2 Credit Risk Assessment Using a Nearest-Point-Algorithm-based SVM with Design of Experiment for Parameter Selection

2.1 Introduction

Credit risk assessment has become an increasingly important area for financial institutions, especially for banks and credit card companies. In the history of financial institutions, some biggest failures were related to credit risk, such as the 1974 failure of Herstatt Bank (Philippe, 2003). In recent years, many financial institutions suffered a great loss from a steady increase of defaults and bad loans from their counterparties. So, for the credit-granting institution, the ability to accurately discriminate the good counterparties and the bad ones has become crucial. In the credit industries, the quantitative credit scoring model has been developed for this task in many years, whose main idea is to classify the credit applicants to be good or bad according to their characters (age, income, job status, etc.) by the model built on the massive information on previous applicants' characters and their subsequent performance.

With the expansion of financial institutions' loan portfolios, the potential benefits from any improvement of accuracy of credit evaluation practice are huge. Even a fraction of a percent increase in credit scoring accuracy is a significant accomplishment (David, 2000). So far, a great number of classification techniques have been used to develop credit risk evaluation models. They are reviewed in the first chapter. Although so many techniques were listed above, it is just a part of all for the credit risk assessment model. Some surveys on credit risk modeling gave more details about some of these techniques, such as Hand and Henley (1997), Thomas (2002), Rosenneberg and Gleit (1994), and Thomas et al. (2005).

Among the different evaluation methods, the SVM approach first proposed by Vapnik (1995, 1998) achieved much good performance relative to other classification techniques. The main idea of SVM is to minimize the upper bound of the generalization error rather than empirical error.

Usually, SVM maps the input vectors into a high-dimensional feature space through some nonlinear mapping function. In the high-dimensional space, an optimal separating hyperplane which is one that separates the data with a maximal margin is constructed. SVM is a powerful method for classification since it has outperformed most other methods in a wide variety of applications, such as text categorization and face or fingerprint identification.

After the invention of SVM, some researchers have introduced SVM to credit risk evaluation problems (Van Gestel et al., 2003; Baesens et al., 2003; Schebesch and Stecking, 2005; Wang et al., 2005; Lai et al., 2006a, 2006c). Typically, Van Gestel et al. (2003) used least squares SVM (LS-SVM) (Suykens et al., 2002) for credit rating of banks and report the results contrasted with some classical techniques. Schebesch and Stecking (2005) used the standard SVM proposed by Vapnik with linear and RBF kernel for consumer credit scoring and at the same time, they used linear SVM to divide a set of labeled credit applicants into subsets of 'typical' and 'critical' patterns which can be used for rejected applicants. Basens et al. (2003) mainly discussed the benchmarking study of various classification techniques based on eight real-life credit datasets Similary, they also used SVM and LS-SVM with linear and RBF kernels and adopt a grid search method to tune the hyperparameters. The experiments show that RBF-LSSVM and NN classifiers yield a good performance in terms of classification accuracy. In the work of Wang et al. (2005), a new fuzzy SVM are proposed. The new fuzzy support vector machine could treat every sample as both positive and negative classes with different memberships. The total error term in the objective function in SVM is weighted by the membership of the data for its class. Recently Lai et al. (2006a, 2006c) adopt SVM and LSSVM to evaluate and analyze credit risk using meta-modeling strategy.

From a viewpoint of computation, almost the training of all kinds of SVM originally requires the solution of a quadratic programming (QP). However, when solving a large scale QP problem, the computation would be very complex. For this purpose, some algorithms have been brought forward to reduce the complexity. In practice, the performance of the SVM is related to its used algorithm. The sequential minimal optimization (SMO) algorithm (Platt, 1998) is an important one whose main ideas is to break the large QP problem into a series of smallest possible QP problem that can be solved analytically. There are other algorithms for this task such as some fast algorithms (Friess et al., 1998; Joachims, 1998). Particularly, one fast iterative algorithm for SVM proposed by Keerthi et al. (2000) is very competitive. The main idea of this algorithm is to transform a particular SVM classification problem into a problem of computing the nearest point between two convex polytopes in the hidden feature space Based on this transformation, a fast iterative nearest point algorithm (NPA) is designed and proposed. As far as we know, none of above mentioned works using SVM for solving credit scoring problem discussed this issue. The purpose of this chapter is to introduce the NPA-based support vector machine (SVM) method for credit risk evaluation to increase the classification accuracy. Particularly, the design of experiments (DOE) (Staelin, 2002) is used for parameter selection of NPA-based SVM to obtain the best parameters.

The rest of this chapter is organized as follows. Section 2.2 briefly introduces the SVM with the NPA algorithm. In Section 2.3, the parameter selection technology based on DOE is briefly discussed and the hybrid algorithm of NPA and the parameter selection is described. The results of the algorithm's testing on a real-life dataset and comparisons with other methods are discussed in Section 2.4. Section 2.5 gives a short conclusion about the chapter.

2.2 SVM with Nearest Point Algorithm

Given a set of training samples $S=\{x_k, y_k\}$, $k=1,...,m$, where x_k is the k_{th} input vectors and y_k is its corresponding observed result. Specially, in credit scoring models x_k denotes the attributes of the applicants or evaluation criteria and y_k is the observed result of timely repayment. If the customer defaults, $y_k = 1$, else $y_k = -1$. Let $I=\{ i \mid y_i = 1, i \in N, (x_i, y_i) \in S\}$, $J=\{ i \mid y_i = -1, i \in N, (x_i, y_i) \in S\}$, we assume that the samples set contains both two kind of cases ($y_k = 1, and\ y_k = -1$), It mean: $I \neq \phi, J \neq \phi, I \bigcup J = \{1,...,m\}$ and $I \bigcap J = \phi$.

Suppose $z_i = \varphi(x_i)$ where $\varphi(\cdot)$ is a nonlinear function that can maps the input space into a higher dimensional feature space (*z*-space). If the training set is linear separable in the feature space, the classifier should be constructed as follows:

$$\begin{cases} w \cdot z_i + b \geq 1 & \forall i \in I \\ w \cdot z_i + b \leq -1 & \forall i \in J \end{cases} \tag{2.1}$$

If there exist a (*w*, *b*) pair for which the constraint in formula (2.1) are satisfied, let $H_+ = \{z : w \cdot z + b = 1\}$ and $H_- = \{z : w \cdot z + b = -1\}$ be the bounding hyperplanes separating the two classes. The distance between the two boundary hyperplanes is $2/\|w\|$. One of the main steps in SVM is to construct the optimal hyperplane which can separate the samples in this *z*-space without errors and the distance between the closest vector to the hyperplane is maximal. It can be found by solving the following quadratic programming problem:

$$\begin{cases} \min J(w,b) = \|w\|^2 / 2 \\ \text{s.t.} \quad w \cdot z_i + b \geq 1, \ \forall i \in I \\ \qquad\quad w \cdot z_i + b \leq -1, \forall i \in J \end{cases} \tag{2.2}$$

In many practical situation, the training samples can not be linear separable in the *z*-space. There is a need to introduce the soft margin for which classification violations are allowed. The ideal objective is that the training can maximize the classification margin and minimize the sum of violations at the same time. These two objectives are difficult to achieve at the same time and thus a trade-off between the two objectives is needed. One popular approach for the trade off is to use the following optimization problem:

$$\begin{cases} \min J(w,b,\xi_k) = \frac{1}{2}\|w\|^2 + C\sum_k \xi_k \\ \text{s.t.} \quad w \cdot z_i + b \geq 1 - \xi_i, \quad \forall i \in I \\ \qquad\quad w \cdot z_i + b \leq 1 + \xi_i, \quad \forall i \in J \\ \qquad\quad \xi_k \geq 0, \forall\, k \in I \bigcup J \end{cases} \tag{2.3}$$

where C is a positive constant denoting a trade-off between the two goals. When C is large, the error term will be emphasized. A small C means that the large classification margin is encouraged. 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:

$$\begin{cases} \max\limits_{\alpha} J(\alpha) = \sum_k \alpha_k - \frac{1}{2}\sum_k \sum_l \alpha_k \alpha_l y_k y_l z_k \cdot z_l \\ \text{s.t.} \quad 0 \leq \alpha_k \leq C, \ \forall\, k \\ \qquad \sum_k \alpha_k y_k = 0 \\ \qquad \text{where } y_k = 1, k \in I \text{ and } y_k = -1, k \in J \end{cases} \tag{2.4}$$

In this problem, since we still do not know the mapping function, that is, z_k is still unknown. One common method for the function is to use the kernel function, $K(x_k, x_l)$, which is the inner product in the feature space, to perform the mapping. One main merit of SVM is that by a kernel function it can make the training data linearly separable in the *z*-space as possible as it can do. Usually, the commonly used kernel functions are linear, polynomial and Gaussian or radial basis function (RBF), which are listed as follows.

(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: $K(x_k, x) = \exp\left(-\left\|x_k - x\right\|^2 / \sigma^2\right)$

Another approach for taking into account the violations is to use the sum of squared violations in the objective function, which is proposed by Friess (1998).

$$\begin{cases} \min\ J(w,b,\xi_k) = \dfrac{1}{2}\|w\|^2 + \dfrac{\tilde{C}}{2}\sum_k \xi_k^2 \\ \text{s.t.} \qquad w \cdot z_i + b \geq 1 - \xi_i, \quad \forall i \in I \\ \qquad\quad w \cdot z_i + b \leq 1 + \xi_i, \quad \forall i \in J \\ \qquad\quad \xi_k \geq 0, \forall\ k \in I \bigcup J \end{cases} \tag{2.5}$$

As pointed out by Friess (1998), using a simple transformation, i.e., $\tilde{w} = \begin{pmatrix} w \\ \sqrt{\tilde{C}}\xi \end{pmatrix}$, $b = \tilde{b}$, $\tilde{z}_i = \begin{pmatrix} z_i \\ e_i / \sqrt{\tilde{C}} \end{pmatrix}$ ($i \in I$), $\tilde{z}_j = \begin{pmatrix} z_j \\ -e_i / \sqrt{\tilde{C}} \end{pmatrix}$ ($j \in J$), (2.5) can be converted into (2.2).

Using $\tilde{w}, \tilde{b}, \tilde{z}_i$ *or* $\tilde{z}_j$ to substitute w, b, z_i in (2.5), we can get an instance of (2.1). So, $\tilde{K}$, the kernel function for the transformed SVM, can be given as follows:

$$\tilde{K}(x_k, x_l) = K(x_k, x_l) + \frac{1}{\tilde{C}}\delta_{kl} \tag{2.6}$$

where

$$\delta_{kl} = \begin{cases} 1 & if\ k = l \\ 0 & otherwise \end{cases} \tag{2.7}$$

One idea to solve the (2.2) is to transform it into a problem of computing the nearest point between two convex polytopes (*U*, *V*) and then use a carefully chosen nearest point to solve it.

Let *coS* is the set of all convex combinations of elements of *S*, it can be denoted as follows:

$$coS = \{\sum_{k=1}^{l} \beta_k s_k : s_k \in S, \beta_k \geq 0, \sum_{k}^{l} \beta_k = 1\} \tag{2.8}$$

Let $U = coS\{z_i : i \in I\}$ and $V = coS\{z_i : i \in J\}$, then the SVM problem shown in (2.2) is equivalent to the following problem of computing the minimum distance between *U* and *V*, that is, nearest point minimization problem.

$$\begin{cases} \min J(\beta) = \| u - v \| \\ \text{s.t.} \quad u = \sum_{i \in I} \beta_i z_i; \quad \beta_i \geq 0, i \in I; \quad \sum_{i \in I} \beta_i = 1 \\ \qquad v = \sum_{j \in J} \beta_j z_j; \; \beta_j \geq 0, j \in J; \; \sum_{j \in J} \beta_j = 1 \end{cases} \tag{2.9}$$

Let (w^*, b^*) denote the solution of (2.2) and (u^*, v^*) be a solution of (2.9), we know that maximum margin of the boundary hyperplanes $= \frac{2}{\| w \|^*} = \| u^* - v^* \|$ and $w^* = \delta(u^* - v^*)$ for the specified δ. Substituting w^* into (2.1), it is easy to derive $\delta = \frac{2}{\| u^* - v^* \|^2}$ and then the following relationship between (w^*, b^*) and (u^*, v^*) can be achieved:

$$\begin{cases} w^* = \dfrac{2}{\| u^* - v^* \|^2} (u^* - v^*) \\ b^* = \dfrac{\| v^* \|^2 - \| u^* \|^2}{\| u^* - v^* \|^2} \end{cases} \tag{2.10}$$

Similarly, (2.4) can also be transformed to an equivalent problem similar to near point problem shown in (2.9).

A number of algorithms have been given for solving the (2.9), such as Gilbert's algorithm (Gilbert et al., 1988) and Mitchell-Dem'yanov-Malozemov algorithm (Mitchel et al., 1974). Recently, another fast and competitive algorithm, Iterative Nearest Point Algorithm (NPA), was proposed by Keerthi et al. (2000). The NPA consists of the following three steps:

1. Set $z = u - v$ from $u \in U, v \in V$;
2. Find an index k satisfying following condition:

$$\begin{cases} k \in I, then \quad -z \cdot z_k + z \cdot u \geq \dfrac{\varepsilon}{2} \| z \|^2 \\ k \in J, then \quad z \cdot z_k - z \cdot v \geq \dfrac{\varepsilon}{2} \| z \|^2 \end{cases} \tag{2.11}$$

where ε is the precision control parameter and satisfies $0 < \varepsilon < 1$. If such an index can not be found, stop with the conclusion that an approximate optimality criterion is satisfied; else go to step 3 with the k found;

3. Choose two convex polytopes, $\tilde{U} \subset U$ and $\tilde{V} \subset V$ such that $u \in \tilde{U}, v \in \tilde{V}$, $z_k \in \tilde{U}$ if $k \in I$, and $z_k \in \tilde{V}$ if $k \in J$. Compute $(\tilde{u}, \tilde{v})$ to be a pair of closest points minimizing the distance between $\tilde{U}$ and $\tilde{V}$. Set $u = \tilde{u}, v = \tilde{v}$ and go back to step 2.

After solving (2.9), the α_k in (2.4) can be achieved easily by computation, and then we obtain the following classifiers:

$$y(x) = sign(w \cdot \varphi(x) + b) = sign(\textstyle\sum_k \alpha_k y_k K(x, x_k) + b) \tag{2.12}$$

2.3 DOE-based Parameter Selection for SVM with NPA

Although SVM is a powerful learning method for classification problem, its performance is not only sensitive to the algorithm that solves the quadratic programming problem, but also to the parameters settings in the SVM formulation. In the process of using SVM, the first issue is to evaluate its parameter’s effectiveness. A obvious method is to train and test the SVM multiple times with keeping the number of training and testing samples at each iteration and then to take the average result of tests as the final index of performance for a single parameter setting. The second issue of using SVM is how to search the best parameter of SVM for a specified problem. An easy and reliable approach is to determine a parameter range, and then to make an exhaustive grid search over the parameter space to find the best setting. Since the evaluation of the performance for one parameter setting is time-consuming, even a moderately high resolution search may result in a large number of evaluations and the computational time will be unacceptable. Relative to grid search method, an alterative method, design of experiment (DOE), proposed by Staelin (2002) can reduce the complexity sharply. This approach is based on principles from design of experiment (DOE) and its main idea is as follows. First of all, a very coarse grid covering the whole search space is defined. Then both the grid resolution and search boundaries are iteratively refined until the stop criteria are satisfied. After each iteration, the search space is reset to be centered on the point with the best performance. If such a procedure causes the search to go beyond the user-given bounds, the center will be adjusted so that the whole space is contained within the user-given bounds (Staelin, 2002). An illustrative figure for DOE parameter search with two iterations is shown in Fig. 2.1.

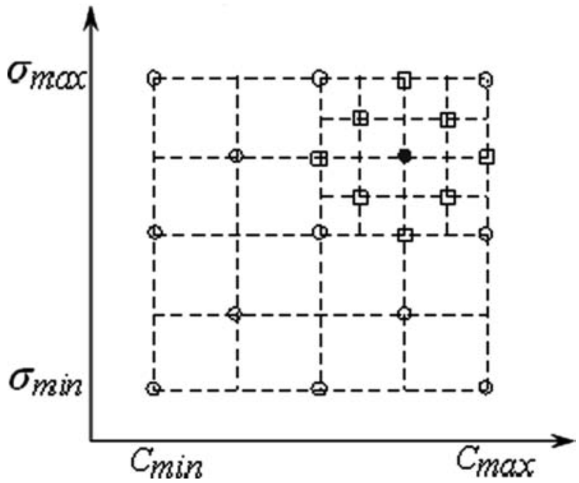

Fig. 2.1. An illustration for DOE parameter search with two iterations

Fig. 2.1 shows the points for evaluation within two iterations searching in method DOE in one possible case, circle denotes points in the first iteration, square denotes points in the second iteration. Roughly speaking, in this pattern (thirteen points per iteration), to achieve the 1×1 precision, the total number of points for evaluation is no more than 39. It can be observed that with the same precision, the grid search method can almost find the overall optimal solution in the searching space, while what the DOE method can find is a local optimal one. However, some experiments show the performance of final results from DOE methods is almost as good as the grid search (Staelin, 2002).

In this chapter, we use the DOE-based parameter search method to find optimal parameters of SVM with nearest point algorithm. For the SVM with NPA and Gaussian kernel function, there are two parameters that need to be tuned. One is $\tilde{C}$ shown in (2.6) and σ in Gaussian kernel function. The main steps of DOE approach to find the parameters for SVM with NPA and Gaussian kernel function are shown as follows

1. Set initial range for C and σ as [*C_min*, *C_max*], [*Sigma_min*, *Sigma_max*], *iter* = 1;
2. While *iter* ≤ *MAXITER* do
   2.1 According to the pattern as shown in Fig. 2.1, one can find the 13 points in the space with [*C_min*, *Sigma_min*] as left bottom corner and [*C_max*, *Sigma_max*] as the right upper corner, denoted by [(*C_min*, *Sigma_min*), (*C_max*, *Sigma_max*)], set *C_Space* = *C_max* – *C_min*, *Sigma_Space* = *Sigma_max* - *Sigma_min*;
   2.2 For each of the 13 points, evaluate its objective function *f*, choose the point *P*0(*C*0, *Sigma*0) with lowest objective function value to be the center, set new range of the searching space. *C_Space* = *C_Space*/2, *Sigma_Space* = *Sigma_Space*/2. If the rectangle area with (*C*0, *Sigma*0) as the center and *C_Space*, *Sigma_Space* as width and height exceeds the initial range, adjust the center point until the new search space is contained within the [(*C_min*, *Sigma_min*), (*C_max*, *Sigma_max*)].
   2.3 Set new searching space, *C_min* = *C*0 - *C_Space*/2, *C_max* = *C*0 + *C_Space*/2, *Sigma_min* = *Sigma*0 - *Sigma_Space*/2, *Sigma_max* = *Sigma*0 + *Sigma_Space*/2, *iter* = *iter* +1;
3. Return the point with best performance, select these best parameters to create a model for classification.

2.4 Experimental Analysis

In this section, we use a real-world credit dataset to test the effectiveness of the proposed SVM model with NPA algorithm and meantime make a comparison with other more than 20 methods.

The credit dataset used in this chapter 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 instance is 1000 including 700 creditworthy cases and 300 default cases. For each applicant, 20 kinds of attribute is available, such as account balance, credit history, loan purpose, credit amount, employment status, personal status and sex, age, housing, telephone status and job. To test the proposed algorithm, some categorical attributes need numerical attributes. A typical method for this requirement is to be coded as integers. This numerical form of dataset is used in our experiments.

In the experiment, the original range for $\log_2 \tilde{C}$ is [-5, 9], and that for $\log_2 \sigma$ is [-9, 5]. We set the *MAXITER* = 4, *N* = 10, the precision control parameter in the NPA $\varepsilon = 0.01$. The training samples are randomly chosen 2/3 total cases; the remaining is the testing samples.

Let the number of creditworthy cases classified as good be GG and classified as bad with GB, denote the number of default cases classified as good with BG and as bad with BB. Three commonly used evaluation criteria measure the efficiency of the classification, which is defined as follows:

$$\text{Good credit accuracy (GCA)} = \frac{GG}{GG+GB} \times 100\% \tag{2.13}$$

$$\text{Bad credit accuracy (BCA)} = \frac{BB}{BG+BB} \times 100\% \tag{2.14}$$

$$\text{Overall accuracy (OA)} = \frac{GG+BB}{GG+GB+BG+BB} \times 100\% \tag{2.15}$$

In addition, to compare the performance with some classification techniques, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), logistic regression (LOG), linear programming (LP), RBF kernel function LS-SVM (RBF LS-SVM), linear kernel function SVM (Lin LS-SVM), RBF kernel function SVM (RBF SVM), linear kernel function SVM (Lin SVM), (all SVM methods adopted a grid search mechanism to tune the parameter), neural networks (NN), naive Bayes classifier (NB), tree augmented naive Bayes classifiers (TAN), decision tree algorithm C4.5, C4.5rules, C4.5dis, C4.5rules dis, k-Nearest-neighbour classifiers KNN10 and KNN100; results of another 5 neural network models are selected from the work of West (2000), including the traditional multilayer perceptron (MLP), mixture of experts (MOE), radial basis function (RBF), learning vector quantization (LVQ) and fuzzy adaptive resonance (FAR), are conducted to the experiments. Accordingly, with the above setting and evaluation criteria, the experimental results are presented in Table 2.1. Note that the final optimal parameters in the proposed SVM with NPA, $\tilde{C}$ and σ, are 45.25 and 32 respectively. In Table 2.1, the partial reported computational results are directly taken from some previous studies, such as West (2000) and Baesens et al. (2003) for comparison purpose because we have the same size of training and testing sample for German credit dataset. Particularly, Baesens et al. (2003) reported the results of 17 classification methods and West (2000) provided results of 5 classification approaches on German credit dataset with different settings.

Table 2.1. Performance comparison of different methods on German credit dataset

Methods	Setting 1			Setting 2		
	OA	GCA	BCA	OA	GCA	BCA
LDA[a]	74.6	90.0	41.0	58.7	42.8	93.3
QDA[a]	71.0	79.5	52.4	71.0	79.5	52.4
LOG[a]	74.6	89.5	41.9	65.0	55.9	84.8
LP[a]	71.9	92.6	26.7	64.7	56.3	82.9
RBF-LS-SVM[a]	74.3	96.5	25.7	75.1	87.3	48.6
Lin LS-SVM[a]	73.7	91.3	35.2	58.7	42.8	93.3
RBF SVM[a]	74.0	92.6	33.3	63.5	54.6	82.9
Lin SVM[a]	71.0	95.6	17.1	66.8	58.5	84.8
NN[a]	73.7	85.2	48.6	70.4	68.6	74.3
NB[a]	72.2	87.8	38.1	62.6	52.4	84.8
TAN[a]	72.5	87.3	40.0	63.2	51.5	88.6
C4.5[a]	72.2	88.2	37.1	68.9	68.1	70.5
C4.5rules[a]	71.0	91.3	26.7	45.5	27.5	84.8
C4.5dis[a]	74.6	87.3	46.7	64.1	57.6	78.1
C4.5rules dis[a]	74.3	89.5	41.0	74.3	89.5	41.0
KNN10[a]	70.7	94.8	18.1	41.3	17.0	94.3
KNN100[a]	68.6	100	0.00	62.0	52.0	83.8
MOE[b]	75.64	85.72	52.25	77.57	86.99	55.43
RBF[b]	74.60	86.53	47.01	75.63	85.76	51.79
MLP[b]	73.28	86.48	52.47	75.04	87.09	46.92
LVQ[b]	68.37	75.07	51.84	71.20	79.15	55.20
FAR[b]	57.23	59.61	51.17	62.29	70.92	41.86
SVM+NPA+DOE[c]	77.46	89.49	49.30	79.64	90.46	54.33

[a] Results from Baesens et al. (2003), setting 1 assumes a cutoff of 0.5, setting 2 assumes a marginal good-bad rate around 5:1

[b] Results from West (2000), setting 1 results are an average of 10 repetitions, setting 2 are an average of three best from the 10 repetitions.

[c] Results of SVM+NPA+DOE in setting 1 are an average of 10 repetitions with the optimal parameters, results in setting 2 are an average of three best from the 10 repetitions.

As can be seen from Table 2.1, we can find the following several conclusions:

(1) The overall credit accuracy of the proposed SVM method with NPA and DOE for parameter optimization is the best of all the list approaches under all settings, followed by MOE neural network model, indicating that the proposed method is an effective credit risk assessment method. The main reason leading to performance improvement is that the design of experiment (DOE) method finds optimal parameters of SVM.

(2) The good credit accuracy and bad credit accuracy of proposed SVM with NPA and DOE is not the best of all the list approaches, but it is still on the top level, implying that the proposed SVM with NPA and DOE has good classification capability in credit risk evaluation problem.

(3) The reported results of the proposed SVM with NPA and DOE in setting 2 obviously better than the results from the proposed SVM with NPA and DOE in setting 1, indicating that the combination of best several results has good generalization than that of all results. This implies the necessity of result selection.

(4) One possible reason resulting in good performance for the proposed SVM with NPA and DOE may be that the proposed SVM with NPA and DOE reduces the computational complexity for the instances. First of all, the fast iterative NPA is used for solving the original quadratic programming problem and thus improving computational efficiency. Second, the search space is constructed with the principles of DOE instead of grid search and thus reducing the computational time. Especially for the application of more complex kernel function with more parameters, the DOE can reduce more computational time with relative to grid search.

2.5 Conclusions

In this chapter, the parameter optimization-based SVM method with radial basis function is used to evaluate the credit risk. In this SVM method, a recently fast iterative nearest points algorithm (NPA) is adopted for solving the original quadratic programming indirectly and a parameter selection technology based on design of experiment principles is used. For illustration and verification, a real-world credit dataset are used and some experiments are performed. The results obtained show that the proposed SVM with NPA and DOE outperforms all the other 22 methods listed in this chapter based on the measure of overall classification accuracy. This indicates SVM with NPA and DOE can provide a promising solution to credit risk assessment.

Though the experiments show that the proposed SVM with NPA and DOE is rather promising, the results reported here are still far from sufficient to generate a determinate statement about the performance of SVM with NPA and DOE. Since the performance is also dependent on the characteristic of the datasets, future research are encouraged to identify the performance of proposed SVM with NPA and DOE and explore the other effective kernel function in this framework.

Springer

http://www.springer.com/978-3-540-77802-8

Bio-Inspired Credit Risk Analysis
Computational Intelligence with Support Vector Machines

Yu, L.; Wang, S.; Lai, K.K.; Zhou, L.

2008, XVI, 244 p., Hardcover
ISBN: 978-3-540-77802-8