Relevance vector machine based infinite decision agent ensemble learning for credit risk analysis

Shukai Li a,⇑, Ivor W. Tsang a, Narendra S. Chaudhari b

a Centre for Computational Intelligence, Nanyang Technological University, Singapore
b Department of Computer Science and Engineering, Indian Institute of Technology Indore, India

ARTICLE INFO

Keywords:
Credit risk analysis
Boosting
Relevance vector machine
Perceptron Kernel

ABSTRACT

In this paper, a relevance vector machine based infinite decision agent ensemble learning (RVMIDEA) system is proposed for the robust credit risk analysis. In the first level of our model, we adopt soft margin boosting to overcome overfitting. In the second level, the RVM algorithm is revised for boosting so that different RVM agents can be generated from the updated instance space of the data. In the third level, the perceptron Kernel is employed in RVM to simulate infinite subagents. Our system RVMIDEA also shares some good properties, such as good generalization performance, immunity to overfitting and predicting the distance to default. According to the experimental results, our proposed system can achieve better performance in term of sensitivity, specificity and overall accuracy.

1. Introduction

Credit risk, the chance that money owed may not be repaid. There is little doubt, however, that the awareness of credit risk has continued to grow. This has been accompanied by an increasing recognition across many sectors of the economy that credit risk needs to be actively managed (Servigny & Renault, 2004). People pay high attention to the potential loss of credit assets in the future, such as: changes in the credit quality (including downgrades or upgrades in credit ratings), variations of credit spreads, and the default event. The role of credit risk analysis is to assess and evaluate the potential credit risk with any customer or borrower, and to advise on decisions about granting credit or providing loans or borrowing facilities (Graham & Coyle, 2000). In other words, credit risk analysis is the method by which one calculates the creditworthiness of a person, business or organization. For many credit granting institutions like commercial banks and credit companies, the ability to discriminate non-default customers from default ones is crucial for the success in their business. Credit risk analysis has become to attract much more attention from financial institutions because of the Asian Financial Crisis in 1997, the subprime mortgage crisis during 2007 and 2009, and Basel II (Lang, Mester, & Vermilyea, 2008) published in 2004. Furthermore, as business competitions for more market share and profit become more and more serious, some financial institutions undertake more risks to achieve competitive superiority in the market.

Accessibility of large databases, and advances in statistical and machine learning methods to generate efficient credit risk models have changed this area fundamentally in last decades. The prediction of credit risk has also been widely studied after realizing its practical purposes like early warning signals for defaults by obligors.

This kind of techniques are widely applied in the corporate and personal credit risk analysis in which there is a need to predict the credit risk of a potential obligor before the debit is approved and extended. Besides that, financial institutions are driven by obligees to employ powerful credit risk models to assess the credit risk of debit. Hence, more accurate quantitative models for prediction are essential in order to perform more accurate credit risk analysis of loan portfolios and access the obligor’s creditworthiness.

1.1. The state of the art

Credit risk analysis is important but also complicated. The most reliable customer may also default his or her debt. Besides that, there are some noisy data from corporate financial statements and personal credit question forms. In order to generate a robust credit risk analysis model, this line of research has began since 1960s, like Beaver (Beaver, 1966) who is one of the earliest researchers to study the prediction of credit risk. Beaver’s analysis includes studying one financial ratio at each time and deciding a cutoff threshold for every ratio. Hereafter, quantitative models (Galindo & Tamayo, 2000; Thomas, 2000) such as linear discriminant analysis and logistic regression (Ederington, 1985) have been applied to predict the credit level of new clients. In addition to
these traditional statistical methods, machine learning techniques, such as rule based reasoning systems (Kim, 1993) and neural networks (Maher & Sen, 1997), were adopted to improve the prediction accuracy around 1990s. Investigations of machine learning methods and their experiments revealed that such methods normally reach higher accuracy than traditional statistical methods (Kim, 1993). Furthermore, hybrid methods (Lin, 2009) and support vector machine (SVM) (Huang, Chen, & Wang, 2007) are also employed in this area recently.

Among various credit risk models, Chen and Shih (2006) and Huang et al. (2004) reported that SVM was competitive and outperformed other classifiers (including neural networks and linear discriminant classifier) in terms of generalization performance. Due to the good performance of SVM, our model will mainly compare with it in our experiments. Furthermore, in order to further improve the generalization performance of existing models, ensemble learning is adopted to enhance them, like neural network and SVM based ensemble learning methods (Yu, Wang, & Lai, 2008, 2010).

1.2. The preferred properties for credit risk modeling

The primary focus of credit risk analysis is to improve the prediction accuracy. In particular, sensitivity (SE) and specificity (SP) are the common performance measures for this task. If SE is low, banks will lose some non-default customers, which will lower down their interest income in the income statement. If SP is low, it will lead to more default, which will write down the provision in the income statement. Therefore, for credit risk analysis, we should analyze the SE and SP separately (Baesens et al., 2003), not just pay attention only to overall accuracy. Moreover, the credit risk model should also pay attention to some special groups of customers which are difficult to classify. If we can catch this kind of customers, it will improve the generalization performance significantly. Besides that, any customer may default due to various reasons, and even non-default customers have probability to default. So the credit risk model should also predict distance to default (DD) (Servigny & Renault, 2004), which means the probabilities for default.

For some machine learning models, like the neural networks, we need to adjust the model structure and parameters, which often increases the complexity. Although neural networks are increasingly found to be powerful in many classification applications, the performance is actually dependent on network model itself, especially on initial conditions, network topologies and training algorithms, which may be one reason why the results of neural networks for credit risk evaluation varies when compared with some statistical methods. To find the optimal neural net model is still a challenging issue (Huang, Chen, Hsu, Chen, & Wu, 2004). If the credit model structure is not relatively stable and has many free parameters to adjust, it will be inconvenient for financial institutions. Therefore, for the consideration of practical usage, the stable structure of the model is preferred.

Besides that, there is some inaccurate information, both in the personal credit data from application forms and corporate financial statements, which lead to some noisy data and cause overfitting (Steckling & Schesch, 2003) in the learning process. Thus, the credit risk model should have a relatively stable structure, good generalization performance, predict Distance to Default, and also overcome overfitting.

1.3. The advantages of our model

Relevance vector machine (RVM) (Tipping, 2001) is a Bayesian sparse Kernel technique for regression and classification which has not been used in credit risk analysis and overcome the limitations of existing methods. It shares the good characteristics of SVM while avoiding some limitations of SVM. For instance, RVM also provides clear connections to the underlying statistical learning theory. The RVM algorithm usually finds a globally optimal solution, and has a simple geometric interpretation (Bishop & Tipping, 2000).

Compared with SVM, the advantages of RVM includes: (1) The number of relevance vectors for RVM is much fewer than the number of support vectors for SVM. (2) Its prediction is probabilistic, which is used to estimate Distance to Default (DD). (3) Unlike SVM, there is no need to estimate the soft margin parameter ‘C’, and even the Kernel parameters are optimized in the learning process of RVM. (4) The Kernel function does not need to satisfy Mercer’s condition (Tipping, 2001). However, to the best of our knowledge, until now RVM has not been used in credit risk analysis. Additionally, RVM typically leads to much sharper models resulting in faster performance on test data while without increasing the error.

In order to further improve the prediction performance of RVM, we apply ensemble learning method to improve RVM. Ensemble learning makes use of many base agents or their different variants to resolve the problem. Ensemble learning usually generalizes better than a single agent. One of the most common and effective ensemble learning frameworks is Adaboost (Freund & Schapire, 1996), which often leads to overfitting. In this paper, we adopt soft margin boosting (Ratsch & Onoda, 2001) to overcome this drawback. Traditional boosting methods often use instance weights to generate diverse input data and make the algorithms focus on the instances which are difficult to classify. In this way, we can separate some special groups of customers which are difficult to classify. In our model, we also adopt instance weights to achieve the above purpose. The instance weights are incorporated to the RVM agent objective function. Therefore, the agent will focus on the high weight instances. Besides that, in each RVM agent, we employ the perceptron Kernel (Lin & Li, 2008) to simulate infinite sub-agents, which forms a three-level ensemble learning systems named RVM based infinite decision agent ensemble learning (RVMидеал) system.

The rest of this paper is organized as follows. Section 2 gives a review on the agent RVM. Section 3 then describes the proposed framework for credit risk analysis. Experimental results are presented in Section 4, and the last section gives some concluding remarks.

For simplicity, throughout the remaining of this paper, we consider a two-class classification problem - non-default and default customers, in which the training data \( \mathbf{Z} = \{ \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n \} \) comprises feature vectors \( \mathbf{X} = \{ \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \} \) along with corresponding binary target variables \( \mathbf{y} = \{ y_1, y_2, \ldots, y_n \} \). \( y_i = 1 \) and \( y_i = -1 \) stand for non-default and default customers respectively. In the agent level, \( y_i = 1 \), \(-1\) is mapped to \( y_i = 1 \), 0 for the need of RVM. Each instance \( \mathbf{x}_i \) is composed of \( m \) features \( \{ x_{i1}, x_{i2}, \ldots, x_{im} \} \) and a weight \( w_i^{(n)} \) in iteration \( n \) for this instance. The base agent we obtained from each iteration is written as \( h_\theta \). Moreover, the operator \( \odot \) means elementwise product.

2. RVM review

Relevance vector machine (RVM) is a probabilistic Bayesian learning framework. It acquires relevance vectors and weights by maximizing a marginal likelihood. The structure of the RVM is described by the sum of products of weights and Kernel functions as follows:

\[
y(\mathbf{X}) = \mathbf{a} \cdot \phi(\mathbf{X}) = \sum_{i=1}^{n} a_i K(\mathbf{x}_i, \mathbf{x}_i) + a_0
\]

where \( a = [a_0, a_1, a_2, \ldots, a_n] \) and \( \phi(\mathbf{X}) = [1, K(\mathbf{x}_1, \mathbf{x}_1), K(\mathbf{x}_1, \mathbf{x}_2), \ldots, K(\mathbf{x}_{n}, \mathbf{x}_n)]' \). The likelihood of the training data is
The posterior distribution for weight \( a \) as follows:

\[
p(a|\mathbf{y}, \sigma^2) = \frac{p(\mathbf{y}|a, \sigma^2)p(a|\mathbf{x})}{p(\mathbf{y}|\mathbf{x})}
= (2\pi)^{-n/2}\sigma^2 \exp\left(-\frac{1}{2}(a-\mu)^T\Sigma^{-1}(a-\mu)\right)
\]

where the posterior covariance and mean are respectively:

\[
\Sigma = (\sigma^2 \Phi^T \Phi + A)^{-1}
\]

\[
\mu = \sigma^2 \Sigma \Phi^T \mathbf{y}
\]

with \( A = \text{diag}(\sigma_0, \sigma_1, \ldots, \sigma_n) \) and \( \Phi = [\phi(x_1), \phi(x_2), \ldots, \phi(x_n)]^T \). The marginal likelihood is also computable,

\[
p(\mathbf{y}|\mathbf{x}, \sigma^2) = \int p(\mathbf{y}|a, \sigma^2)p(a|\mathbf{x})da
= (2\pi)^{-n/2}\sigma^2 \exp\left(-\frac{1}{2}(\mu - \mathbf{y})^T \Sigma^{-1}(\mu - \mathbf{y})\right).
\]

The derivatives of the marginal likelihood is set to zero and the re-estimation equations is as follows:

\[
\gamma_i^{new} = \frac{\gamma_i}{\beta_i^{2}}
\]

\[
(\sigma^2)^{new} = \frac{\|\mathbf{y} - \Phi \mu\|^2}{n - \sum_i \beta_i^2}
\]

where \( \gamma_i \) is the ith diagonal component of the posterior covariance \( \Sigma \) given by (5). Learning therefore proceeds by choosing initial values for \( \mathbf{a} \) and \( \sigma^2 \), evaluating the mean \( \mu \) and covariance \( \Sigma \) of the posterior using (6) and (5), respectively. Then the hyperparameters \( \mathbf{a} \) and \( \sigma^2 \) are alternately re-estimated, using (8) and (9). This process is repeated until a suitable convergence criterion is satisfied.

3. Methodology formulation

In this section, we present our proposed RVM\(_{\text{ideal}}\) approach for credit risk analysis. The architecture of our proposed framework is depicted in Fig. 1. It consists of three levels. In the base level, we make use of the perceptron Kernel to simulate infinite subagents. Then in the middle level, the RVM algorithm using the perceptron Kernel is performed to train weak learner \( h_t \), and the error function is computed and sent to boosting. In the top layer, boosting algorithm is used to update the instance weights and sends them back to the RVM agents. Then in the next iteration, a new RVM agent is trained based on the updated instance weights. This process is repeated until convergence. At the end, the final hypothesis makes prediction based on the ensemble of the prediction functions of all agents. The details of this framework are given as follows.

3.1. Base agent creation

Boosting is a powerful technique for combining multiple learning agents to produce a form of committee whose performance is usually significantly better than that of any single base agent. Boosting can also give good results even if the base agents have the performance that is only slightly better than random (Freund & Schapire, 1996), and hence the base agents are known as weak learners.

3.1.1. Instance diversity

In the basic form of boosting, the base agents are trained in a sequence, and each base agent is trained using a weighted form of the data set in which the weighted coefficient associated with each instance depending on the performance of the previous agents. In particular, instances that are misclassified by one of the base agents are given greater weight when used to train the next agent in the sequence. Once all the agents have been trained, their predictions are then combined through a weighted majority voting scheme.

The instance weight \( w_i \) is initially set as \( 1/n \) for all instances. We assume that we have a procedure available for training a base agent using weighted data through the given function. At each
stage of the algorithm, boosting trains a new agent using the data set in which the weighting coefficients are adjusted according to the performance of the previously trained agent, so as to give greater weights to the misclassified instances.

Comparing the true label \( y_i \) and the prediction \( h_t(x_i) \) given by the \( t \)th agent, we define the margin function as

\[
p(x) = f_t(h_t(x_i), y_i)
\]

(14)

And based on the margin function, the weight update equation is defined as

\[
w_{t+1} = f_t(w_t, p)
\]

(15)

The instance weights are changed in different iterations based on the margin function in the former iteration, and the detailed form of (14) and (15) will be given in Section 3.3.

3.1.2. Infinite decision subagents via perceptron Kernel

The base agent RVM is a Kernel method, so we need to select a proper Kernel fitting for both RVM and boosting. The most common used Kernel functions are polynomial, radial basis and hyperbolic tangent functions, none of which take consideration for the needs of ensemble learning.

Recently, there is some research (Lin & Li, 2005) done on the relationship between Kernel methods and ensemble methods. Although the number of agents can be infinite in theory, most existing models only utilize a small number of finite agents, which limit the capacity the number of agents can be infinite in theory, most existing models, and (11) becomes

\[
\ln[p(y|a)p(a|x)] = \sum_{i=1}^{n} w_i [y_i \ln \eta(y(x_i)) + (1 - y_i) \ln(1 - \eta(y(x_i)))]
\]

and (11) becomes

\[
\]

Here, the first term is the sum of errors, and the second is the regularizer. (21) is maximized to find the \( a \). In the process, the instance with larger weight is assigned correct label first.

The first and second derivatives of the log posterior distribution are then given by

\[
\nabla \ln[p(y|a)p(a|x)] = (\Phi \circ \omega)(y - y(x)) - a \Phi
\]

(22)

\[
\nabla^2 \ln[p(y|a)p(a|x)] = -[(\Phi \circ \omega)' \Phi + a]
\]

(23)

where \( y(x) = [y(x_1), y(x_2), \ldots, y(x_n)]' \) and \( \Phi = \text{diag} (\beta_1, \beta_2, \ldots, \beta_n) \) with \( \beta_i = \sigma(y(x_i)) [1 - \sigma(y(x_i))] \). The negative second derivative represents the inverse covariance matrix for the Gaussian approximation to the posterior distribution. The mode of the resulting approximation to the posterior distribution, corresponding to the mean of the Gaussian approximation, is obtained by setting (22) to zero, and giving the mean and covariance of the Laplace approximation in the form

\[
a_{\text{LAP}} = \Phi^{\dagger} (\Phi \circ \omega)' (y - y(x))
\]

(24)

\[
\Sigma = -((\Phi \circ \omega)' \Phi + a)^{-1}.
\]

(25)

Now we use this Laplace approximation to evaluate the marginal likelihood as

\[
p(y|x) = \int p(y|a)p(a|x)da
\]

\[
\approx p(y|a_{\text{LAP}})p(a_{\text{LAP}}) (2\pi)^{-n/2} |\Sigma|^{-1/2}
\]

(26)

If we substitute for \( p(y|a_{\text{LAP}}) \) and \( p(a_{\text{LAP}}) \) and then set the derivative of the marginal likelihood with respect to \( y \) equal to zero, we obtain

\[
\gamma_i = \frac{1}{w_{\text{LAP}}}
\]

(27)

where \( \gamma_i = 1 - x_i \Sigma x_i \). This update formula is identical to the (8). Hence, the remain steps are the same as the RVM in regression case. The algorithm alternately re-estimates the posterior mean and covariance, using (24) and (25), and re-estimates the hyperparameters, through (27), and until a suitable convergence criterion is satisfied.

3.3. Multiagent ensemble learning

In order to avoid overfitting and get a good generalization performance, we introduce soft margin to boosting, which is the same as Ratsch did (Ratsch & Onoda, 2001). First we define the margin for Eq. (14) as

\[
\rho(x, c) = yf(x) = y \sum_{i=1}^{t} c_i h_i(x_i).
\]

(28)

In Adaboost, after many iterations, it satisfies the inequalities

\[
\rho(x, c) \geq Q \quad (i = 1, 2, \ldots, n)
\]

(29)

If \( Q > 0 \), all the instances are classified according to their possibly wrong labels, which leads to overfitting in the presence of noise. Therefore, we relax the margin,

\[
\hat{\rho}(x, c) = \rho(x, c) + \mu_{t+1}(x)
\]

(30)

where \( C \) is a priori chosen constant and \( \mu_{t+1}(x) = \sum_{i=t}^{t+1} c_i h_i(x) \). Based on Ratsch's result, the weight update function of Eq. (15) becomes

\[
w_{t+1} = \exp \left\{ - \frac{1}{2} \left[ \rho(x, b') + C |b'\mu_{t+1}(z) | \right] \right\}
\]

(31)
where \( b_i = \text{argmin}_{b_i \geq 0} \sum_{t=1}^{n} \exp \left\{ -\frac{1}{2} \left[ \phi(z_t, b^i) + C \| h_i(z_t) \| \right] \right\} \) with \( b^i = [b_1, b_2, \ldots, b_t] \), and \( Z_i \) is the normalization constant, such that \( \sum_{i=1}^{n} w_t(z_i) = 1 \).

In this way, the margin is computed based upon the \( h_t \) and the ground true \( y \), and it is relaxed to get the new soft margin by adding regularizer. If there are noisy data, it limits the increase of the weights. RVM is retrained based on the updated instance space. In our newly formalized RVM model, the instances with higher weights will be correctly classified first. Then we gain a new prediction function \( h_{t+1} \). This process is repeated until convergence. Finally, it has run for \( T \) iterations, and then we get final hypothesis as

\[
f(x) = \sum_{i=1}^{T} c_i h_t \quad \text{where} \quad c_i = \frac{b_i}{\sum_{i=1}^{T} |b_i|}
\]

In order to verify the effectiveness of the proposed model, the real world credit card approval experiments are conducted and analyzed.

4. Experiments

In this Section, two credit card application approval data sets are adopted, which are Australian and Japanese data sets from UCI machine learning repository (http://www.archive.ics.uci.edu/ml/datasets.html).

In the Australian credit application data set, there are 307 instances of creditworthy applicants and 383 instances whose credit is not creditworthy. Each sample is characterized by 14 attributes including 6 numerical and 8 categorical attributes, and one target attribute (non-default or default). All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. There are 37 instances with missing values. The missing ones are replaced by either the mode of the attribute if it is a categorical attribute or the mean of the attribute if it is continuous.

In the Japanese consumer credit card application approval data set, all attribute names and values have also been changed to meaningless symbols for confidentiality. This time we delete the data with missing attribute values, and then we obtain 653 data with 15 features, in which 357 cases were granted credit and 296 cases were refused.

4.1. Experimental setup

Here, 1/2 of the instances is for the training set, and 1/4 is for validation and testing set respectively. In the credit risk analysis, detecting non-default and default customers are both important. If we do not separate the default customers, it will lead to default. Otherwise, if we predict non-default customers wrongly, we will lose non-default customers, which loses down our interest. Therefore, we use the following measures,

\[
SE = \frac{\text{Number of true positives}}{\text{Number of true positives} + \text{Number of false negatives}}
\]

\[
SP = \frac{\text{Number of true negatives}}{\text{Number of true negatives} + \text{Number of false negatives}}
\]

\[
BA = \frac{SE + SP}{2}
\]

Table 1: Testing accuracy (%) on Australian credit data set.

<table>
<thead>
<tr>
<th>Method</th>
<th>BA</th>
<th>SE</th>
<th>SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGR</td>
<td>86.5±2.2</td>
<td>91.1±2.8</td>
<td>81.8±3.2</td>
</tr>
<tr>
<td>SVM</td>
<td>86.3±1.2</td>
<td>89.5±1.4</td>
<td>83.2±1.8</td>
</tr>
<tr>
<td>RVM</td>
<td>88.2±1.5</td>
<td>92.6±1.6</td>
<td>83.8±2.2</td>
</tr>
<tr>
<td>RVMAda (T=100PK)</td>
<td>90.5±1.0</td>
<td>94.7±1.4</td>
<td>86.3±1.5</td>
</tr>
<tr>
<td>RVMAda (T=200PK)</td>
<td>90.6±1.0</td>
<td>94.8±1.2</td>
<td>86.3±1.5</td>
</tr>
<tr>
<td>RVMAda (T=100GK)</td>
<td>93.7±1.2</td>
<td>97.2±1.7</td>
<td>90.2±1.8</td>
</tr>
<tr>
<td>RVMAda (T=200GK)</td>
<td>93.8±1.2</td>
<td>97.3±1.7</td>
<td>90.2±1.8</td>
</tr>
<tr>
<td>RVMAda (T=100PK)</td>
<td>95.5±1.0</td>
<td>98.4±1.6</td>
<td>92.7±1.4</td>
</tr>
<tr>
<td>RVMAda (T=200PK)</td>
<td>95.5±1.0</td>
<td>98.4±1.6</td>
<td>92.7±1.5</td>
</tr>
</tbody>
</table>

Table 2: Testing accuracy (%) on Japanese credit data set.

<table>
<thead>
<tr>
<th>Method</th>
<th>BA</th>
<th>SE</th>
<th>SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGR</td>
<td>74.6±4.3</td>
<td>75.8±5.6</td>
<td>73.4±6.2</td>
</tr>
<tr>
<td>SVM</td>
<td>78.3±4.1</td>
<td>80.4±5.1</td>
<td>76.2±6.1</td>
</tr>
<tr>
<td>RVM</td>
<td>79.5±3.4</td>
<td>82.2±4.3</td>
<td>76.7±4.8</td>
</tr>
<tr>
<td>RVMAda (T=100PK)</td>
<td>83.2±3.6</td>
<td>84.3±4.7</td>
<td>82.1±5.1</td>
</tr>
<tr>
<td>RVMAda (T=200PK)</td>
<td>83.2±3.6</td>
<td>84.3±4.6</td>
<td>82.1±5.1</td>
</tr>
<tr>
<td>RVMAda (T=100GK)</td>
<td>86.3±3.3</td>
<td>89.3±4.3</td>
<td>83.3±4.7</td>
</tr>
<tr>
<td>RVMAda (T=200GK)</td>
<td>86.3±3.3</td>
<td>89.3±4.4</td>
<td>83.3±4.7</td>
</tr>
<tr>
<td>RVMAda (T=100PK)</td>
<td>88.0±3.3</td>
<td>91.5±4.2</td>
<td>84.6±4.9</td>
</tr>
<tr>
<td>RVMAda (T=200PK)</td>
<td>88.0±3.3</td>
<td>91.5±4.2</td>
<td>84.6±4.9</td>
</tr>
</tbody>
</table>

consider the capacity to detect both non-default and default customers.

The training, validation and testing set are chosen randomly, all the methods are run 20 times and the average performances are evaluated to reduce statistical varieties. Finally, we give the prediction accuracy based on the testing set, which are shown in Tables 1 and 2. The best performance is listed in bold.

4.2. Compared methods

We first compare our model RVM based infinite decision agent (RVMideal) system with logistic regression (LOGR), SVM and RVM. LOGR is one of the best statistical credit risk models, and SVM is among the best machine learning credit risk models. Thus, these two models are typical baseline methods used in credit risk analysis.

For SVM, we need to choose regulation parameter \( C \) and the Gaussian Kernel parameter \( \sigma \). \( C \) is selected in the range of \([0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000]\). In particular, the width \( \sigma \) of the Gaussian Kernel exp \((-|z|^2/2\sigma^2)) \) is picked via \((0.25, \sqrt{\gamma}, 0.5\sqrt{\gamma}, \sqrt{\gamma}, 2\sqrt{\gamma}, 4\sqrt{\gamma}) \) where \( \gamma \) is the average distance from all pairs of instances. For RVM, all the parameters including the parameter in the Gaussian Kernel is adaptive by itself. Therefore, no validation is needed for RVM. Besides that, we also compare with RVM based Adaboost (RVMAda). For the soft margin boosting, the regularization parameter \( C \) is selected from \([0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000]\). We also adopt the Gaussian Kernel (GK) and the perceptron Kernel (PK) individually for our model.

4.3. Analysis of experimental results

We compare the BA, SE and SP in Tables 1 and 2. In addition, the test error \((1 - BA), (1 - SE) \) and \((1 - SP) \) are presented for different number of agents in Fig. 2. Based on the above result, several important conclusions are drawn in the following,

(a) For the single agent, it is obvious that the LOGR, SVM and RVM perform generally the same in Australian credit data set, while SVM and RVM are better in Japanese case. For
RVM and SVM, they are both Kernel methods, but RVM does not need to search the optimal values for the model parameters. All parameters in RVM are optimized by itself, even including the parameters for the Gaussian Kernel.

(b) For the ensemble models, due to the ensemble output of boosting (Freund & Schapire, 1996), they can improve the prediction performance of a single agent significantly. Thus, the ensemble models RVM_{Ideal} and RVM_{Ada} are better than the single agent models LOGR, SVM and RVM. Moreover, the performance of RVM_{Ada} is inferior to that of RVM_{Ideal}. It is possibly because RVM_{Ada} does not have soft margin, which will lead to overfitting, i.e. inferior prediction performance on testing set.

(c) Moreover, we compare our model with the Gaussian Kernel and the perceptron Kernel. From the experimental result in Tables 1 and 2, RVM_{Ideal}(PK) performs the best in term of the BA, SE and SP. The possible reason is that the perceptron Kernel is equal to infinite subagents, which fit for the
ensemble learning better. In addition, there is no Kernel parameter in the perceptron Kernel needed to be optimized; while the Kernel parameter in the Gaussian Kernel is optimized through the learning process.

(d) Furthermore in Fig. 2, for all the ensemble methods, it is obvious that the slope of $RVM_{\text{Ideal}}(PK)$ is higher than $RVM_{\text{Auto}}$, and the slope of $RVM_{\text{Ideal}}$ is higher than $RVM_{\text{Auto}}$, which means the $RVM_{\text{Ideal}}(PK)$ converges most quickly in the three ensemble models, because of infinitely many perceptrons and its immunity from overfitting. The slope often decreases when $T$ grows bigger, and it nearly converges to 0 after $T = 80$. We also find there is nearly no difference from $T = 100$ to $T = 200$. Thus, it is proper to adopt the model at $T = 100$.

5. Conclusion

In this paper, we first give a brief survey of credit risk analysis and its employed models, and also compare advantages and disadvantages for different kinds of credit risk models. Furthermore, we summarize the preferred properties of credit risk modeling. Based on these properties and existing machine learning techniques, we propose our model $RVM_{\text{Ideal}}$. For the single agent, we mainly compare RVM with SVM. We also employ the perceptron Kernel to simulate the infinite subagents. In this way, our system becomes a three-level ensemble learning system. Moreover, we also update the objective function and the related functions in each RVM agent to fit the ensemble learning system. In order to better ensemble the agents, we adopt soft margin boosting to overcome overfitting.

We also perform comprehensive experiments on two credit data sets. Our proposed model outperforms existing statistical and machine learning methods. Besides the good accuracy, all the parameters in the agent level are optimized by itself. Based on the good property of RVM, we can also compute DD for each instance. All in all, our model $RVM_{\text{Ideal}}$ has a stable structure, good generalization performance, can overcome overfitting and predict DD in the same time.

References


