credit scoring with kernel learning methods. Eur. J. Oper. Res. 183 (3), 1521–1536.
Yu, L.A., Wang, S.Y., Lai, K.K., 2008. Bio-Inspired Credit Risk Analysis. Springer-Verlag, Berlin.
Yu, L.A., Huang, W., Lai, K.K., et al., 2006. A reliability-based RBF network ensemble model for foreign exchange rates predication. Lect. Notes Comput. Sci. 4234, 380–389.
Yuan, S.F., Chu, F.L., 2007. Fault diagnosis based on support vector machines with parameter optimisation by artificial immunisation algorithm. Mech. Syst. Signal. Process. 21 (3), 1318–1330.
Zhang, G.Q., Hu, M.Y., Patuwo, B.E., et al., 1999. Artificial neural networks in bankruptcy prediction: general framework and cross-validation analysis. Eur. J. Oper. Res. 16 (1), 16–32.