Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 36 (2009) 2625–2632
Expert Systems with Applications
www.elsevier.com/locate/eswa

Credit scoring algorithm based on link analysis ranking with support vector machine

Xiujuan Xu *, Chunguang Zhou, Zhe Wang

College of Computer Science, Key Laboratory of Symbol Computation and Knowledge, Engineering of the Ministry of Education, Jilin University, Changchun 130012, China

Abstract

Credit scoring is very important in business, especially in banks. We want to describe a person who is a good credit or a bad one by evaluating his/her credit. We systematically proposed three link analysis algorithms based on the preprocess of support vector machine, to estimate an applicant's credit so as to decide whether a bank should provide a loan to the applicant. The proposed algorithms have two major phases which are called input weighted adjustor and class by support vector machine-based models. In the first phase, we consider the link relation by link analysis and integrate the relation of applicants through their information into input vector of next phase. In the other phase, an algorithm is proposed based on general support vector machine model. A real world credit dataset is used to evaluate the performance of the proposed algorithms by 10-fold cross-validation method. It is shown that the genetic link analysis ranking methods have higher performance in terms of classification accuracy.

Keywords: Credit scoring; Link analysis ranking algorithm; Support vector machine

1. Introduction

Recently, data mining has developed rapidly over last several years, which has expanded to business and finance (Ester, Ge, Jin, & Hu, 2004; Kleinberg, Papadimitriou, & Raghavan, 1998), especially credit industry (Thomas & Edelman, 2002). For banking institutions, loans are often the primary source of credit risk. To evaluate the risk of these loans, banks use credit scoring models and credit rating to estimate default risk on a single loaner basis (Chen & Huang, 2003). However, with the intense competition of credit card issues and banks, more and more people can have credit cards and get loans from banks which have not check the applicants' credit status thoroughly.

Furthermore, many different algorithms have been proposed in pervious literature of credit scoring. The credit scoring models are developed to categorize applicants as either accepted or rejected with considering the applicants' characteristics such as age, income, and marital condition. Credit scoring is a basic binary classification task in finance. Many studies have contributed to increasing the accuracy of the classification model with various kinds of statistical tools. With the rapid growth in the credit, credit scoring models with low discriminatory power can lead to underpricing of bad and overpricing of good loans (Blochlinger & Leippold, 2006). So credit scoring need high accuracy to avoid bad debts.

Our contributions are as follows. The paper introduces a link analysis-based support vector machine (SVM) method to classify the applicants who apply for new credit cards or loans to banks. The novel approach has two phases for credit scoring. In the first phase, we present a new applicant's matrix to find the representative applicants and then find important information from the matrix by link analysis method. We compute the scoring for every applicant to distinguish good applicants and bad applicants. We call the new matrix "co-information matrix" (CIM), so that we can find the elements which are important for influence.

* Corresponding author.
E-mail addresses: xuxiujuan666@yahoo.com.cn (X. Xu), cgzhou@jlu.edu.cn (C. Zhou), wz2000@jlu.edu.cn (Z. Wang).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.01.024

We proposed the interpretation of the new model. In the second phase, we use general support vector machines (SVM) model with new input feather space.

The proposed credit scoring model is a hybrid approach using link analysis ranking techniques to preprocess samples into weighted information, and SVM techniques to build classifiers. The ranking process depends on information's values. In the SVM stage, SVM creates models to classify applicants.

The paper is organized as follows: the next section reviews some relate work about link analysis ranking and SVM in credit scoring. Sections 3 and 4 deal with the main contribution of this study. Section 3 gives a link analysis ranking model for credit data. Section 4 gives a main frame for SVM. Section 5 demonstrates the empirical evaluation including the experimental setup and the results. Conclusions are drawn in Section 6.

2. Relate works

First we describe some of the previous link analysis ranking literature (Getoor & Diehl, 2005), which has been improved and extended many aspects over several years. With the development of the Internet, information retrieve technique of search engine has developed rapidly. Link analysis technique is grown based on topology structure of the web. Link analysis ranking algorithm has exploited from web page ranking to the other fields.

Then we review some related literature in the area of support vector machine, upon which this work builds. Support vector machines are a popular data mining technique which have obtained high performance in many applications, such as credit scoring, financial time series prediction and spam categorization and so on (Martens, Baesens, Van Gestel, & Vanthienen, 2007).

Hsieh (2005) proposed a hybrid mining approach for credit scoring model. In his experiments, he specially lists the distribution of the relative importance for each input variable when using neural network.

2.1. Link analysis relate works

In the case of Web search there are two most influential hyperlink search algorithms, that is, PageRank (Brin & Page, 1998) and HITS (Kleinberg, 1998), which are related to social network. They exploit the hyperlinks of Web to rank pages according to their levels of authority. Ito Takahiko, Shimbo Masashi, Kudo Taku, and Matsumoto Yuji (2004, 2005) explored the application of kernels methods to link analysis. Borodin, Rosenthal, Roberts, and Tsaparas (2005) introduced a theoretical framework for the study of link analysis ranking algorithms. They worked with the hubs and authorities framework defined by Kleinberg (1998).

With the development of link analysis ranking, it has led to a surge of research activity in the area of information. There are a lot of workshops in many mainstream conferences, such as LinkKDD (2004, 2005, 2006), SDM (2004), SIAM (2005) and so on.

Apart from search ranking, hyperlinks are also useful for finding Web communities (Flake, Lawrence, & Giles, 2000). Beyond explicit hyperlinks on the Web, links in other contexts are useful too, for example, for ranking order individuals in a given social network in terms of a measure of their importance. Social network has become an important part in link analysis, which is the study of social entities and their interactions and relationships (Thelwall, 2006). The interactions and relationships can be described as a network or graph. From the network, we can study the properties of its structure, and the role, position and prestige of each social actor. We can also find various kinds of sub-graphs, e.g., communities formed by groups of actors. And we are more interested in social communities so as to find creditable communities from applicants, and banks can gain profit from such communities.

The technique used in link rank analysis can be extent with some latent topic models beside web graph. We would like to mine the relationship based on attributes and link structure. We are trying to construct credit communities' graphs and find important information which affects the results.

There is a fundamental assumption of link analysis that an edge between two nodes in citation graphs signifies that two nodes are in some sense related (Ito Takahiko et al., 2004). It is famous that birds of a feather flock together and things of one kind come together. Therefore, we will divide all applicants into two basis class, that is, good or bad, by their characters. Then we give another fundamental assumption in social network that two persons have the similar information of application such as income, marital condition, and then they may be possible to have similar credit. In another word, more similar information two persons have, more similar they have credit on their behavior.

2.2. SVM relate works with credit scoring

Support vector machines (Hsu, Chang, & Lin, 2003) are state-of-art data mining techniques which are popular to data classification. Recently Gold and Sollich (2005), Gold, Holub, and Sollich (2005) proposed a Bayesian method for tuning the hyperparameters of a support vector machine (SVM) classifier. They used the Nystrom approximation to the SVM kernel and their method significantly reduces the dimensionality of the space to be simulated in the hybrid Monte Carlo simulation. And then least squares support vector machine (LS-SVM) classifiers (Van Gestel et al., 2006) were applied within the Bayesian evidence framework in order to automatically infer and analyze the creditworthiness of potential corporate clients.

In credit industry, SVM has been claimed to be effective and accurate tool for credit analysis (Huang, Chen, & Wang, 2007). And various methods have been extensively employed in the credit scoring business. Piramuthu (2006)

present some machine learning methods for credit risk evaluation. Specifically, they considered the effects of preprocessing of credit risk evaluation data used as input for machine learning methods. Schebesch and Stecking (2005) extent standard SVM methods to non-standard cases with weighted classes and moderated outputs. They dramatically improved cost related performance of out of sample classification.

Yang (2007) proposed a novel and practical adaptive scoring system based on an incremental kernel method. And the scoring model is adjusted according to an on-line update procedure that can always converge to the optimal solution without information loss or running into numerical difficulties.

However, the SVM classifier is a complex mathematical method and rather to incomprehensible for humans. In many real-life applications both accuracy and comprehensibility are desired, for example, credit risk evaluation. Martens et al. (2007) proposed rule extraction techniques for SVM and introduced two techniques, Trepan and G-REX, taken from the artificial neural networks domain.

Huang et al. (2007) proposed three strategies to apply SVM so as to improve the performance of credit scoring model based on general SVM model. They pointed that there are two major aspects that we should consider cautiously when we use SVM method to solve credit scoring problem. One is how to select the optimal input feature and the other is how to set good kernel parameters.

3. Find importance matrix about information

We consider the problem of credit scoring from a simple graph. Given an applicant' credit information dataset DB which contains n applicants. And every applicant has m related information. Such dataset is denoted by a bipartite graph $G = \{F, P, E\}$, where applicants process of their information. The set of vertices has two subsets, one set P denotes all applicants and the other set F denotes all information. The set E of edges denotes correlative relation between applicants and information. We stipulate that an edge e_{ij} from p_i to f_j means that applicant p_i is how to relate to information f_j. $||$ denotes the cardinality of a set, and $|F| = m$, $|P| = n$. Hence all applicants can be denoted as $p = \{p_1, p_2, \ldots, p_n\}$ and information vector can be denoted as $f = \{f_1, f_2, \ldots, f_m\}$.

For an applicant $(\mathbf{x}_i, y_i)$, $(i = 1, 2, \ldots, m)$, $\mathbf{x}_i$ can be described as $\{e_{i1}, e_{i2}, \ldots, e_{im}\}$, which shows the value of information of the application. So all applicants can be described as a $n \times m$ matrix $A = [a_{ij}]$. Here a_{ij} denotes the value of information f_j of applicant p_i. And binary class labels $y_i \in \{+1, -1\}$. Therefore let A denote the information matrix of applicants without classification.

Here we consider link analysis ranking method in credit scoring problem. Ranking is an integral component of any information retrieval system (Borodin et al., 2005).

3.1. Main frame

Main steps are as follows based on link analysis ranking framework.

Input data: an applicant dataset DB, which can be denoted to matrix $A = [a_{ij}]$.

Output data: information weight.

(1) Find weighted co-information matrix by applicants' matrix (Section 3.2).
(2) Use link analysis to sort information (Section 3.3).
(3) Get new input feature vector (Section 3.4).
(4) Run SVM classifier (Section 4).

3.2. Find normalized weighted matrix by applicants' matrix

First we get a basis information matrix from the source matrix through data preprocessing. For continuous variable, list the information matrix of all applicants and regularize/normalize every row as follows:

$$a_{ij} = \frac{a_{ij}}{\max(a_{kj})} \tag{1}$$

So we can form a new matrix called 'k-ranking row average weighted coefficient'. From now on, let A denote the matrix after data preprocess by the formulation (1).

Therefore, A is the adjacency matrix of an information graph G. The symmetric matrix $A^{\mathrm{T}}A$ is called co-information matrix of G. The co-information graph of G is the weighted undirected graph induced by taking $A^{\mathrm{T}}A$ as its adjacency matrix. Similarly, the symmetric matrix AA^{T} is called the co-applicant matrix of G. The co-applicant is the weighted undirected graph induced by taking AA^{T} as its adjacency matrix.

Co-information coupling and co-applicant coupling are new methods of computing relatedness (or similarity) between applicants from an information graph. Co-information coupling reflected the relatedness between information as the number of the other information similar to them both.

Let $B = A^{\mathrm{T}} \times A$. So $B = [b_{ij}]$. So the matrix B is non-negative integer matrix. It is easy to see that a line $b_{i1}, b_{i2}, \ldots, b_{im}$ in the matrix B describes the relationship between information f_i and the other information. That's means, each element of B represents the value of co-information coupling.

$$\begin{aligned} b_{ij} &= |a_{1i}, a_{2i}, \ldots, a_{ni}| \times |a_{1j}, a_{2j}, \ldots, a_{nj}|' \\ &= \sum_{k=1}^{n} a_{ki} a_{kj} \quad (k = 1, 2, \ldots, n) \end{aligned} \tag{2}$$

In link analysis there are a fundamental assumption that an edge between nodes in graphs signifies that two nodes are in some sense related (Ito et al., 2004, 2005). When analyzing credit data, we need differentiate discrete variable and continuous variable because of the information available. We think that applicant a_i is similar to applicant a_j

on information p_k either if a_{ik} is close to a_{jk} for continuous variables or if a_{ik} is equal to a_{jk} for discrete variables. Then we get 1/0 element in B matrix after we deal with every element.

3.2.1. For discrete variable

If an information variable is discrete, then the corresponding applicants are similar if the values are the same.

$$a_{ij}a_{jk} = \begin{cases} 1 & \text{if } a_{ik} = a_{jk} \\ 0 & \text{if } a_{ik} \neq a_{jk} \end{cases} \tag{3}$$

3.2.2. For continuous variable

If an information variable is continuous, then the corresponding applicants are similar if the values are close.

$$a_{ij}a_{jk} = \begin{cases} 1 & \text{if } |a_{ij} - a_{jk}| \leqslant \text{threshold} \\ 0 & \text{if } |a_{ij} - a_{jk}| > \text{threshold} \end{cases} \tag{4}$$

So the next problem is how to decide the threshold. We want to use mean square error of a row w because the row w shows the vector consist of the information w of all applicants. Let a row w denoted as $a_w = |a_{1w}, a_{2w}, \ldots, a_{nw}|'$. So it is easy to get the following formulation:

$$\text{avg} = \frac{1}{n}\sum_{k=1}^{n} a_{kw} \tag{5}$$

$$\text{coefficient}_w = \frac{1}{n}(|a_{1w} - \text{avg}| + \ldots + |a_{nw} - \text{avg}|) \tag{6}$$

$$= \frac{1}{n}\left(\sum_{u=1}^{n} |a_{uw} - \text{avg}|\right) \tag{7}$$

Mean square error shows the average difference of the information in the row w. So let the threshold be equal to mean square error. That is means, threshold = coefficient.

3.3. Find weight in the matrix

In this section we work within the hubs and authorities framework defined by (Kleinberg, 1998).

3.3.1. HARA algorithm

In this framework, every applicant can be thought of as having one identity, differed from original HITS algorithm (Kleinberg, 1998). We could think that the applicant is corresponding to the hub and information is corresponding to the authority in the web. In web a graph G as a bipartite graph where hubs point to authorities. Similarly, in credit scoring, applicants possess of information. The hub identify captures the quality of the applicant as a pointer (possessor) to useful information, and the authority identify captures the quality of the applicant as an information itself. So the new proposed algorithm is called hub authority ranking applicants (HARA for short) algorithm.

A good applicant has good information just as good hub points to good authority, and good information is information that belongs to good applicants just like good authority is an authority pointed by good hubs.

So for information matrix, applicants weights can be described as an n-dimension vector h, where h_i is the hub weight of applicant p_i. Similarly, information weights can be described as n-dimension vector a, a_i value shows the authority weight of information.

So we have two formulas as follows:

$$a_i = \sum_{j \to i} h_j \tag{8}$$

$$h_j = \sum_{i \to j} a_i \tag{9}$$

The adjacent matrix $\boldsymbol{A}$ of the base set is an $n \times m$ matrix with entries of either 0 or 1. The matrix entry $\boldsymbol{A}[i,j]$ is set to 1 if page i has a hyperlink to page j; otherwise, it set to 0. The hub weights and authority weights are represented by p and f. The above update rule can be written as follows:

$$a = A^{\mathrm{T}} \cdot h \tag{10}$$

$$h = A \cdot a \tag{11}$$

The algorithm is summarized as follows:

HACS algorithm.

(1) Initialize all weights to 1.
(2) Repeat until the weights converge:
   (a) For every hub $i \in H$

$$h_j = \sum_{j \to i} a_i$$

   (b) For every authority $i \in A$

$$a_i = \sum_{j \to i} h_j$$

   (c) Normalize

The sequence of hub weights and authority weights converges to the principle eigenvector of AA^{T} and $A^{\mathrm{T}}A$, respectively. Furthermore, the converged weights are independent of the choice of initial weights.

Finally, let the authority weight $\text{auth}_1 = |a_1, a_2, \ldots, a_m|$.

3.3.2. HubAvgRA algorithm

The intuition of the HubAvgRA algorithm is that a good information (hub) should belong (point) only (or at least mainly) to good applicants (authorities), rather than to both good and bad applicants (authorities). The idea comes from the HubAvg algorithm (Borodin et al., 2005). So we try to collect good information to denote the good applicants. The algorithm is called HubAvb ranking applicants algorithm (HubAvgRA for short).

HubAvgRA algorithm

(1) Initialize all weights to 1
(2) Repeat until the weights converge:

(a) For every hub $i \in H$

$$h_j = \frac{1}{\text{Indegree}(i)} \sum_{i \to j} a_i$$

(b) For every authority $i \in A$

$$a_i = \sum_{j \to i} h_j$$

(c) Normalize

Similarly, let the current authority weight auth_2.

3.3.3. ATkRA algorithm

Authority-Threshold algorithm (Borodin et al., 2005), sets the hub weight of node i to be the sum of the k largest authority weights. That's means; a node is a good hub if it points to at least k good authorities. The value of k is passed as a parameter to the algorithm. So we also want to use the idea to deal with applicants' matrix. …. The algorithm is called AT k ranking applicants algorithm (ATkRA for short).

ATkRA algorithm

(1) Initialize all weights to 1
(2) Repeat until the weights converge:
(a) For every hub $i \in H$

$$h_j = \frac{1}{\text{Indegree}_k(i)} \sum_{i \to j} a_i$$

(b) For every authority $i \in A$

$$a_i = \sum_{j \to i} h_j$$

(c) Normalize

Similarly, let the current authority weight auth_3.

3.4. Get new input feature from information matrix

Consequently, the relative importance of information could be achieved by the authority ranking list. Now the authority ranking list is the weight value of input feature.

Let $C = [c_{ij}]$. Then c_{ij} is defined as follows:

$$c_{ij} = (\text{auth}_k)_{ij} \times e_j \quad (k = 1, 2, 3) \tag{12}$$

So the matrix C is a non-negative integer matrix. It is easy to see that a line $c_{i1}, c_{i2}, \ldots, c_{in}$ in the matrix C describes the relationship between an applicant p_i and the other applicants, where c_{ij} denotes the result of information k of applicant p_i multiply by information k of applicant p_j, $k = 1, 2, \ldots, m$.

4. Basic concept of SVM classifier

In this section we give basic concept of SVM classifier. Support vector machines (Vapnik, 1998) were developed based on the structural risk minimization principle from statistical learning theory.

Then we use general SVM model for classifier with input feature A.

4.1. Support vector classification

The main idea of support vector machine is to construct a hyperplane as the decision surface such that the margin of separation between positive and negative examples is maximized.

For a training set of examples, with input data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)$, $\mathbf{x} \in R^n$, and corresponding binary class labels $y_i \in \{+1, -1\}$. There are a hyperplane as Eq. (13) when those can be separated line.

$$\langle \omega, \mathbf{x} \rangle + b = 0 \tag{13}$$

SVM finds optimal separating hyperplane with the maximum margin by solving the following optimization problem:

$$\text{Min} \qquad \frac{1}{2}\omega^{\mathrm{T}}\omega \tag{14}$$
$$\text{subject to :} \quad y_i(\langle \omega \cdot x_i \rangle + b) - 1 \geqslant 0$$

Training SVM for pattern recognition, leads to the quadratic optimization problem as follows:

$$\text{Max} \qquad W(\alpha) = \sum_{k=1}^{l} \alpha_k - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i \, y_j \langle \mathbf{x}_i, y_j \rangle \tag{15}$$
$$\text{subject to :} \quad 0 \leqslant \alpha_i \leqslant C, \quad (i = 1, \ldots, m)$$
$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

where l is the number of training examples and α is a vector of l variables.

Followed by the steps described in the linear generalized case, we obtain decision function of the following form:

$$f(x) = \text{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i^* \langle \Phi(\mathbf{x}), \Phi(x_i) \rangle + b^* \right) \tag{16}$$
$$= \text{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i^* \langle k(\mathbf{x}, x_i) \rangle + b^* \right) \tag{17}$$
$$\text{subject to :} \quad 0 \leqslant \alpha_i \leqslant C, \quad (i = 1, \ldots, m)$$
$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

where $k(\mathbf{x}, x_i)$ is the kernel function performing the non-linear mapping into feature space, and the constraints are unchanged.

Here C is penalty parameter to tradeoff radius and slack variable ξ.

Radial basis function (RBF) is a common kernel function as follows:

$$k(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2) \tag{18}$$

5. Experimental results

In this section we present our results of the experiments on a credit dataset from real world. We compare the efficiency of the proposed algorithms with the previous best method SVM (Huang et al., 2007), because the algorithm based on SVM is the state of art in credit scoring model literature.

First we give our experimental data sets and then our results in datasets.

All experiments are conducted on a 2.6 GHZ Intel PC with 512MB main memory, using the Microsoft Windows XP. We use SVM^{light} (Joachims, 1999) to test the performance of SVM.

5.1. Real world credit data set

The real world data set is the Australian credit data set, which is available from the UCI repository of machine learning database (Murphy & Aha, 2001).

The Australian credit dataset, has 307 instances of creditworthy applications and 383 instances of credit worthless. Each instance has 15 attributes $\{a_1, a_2, \ldots, a_{15}\}$ in which the last one a_{15} described class (accepted or rejected). All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. This dataset is interesting because there is a good mix of attributes – continuous, nominal with small numbers of values, and nominal with larger numbers of values.

We use the method to the dataset by randomly partitioning into training and independent test sets using 10-fold cross-validation. The test set is used to guarantee that our results are valid and can be generalized to make predictions to new data. Each of the ten random partitions performs as an independent holdout test set for the credit scoring model with the rest of four partitions. The benefits of cross-validation are that the impact of data dependency is minimized and reliability of results is improved.

5.2. Experimental results

For SVM's parameter, we use gird search method to find good parameters. Here the performance of the proposed classifiers is benchmarked against general SVM for credit scoring applications. Our SVM classification model uses radial basis function (RBF) kernels. There are two parameters associated with the RBF kernels: C and γ. The parameter controls the width of the area of strong activation of a radial basis function around its center vector input space.

Table 1 summarizes the results of the grid-search using 10-fold cross-validation. The best classification accuracy of every algorithm has been overline on the results.

From Table 1, it is easy to see that HARA, HubAvgRA and ATkRA are competitive with general SVM in this case. Our proposed algorithms have a better performance in credit scoring. This can be explained by using our link analysis structure. The overall classification accuracy of the holdout data turned out to be 77.10% by general SVM method. With average accuracy value 89.28%, we get a significant improvement. As shown in Table 1, the classification accuracy of each class is acceptable and satisfactory level compared to general SVM method.

In addition, SVM is significantly inferior to HubAvgRA and ATkRA algorithms. HubAvgRA and ATkRA algorithms approximately shows very close results with HARA algorithm.

5.3. Two class error analysis

There are two class errors which need analysis. If we misclassify a good application to bad, we call the error "Error 1". On the other side, if we misclassify a bad application to good, then the error is called "Error 2". We should distinguish the two class errors because the cost of every error caused is not similar. Error 2 often has more cost than Error 1, so we should descend the happen of Error 2.

This dataset requires use of a cost matrix in Table 2 (Murphy & Aha, 2001). The rows in Table 2 represent the actual classification and the columns the predicted classification. It is worse to class a customer as good when they are bad (5), than it is to class a customer as bad when they are good (1).

When building highly accurate scoring models, misclassification patterns emerge frequently (Kim & Sohn, 2004). The results of cost value according to those algorithms are summarized in Table 3. The experimental results indicate that our proposed algorithms provide better error

Table 2
Cost matrix

	Class 1 (good)	Class 2 (bad)
Class 1 (good)	0	1
Class 2 (bad)	5	0

Table 1
Results of Australian credit dataset (%)

Model	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Overall
SVM	79.71	75.36	82.61	78.26	78.26	78.26	73.91	76.81	75.36	72.46	77.10
HARA	88.41	97.10	86.96	88.41	85.51	88.41	89.86	84.06	91.30	92.75	89.28
HubAvgRA	88.41	91.30	86.96	86.96	84.06	88.41	88.41	86.96	89.86	92.75	88.41
ATkRA	85.51	91.30	86.96	86.96	85.51	84.06	84.06	82.61	86.96	91.30	86.52

Table 3
Two type of error analysis

No.	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Overall
SVM	54	41	36	55	47	65	62	60	41	75	53.6
HARA	28	6	13	12	14	20	27	31	18	17	18.6
HubAvgRA	24	18	21	17	23	20	28	21	19	13	20.4
ATkRA	22	14	13	13	14	23	31	20	21	14	18.5

ability to credit scoring than general SVM algorithm. In particular, ATkRA gets the best results when considering the two type of error.

6. Conclusions and future research

This study presents a link analysis-based work to credit industry that demonstrates the advantages of weighted SVM to credit analysis and showed its attractive classification power compared to the existing methods. Using some minor modification of the standard SVM dramatically improves cost related performance of out of sample classification. In particular, this study utilizes a grid-search technique using 10-fold cross-validation in order to choose optimal values on C and γ. The results of empirical analysis show that our proposed algorithms outperform the general SVM method.

Further research should include collecting more information of applications so as to enhance precision of credit scoring.

And we should consider further works after approving applicants, for example, applicants maintenance, market analysis, applicants profit analysis model and so on.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant Nos. 60433020, 60673099, the Key Laboratory for Symbol Computation and Knowledge Engineering of the National Education Ministry of China, and 985 Project: Technological Creation Support of Computation and Software Science.

References

Blochlinger, A., & Leippold, M. (2006). Economic benefit of powerful credit scoring. *Journal of Banking & Finance, 30*(3), 851–873.

Borodin, A., Rosenthal, J. S., Roberts, G. O., & Tsaparas, P. (2005). Link analysis ranking: Algorithms, theory and experiments. *ACM Transactions on Internet Technologies (TOIT), 5*(1), 231–297.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international world wide web conference*. Brisbane, Australia.

Gold, C., Holub, A., & Sollich, P. (2005). Bayesian approach to feature selection and parameter tuning for support vector machine classifiers. *Neural Networks, 18*(5–6), 693–701.

Gold, C., Sollich, P. (2005). Fast Bayesian support vector machine parameter tuning with the Nystrom method. In *Proceeding of the international joint conference on neural networks* (pp. 2820–2825), Montreal, Canada.

Ester, M., Ge, R., Jin, W., & Hu, Z. (2004). A micro-economic data mining problem: Customer-oriented catalog segmentation. In *Proceedings of 10th ACM SIGKDD international conference on knowledge discovery and data mining (KDD'2004)* (pp. 557–562). Seattle, WA, USA.

Flake, G.W., Lawrence, S., Giles, C.L. (2000). Efficient identification of web communities. *KDD 2000* (pp. 150–160). Boston, MA, USA.

Getoor, L., & Diehl, C. P. (2005). Link mining: A survey. *SIGKDD Explorations, 7*(2), 3–12.

Hsieh, N. C. (2005). Hybrid mining approach in the design of credit scoring models. *Expert Systems with Applications, 28*(4), 655–665.

Hsu, C.W., Chang, C.C., Lin, C.J. (2003). A practical guide to SVM classification. Available from http://www.csie.ntu.edu.tw/cjlin/papers/guide/guide.pdf.

Huang, C. L., Chen, M. C., & Wang, C. J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications, 33*(4), 847–856.

Ito Takahiko, Shimbo Masashi, Kudo Taku, & Matsumoto Yuji (2004). Application of kernels to link analysis: First results. In *Proceedings of the second workshop on mining graphs, trees and sequences (MGTS'04)* (pp. 13–24), Pisa.

Ito Takahiko, Shimbo Masashi, Kudo Taku, & Matsumoto Yuji (2005). Application of kernels to link analysis. In *Proceedings of the eleventh ACM SIGKDD international conference on knowledge discovery and data mining (KDD-2005)* (pp. 586–592).

Joachims, T. (1999). Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, & A. Smola (Eds.), *Advances in kernel methods – support vector learning*. MIT-Press.

Kim, Y. S., & Sohn, S. Y. (2004). Managing loan customers using misclassification patterns of credit scoring model. *Expert Systems with Applications, 26*(4), 567–573.

Kleinberg, J. M. (1998). Authoritative source in a hyperlinked environment. In *Proceedings of the 9th ACM-SIAM symposium on discrete algorithms, ACM* (pp. 668–677).

Kleinberg, J., Papadimitriou, C., & Raghavan, P. (1998). A microeconomic view of data mining. *Journal of Data Mining and Knowledge Discovery, 2*(4), 311–324.

Martens, D., Baesens, B., Van Gestel, T., & Vanthienen, J. (2007). Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research, 183*(3), 1466–1476.

Chen, M. C., & Huang, S. H. (2003). Credit scoring and rejected instances reassigning through evolutionary computation techniques. *Expert Systems with Applications, 24*(4), 433–441.

Murphy, P. M., & Aha, D. W. (2001). *UCI repository of machine learning databases*. Department of Information and Computer Science, University of California, Irvine, CA, http://www.ics.uci.edu/mlearn/LRepository.html.

Piramuthu, Selwyn (2006). On preprocessing data for financial credit risk evaluation. *Expert Systems with Applications, 30*(3), 489–497.

Schebesch, Klaus, B., & Stecking Ralf (2005). Support vector machines for credit scoring: Extension to non standard cases. In D. Baier, & K.-D. Wernecke (Eds.), *Innovations in classification, data science and information systems (pp. 498–505)*.

Thelwall, M. (2006). Interpreting social science link analysis research: A theoretical framework. *Journal of the American Society for Information Science and Technology, 57*(1), 60–68.

Thomas, Lyn C., Edelman, David B., & Crook, Jonathan N. (2002). Credit scoring and its applications.

Van Gestel, T., Baesens, B., Suykens Johan, A. K., Van den Poel, D., Baestaens, D. E., & Willekens, M. (2006). Bayesian kernel based classification for financial distress detection. *European Journal of Operational Research, 172*(3), 979–1003.

Vapnik, V. (1998). *Statistical learning theory*. Chichester, UK: Wiler.

Workshop on Link Analysis and Group Detection (LinkKDD2004). http://www.cs.cmu.edu/dunja/LinkKDD2004/.

Workshop on Link Discovery: Issues, Approaches and Applications (LinkKDD-2005) (2005). http://www.isi.edu/LinkKDD-05/.

Workshop on Link Analysis: Dynamics and Static of Large Networks (LinkKDD2006) (2006). http://kt.ijs.si/Dunja/LinkKDD2006/.

Workshop on Link Analysis. SDM (2004). http://www-users.cs.umn.edu/aleks/sdm04w/.

Workshop on Link Analysis, Counterterrorism and Security. SIAM (2005). http://www.cs.queensu.ca/skill/siamworkshop.html.

Yang, Y. X. (2007). Adaptive credit scoring with kernel learning methods. *European Journal of Operational Research, 183*(3), 1521–1536.