https://doi.org/10.1016/j.asoc.2018.07.005

Stationary Mahalanobis Kernel SVM for Credit Risk Evaluation

Hao Jiang

Department of Mathematics, School of Information, Renmin University of China, No. 59 Zhong Guan Cun Street, HaiDian District, Beijing, China

Wai-Ki Ching

Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong

Ka Fai Cedric Yiu

Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Yushan Qiu[☆]

College of Mathematics and Statistics, Shenzhen University, Nanhai Avenue 3688, 518060 Shenzhen, China

Abstract

This paper proposed Mahalanobis distance induced kernels in Support Vector Machines (SVMs) with applications in credit risk evaluation. We take a particular interest in stationary ones. Compared to traditional stationary kernels, Mahalanobis kernels take into account on feature's correlation and can provide a more suitable description on the behavior of the data sets. Results on real world credit data sets show that stationary kernels with Mahalanobis distance outperform the stationary kernels with various distance measures and they can also compete with frequently used kernels in SVM. The superior performance of our proposed kernels over other classical machine learning methods and the successful application of the kernels in large scale credit risk evaluation problems may imply that we have proposed a new class of kernels appropriate for credit risk evaluations.

Keywords: Mahalanobis Distance; Support Vector Machine (SVM); Indefinite; Stationary Kernel; Credit Risk.

[☆]To whom correspondence should be addressed

Email addresses: jiangh@ruc.edu.cn (Hao Jiang), wching@hku.hk (Wai-Ki Ching), cedric.yiu@polyu.edu.hk (Ka Fai Cedric Yiu), yushan.qiu@szu.edu.cn (Yushan Qiu[☆])

1. Introduction

Credit risk assessment is extremely important for financial and banking industries. In the past decades, a number of credit risk evaluation methods have been proposed such as discriminant analysis, decision tree, K-nearest neighborhood and linear programming, logistic regression, etc. These methods based on traditional statistics have good interpretability. Artificial Intelligence (AI) techniques such as Artificial Neural Networks (ANN), Genetic Algorithm (GA) and Support Vector Machines (SVMs) have also been employed in risk assessment problems. Though lacking clear interpretability, AI techniques are empirically shown to be advantageous to traditional statistical models for credit risk evaluation and therefore have been widely adopted.

SVM, a supervised machine learning technique, was originally introduced by Vapnik [1]. In machine learning, SVMs [2] are traditionally regarded as one of the best algorithms in terms of minimizing the structural misclassification probability. SVM works by embedding the data into a high dimensional feature space and the kernel in SVM plays a major role in model formulation. Kernel trick in SVM assures that we do not need to calculate the embedding function explicitly as long as we can construct a proper kernel matrix. Authors in [3] have proposed a new form of regularization that is able to utilize the label information of a data set for learning kernels. Pan et al., [4] proposed a framework which integrates multiple sources of information and enables us to develop flexible and effective kernel matrices. Authors in [5] suggested a novel supervised and nonlinear approach to enhance the classification power of nonnegative matrix factorization. The Positive Semi-Definite (PSD) property [6] of a kernel matrix is required to ensure the existence of a Reproducing Kernel Hilbert Space (RKHS) where a convex optimization formulation can be deduced to yield an optimal solution. Informally speaking, a kernel consists of embedding general points into an inner product space. The inner product colloquially measures the extent of overlaps or similarity between two different data vectors in their feature space. Therefore, models for credit risk evaluation based on SVM to some extent relies heavily on the kernel functions in describing the relationship of the original data. A number of credit risk evaluation models have been proposed based on SVMs [7, 8, 9, 10]. In [7], a least squares SVM model was proposed to allocate and charge bank capital. It was compared with ordinary least squares (OLS) regression, ordinal logistic regression (OLR)and multilayer perceptrons (MLPs). Results show that LS-SVMs are significantly better when contrasted with the classical techniques. Huang et al. [8] integrated SVM with neural networks for dealing with credit rating problems. The model was compared with linear regression model and logistic regression model. Support vector machines showed comparable results achieved to that of back propagation neural networks and are better than logistic regression and linear regression models. Besides, the results from neural network model are utilized for variable interpretation and helped to determine the relative importance of the input variables. SVM-based meta model was later proposed in [9] for business risk identification. They pointed out the drawbacks of neural networks trapping in local minima

and SVM (RBF kernels) stuck in overfitting, and proposed using original data sets to generate different training sets, in order to train a number of SVM models for final metamodel integration. In [10], authors proposed a least squares fuzzy SVM approach to credit risk evaluation, where fuzzy membership was introduced for modeling sample labels. Hybrid or ensemble models were also widely applied in credit risk evaluation problems [11, 12, 13, 14, 15].

Most of previous research works focus on the effect on a proper model framework in improving the prediction accuracy but neglect the distribution relationship between data points. Mahalanobis distance is a new measure in modeling the relationship between two data points in the data set. It was previously used in one-class SVM [16, 17], it was also incorporated into fuzzy c-means clustering for fuzzy SVM description in multi-class classification [18]. The application of Mahalanobis distance for financial forecast can also be found in [19, 20] where Mahalanobis distance was mainly used to correct the weakness of Euclidean distance in calculating feature correlations. In [19], a support vector regression framework was proposed for financial forecast. [20] proposed a strategy using Mahalanobis distance, Gram-Schmidt orthogonalization to do financial crisis predictions. However, the integration of Mahalanobis distance with SVM kernels in credit risk evaluation has been less extensively addressed.

Our focus in this problem is stationary kernel construction where we assume the distance between two simultaneously translated vectors will be the same as the one without translation. This property is important as it can keep the relative geometric relationship of data points. In this paper, by taking into consideration on the distribution relationship between data points, we propose Mahalanobis distance based kernels in conjunction with SVMs for credit risk evaluation, where we formulate a number of new kernels: MRBF, MPower and MLog kernels. We show that these kernels are all suitable for SVM framework. MRBF kernel is a valid mercer kernel, MPower kernel and MLog Kernel like Power kernel, Log kernel that are conditionally positive definite for degree no greater than 2. The two kernels have been successfully applied in image recognition but have not been applied in credit risk assessment before. We show that our constructed kernels are all stationary kernels and they outperform the stationary kernels with a number of distance measures. Besides, they are more competitive to frequently used kernels in SVMs. These kernels can provide a more suitable description on the behavior of the data sets and shown to be suitable in credit risk evaluation problems. The remainder of the paper is structured as follows. In Section 2 we introduce some background and then propose Mahalanobis distance based stationary kernels, with a focus on some indefinite kernels. Section 4 describes the materials and experimental results with comparative analysis. We finally give conclusions and future work in the last section.

2. Methods

2.1. Preliminaries

Support Vector Machines (SVMs) are very popular machine learning techniques for both classification and regression analysis, which have been applied in numerous fields such as text categorization, pattern recognition etc. SVM constructs a hyperplane to maximize the margin between different classes and the optimization problem can be expressed as follows:

$$\begin{cases} \text{subject to} & \text{Maximize } \frac{2}{\|\mathbf{w}\|} \\ & y_i(\mathbf{w}\cdot\mathbf{x}_i - b) \geq 1 \text{ for any } i \in \{1,2,\ldots,n\} \end{cases} \tag{1}$$

where $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}, \mathbf{x}_i \in \mathbb{R}^n, y_i \in \{-1, 1\}$ stand for data set of n data instances with corresponding class annotations.

Applying duality theory, we can obtain a hyperplane by considering optimization problem:

$$\begin{cases} \text{subject to} & \text{Maximize} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\alpha^T \mathbf{H}\alpha \\ & \alpha_i \geq 0 \text{ for any } i \in \{1,2,\ldots,n\} \\ & \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{2}$$

where $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]$ and

$$\mathbf{H} = \begin{pmatrix} y_1^2\mathbf{x}_1^T\mathbf{x}_1 & y_1y_2\mathbf{x}_1^T\mathbf{x}_2 & \ldots & y_1y_n\mathbf{x}_1^T\mathbf{x}_n \\ y_2y_1\mathbf{x}_2^T\mathbf{x}_1 & y_2y_3\mathbf{x}_2^T\mathbf{x}_3 & \ldots & y_2y_n\mathbf{x}_2^T\mathbf{x}_n \\ \vdots & \vdots & \ddots & \vdots \\ y_ny_1\mathbf{x}_n^T\mathbf{x}_1 & \ldots & \ldots & y_n^2\mathbf{x}_n^T\mathbf{x}_n \end{pmatrix}.$$

The convex quadratic problem can ensure a solution for α. If we assume that $S = \{i|\alpha_i > 0\}$ is the set of support vectors, we can determine new data point through decision function: $y = \text{sign}(\mathbf{w^T x} - b)$ where

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, b = \frac{1}{|S|}\sum_{i\in S}(\mathbf{w^T x}_i - y_i).$$

Soft margin [21] can be developed dealing with not fully linear separable data sets, allowing for mislabeled instances.

In the case of nonlinearly separable data sets, we can nonlinearly map input vectors into higher dimensional feature space [22]. The kernel matrix is then constructed through pairwise comparisons. For example,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j)$$

where $\mathbf{x}_i \rightarrow \phi(\mathbf{x}_i)$ is a mapping from original input space into transformed feature space.

A kernel is said to be stationary [23] if $\forall \mathbf{c} \in R^p$, we have

$$K(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_i + \mathbf{c}, \mathbf{x}_j + \mathbf{c}).$$

Stationary kernels have been widely applied in computer science. In the following, we will propose a number of Mahalanobis distance induced stationary kernels.

2.2. Mahalanobis RBF Kernel

Suppose we are given a number of data instances

$$\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}, \mathbf{x}_i \in \mathbb{R}^p, y_i \in \{-1, 1\}$$

we propose the construction of Mahalanobis RBF Kernel in the following way:

$$K_{\text{maha}}(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{e}^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||_m^2}{\lambda}}$$

where λ is a parameter and

$$||\mathbf{x}_i - \mathbf{x}_j||_m^2 = (\mathbf{x}_i - \mathbf{x}_j)^T \cdot S^{-1} \cdot (\mathbf{x}_i - \mathbf{x}_j)$$

refers to the square of Mahalanobis distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, where S is the covariance matrix of the data set.

In the following, we will give theoretical verification on the validity of the kernel. This is equivalent to find the feature map: $\phi : \mathbf{R}^p \rightarrow H$, such that

$$K_{\text{maha}}(\mathbf{x}_i, \mathbf{x}_j) = < \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) >_H$$

where H is some Hilbert space.

Let

$$\Omega = \left\{ f(x) : R^p \rightarrow C | \int_{R^p} |f(x)|^2 \mathrm{e}^{-\frac{||x||^2}{2}} dx < \infty \right\}$$

be the space of square integrable complex-valued functions on R^p. Then we have

$$< f(\mathbf{x}), g(\mathbf{x}) >= \int_{R^p} \overline{f(\mathbf{x})} g(\mathbf{x}) \mathrm{e}^{-\frac{||\mathbf{x}||^2}{2}} dx.$$

Denote

$$\phi(\mathbf{x})(\mathbf{t}) = \mathrm{e}^{\frac{i(S^{-\frac{1}{2}}\mathbf{x}, \mathbf{t})}{\sqrt{\lambda/2}}}$$

where S is the covariance matrix between $\mathbf{x}$ and $\mathbf{y}$. Then we have

$$\begin{aligned} < \phi(\mathbf{x}), \phi(\mathbf{y}) >_\Omega &= \frac{1}{\sqrt{2\pi}} \int_{R^p} \mathrm{e}^{\frac{i(S^{-\frac{1}{2}}(\mathbf{y}-\mathbf{x}), \mathbf{t})}{\sqrt{\lambda/2}}} \mathrm{e}^{-\frac{||\mathbf{t}||^2}{2}} d\mathbf{t} \\ &= \mathrm{e}^{-\frac{||S^{-\frac{1}{2}}(\mathbf{y}-\mathbf{x})||^2}{\lambda}} \frac{1}{\sqrt{2\pi}} \int_{R^p} \mathrm{e}^{-||t - i\frac{S^{-\frac{1}{2}}(\mathbf{y}-\mathbf{x}), \mathbf{t})}{2\sqrt{\lambda/2}}||^2} d\mathbf{t} \\ &= \mathrm{e}^{\frac{-||\mathbf{y}-\mathbf{x}||_m^2}{\lambda}} \end{aligned}$$

In terms of infinite mapping, we can just assume

$$\hat{\mathbf{x}} = \mathbf{S}^{-\frac{1}{2}}\mathbf{x} \quad \text{and} \quad \hat{\mathbf{y}} = \mathbf{S}^{-\frac{1}{2}}\mathbf{y}$$

and the mapping function is then given by

$$\phi(\mathbf{x}) = e^{-\frac{1}{\lambda}\hat{\mathbf{x}}^2}\left(1, \sqrt{\frac{2\lambda}{1!}}\hat{\mathbf{x}}, \sqrt{\frac{(2\lambda)^3}{3!}}\hat{\mathbf{x}}^{\mathbf{3}}, \ldots, \right).$$

Therefore, Mahalanobis kernel satisfies the Mercer Theorem [24] and can be used as a valid kernel.

2.3. Mahalanobis Log Kernel

Apart from RBF kernel which is a typical stationary kernel, we introduce Log kernel in this subsection. The construction of Log kernel is shown in the following:

$$K_{\text{Log}}(\mathbf{x}_i, \mathbf{x}_j) = -\log(1 + ||\mathbf{x}_i - \mathbf{x}_j||^d), \quad (d \text{ is a parameter}).$$

We next show that Log kernel is indefinite. Note that the matrix version of Log kernel with considered data set takes the form:

$$K_{\text{Log}} = \begin{pmatrix} 0 & k_{11} & \ldots & k_{1n} \\ k_{21} & 0 & \ldots & k_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ k_{n1} & k_{n2} & \ldots & 0 \end{pmatrix}$$

where $k_{ij} = -\log(1 + ||\mathbf{x}_i - \mathbf{x}_j||^d), i, j \in \{1, 2, \ldots, n\}$.

The kernel matrix is symmetric, so it can be translated into a diagonal form. Note that the diagonal entries of the kernel matrix is uniformly 0. We can see that the trace of the matrix is 0, meaning that the summation of all the eigenvalues is 0, indicating there may exist positive and negative eigenvalues except all the values of the data set is 0. Therefore, we conclude that Log kernel is indefinite. Mahalanobis Log kernel is constructed as:

$$K_{\text{MLog}}(\mathbf{x}_i, \mathbf{x}_j) = -\log(1 + ||\mathbf{x}_i - \mathbf{x}_j||_{\text{m}}^d), \quad (d \text{ is a parameter}).$$

It is straightforward to check that Mahalanobis Log kernel is also indefinite. However, the kernel is conditional positive definite when $d \leq 2$ [25] which is suitable for SVM framework.

Considering the stationary property after Mahalanobis distance is incorporated, we can see that the new kernel is still stationary. We prove the property as follows. We have

$$||\mathbf{x}_i - \mathbf{x}_j||_m^2 = (\mathbf{x}_i - \mathbf{x}_j)^T \cdot S^{-1} \cdot (\mathbf{x}_i - \mathbf{x}_j)$$

where

$$S_{ij} = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_{ki} - \overline{\mathbf{x}}_{\cdot i})(\mathbf{x}_{kj} - \overline{\mathbf{x}}_{\cdot j}).$$

Therefore, we have

$$||\mathbf{x}_i + \mathbf{c} - (\mathbf{x}_j + \mathbf{c})||_m^2 = (\mathbf{x}_i - \mathbf{x}_j)^T \cdot \tilde{S}^{-1} \cdot (\mathbf{x}_i - \mathbf{x}_j).$$

If we can prove $\tilde{S} = S$, then the stationary property still holds. Note that

$$\begin{aligned} \tilde{S}_{ij} &= \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_{ki} + \mathbf{c}_{ki} - (\overline{\mathbf{x}}_{\cdot i} + \mathbf{c}_{ki}))(\mathbf{x}_{kj} + \mathbf{c}_{kj} - \overline{\mathbf{x}}_{\cdot j} + \mathbf{c}_{kj}) \\ &= \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_{ki} - \overline{\mathbf{x}}_{\cdot i})(\mathbf{x}_{kj} - \overline{\mathbf{x}}_{\cdot j}). \end{aligned}$$

Therefore, $\tilde{S}_{ij} = S_{ij}$ for $i, j \in \{1, 2, \ldots, n\}$ This proves the stationary property of Mahalanobis Log kernel.

2.4. Mahalanobis Power Kernel

The Power kernel is also a stationary kernel [26]. It is also known as the (unrectified) triangular kernel. The construction of Power kernel is shown in the following:

$$K_{\text{Power}}(\mathbf{x}_i, \mathbf{x}_j) = -||\mathbf{x}_i - \mathbf{x}_j||^d, \quad (d \text{ is a parameter})$$

We next show that Power kernel is indefinite. Note that the matrix version of Power kernel with considered data set is of the similar structure with Log Kernel where all the diagonal elements are 0. Using the same argument, we conclude that Power kernel is indefinite. Mahalanobis Log kernel is constructed as follows:

$$K_{\text{MPower}}(\mathbf{x}_i, \mathbf{x}_j) = -||\mathbf{x}_i - \mathbf{x}_j||_{\text{m}}^d, \quad (d \text{ is a parameter}).$$

It is straightforward to see that Mahalanobis Power kernel is also indefinite. However, the kernel is conditional positive definite when $d \leq 2$ [25] which is suitable for SVM framework. Regarding the stationary property, we can also prove that Mahalanobis Power kernel is stationary following the same procedures in the previous subsection.

3. Real Data Examples

3.1. Materials

In this paper, we adopt publicly available credit evaluation data sets from UCI Machine Learning Repository. One of the data sets is related to Japanese Credit Approval [27]. We remark that within the data set there is a good mix of attributes – continuous, nominal with small numbers of values, and nominal with larger numbers of values. A few missing values are substituted as 0. For nominal attributes A1, A4-A7, A9-A10, A12-A13, we uniformly replace a with 0, b with 1, and replace z with 25 under the same rule to get numerical attribute

expressions. In total, there are 690 instances with 15 attributes, of which 383 cases were granted credit and 307 cases were refused. Another data set is about German Credit Evaluation. The original dataset provided by Prof. Hofmann contains categorical/symbolic attributes. We used the numerical version of the data under name 'german.data-numeric'. The data was edited in Strathclyde University and several indicator variables were added to make it suitable for algorithms that need numerical attributes. In total, there are 1000 instances with 24 attributes, of which 700 cases were evaluated as 'good clients' and 300 cases were evaluated as 'bad clients'.

3.2. Models for Comparison

1. **Decision Tree:** Decision tree classifier is commonly used in data mining. Decision tree works as a predictive model that maps observations to their target values. In these tree structures, leaves represent classifications and branches represent conjunctions of features that lead to those classifications.
2. **K-Nearest Neighborhood:** The K-nearest neighbor algorithm is the simplest method among all machine learning algorithms. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its K nearest neighbors (K is a positive integer, typically small). If $K = 1$, then the object is simply assigned to the class of its nearest neighbor.
3. **Naive Bayes:** Naive Bayes Classifier is based on Bayes' theorem which has simplified assumptions on independence of variables, thus ensuring efficiency in parameter estimation and model evaluation. One of the advantages of Naive Bayes classifier lies in the capability of handling arbitrary number of variables and it only needs small size of training data. It is robust to noise by explicit calculation of probabilities.
4. **Linear Discriminant Analysis:** Linear discriminant analysis classifier assumes the conditional probability density functions of a sample $\mathbf{x}$ to be normally distributed, where the characteristic functions can be described as follows(Σ is a full rank matrix):

$$\Phi(\mathbf{x}|y=k) = e^{i\mu_k'\mathbf{t}-\frac{1}{2}\mathbf{t}'\Sigma\mathbf{t}},\ k=0,1.$$

The class of the sample $\mathbf{x}$ is determined according to the log likelihood ratio values:

$$Decision(\mathbf{x}) = \Sigma^{-1}(\mu_1-\mu_0)\cdot\mathbf{x} - \frac{1}{2}(T-\mu_0\Sigma^{-1}\mu_0+\mu_1\Sigma^{-1}\mu_1).$$

4. Experimental Results

In this section, we first report the performance comparison between Mahalanobis distance induced stationary kernels and the original stationary kernels. We then compare our proposed Mahalanobis Kernels with a number of state-of-the-art models. In the experimental setup, 10 times 5-fold cross validation are

True \ Result	P'	N'
P	True Positive (TP)	False Negative (FN)
N	False Positive (FP)	True Negative (TN)

Table 1: Definitions

conducted on the considered data sets. Overall accuracy, Type I accuracy, and Type II accuracy are used to measure the performance of the models.

In the context of classification, suppose the two true classes are P (Positive) and N (Negative), while the predicted positive and negative classes are P' and N', respectively. This is illustrated by the table below:

The overall accuracy is the percentage of correctly predicted instances and is denoted as

$$\text{Overall} = \frac{TP + TN}{P + N}.$$

Type I and Type II accuracy, respectively, measure the class-specific accuracies and they are denoted as follows:

$$\text{Type I} = \frac{TN}{TN + FP} \quad \text{and} \quad \text{Type II} = \frac{TP}{TP + FN}.$$

4.1. Mahalanobis RBF Kernels vs RBF Kernels

Figures 1 to 4 present the performance of Mahalanobis RBF Kernel in conjunction with SVM when compared to RBF Kernel in credit risk evaluation. The results on Japan Data are summarized in Figures 1 and 2.

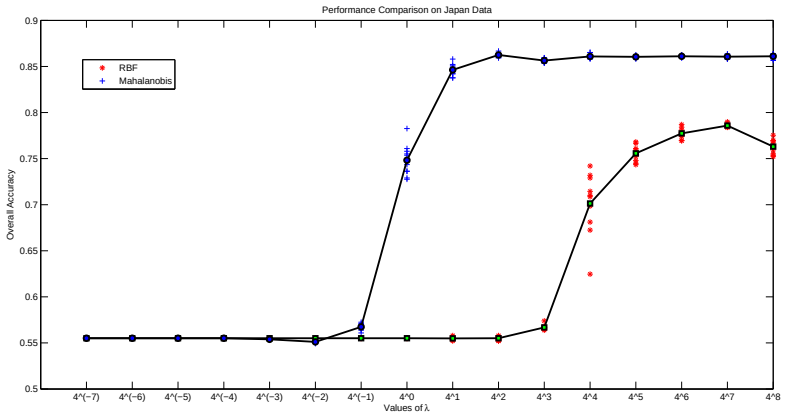
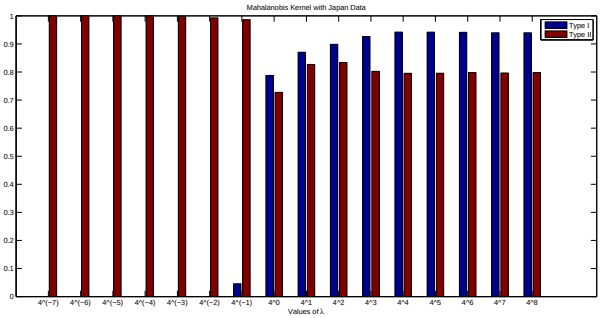
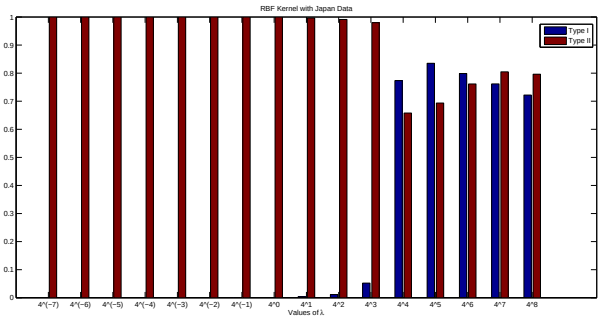


Figure 1: Overall Accuracy between Mahalanobis RBF Kernel and RBF Kernel in Japan Data

Figure 1 shows the overall accuracy of the two considered kernels with the same parameter $\lambda \in \{4^{-7}, 4^{-6}, \ldots, 4^{8}\}$. Line in blue '+' with '$\circ$' measures the average overall accuracy for Mahalanobis RBF kernel. Line in red '$\star$' with green '$\square$' measures the average overall accuracy for RBF kernel. With the increment of λ, we can see that the accuracy keeps steady and then increases drastically. The best performance is achieved in $\lambda = 4^2$ for Mahalanobis RBF kernel and $\lambda = 4^7$ in RBF Kernel. The superiority of Mahalanobis RBF Kernel over RBF Kernel on Japan Data is clearly demonstrated. Considering all the possible λ in the respected kernels, we can see that Mahalanobis RBF kernel is better than RBF Kernel. Besides, the overall accuracy of Mahalanobis RBF kernel is around 85% while RBF kernel can only achieve 75% at most when $\lambda > 4^2$. The value of λ has little effect on the performance of Mahalanobis RBF kernel when $\lambda > 4^2$ as we can see a relatively stable performance onwards. However, the effect of λ on RBF kernel is more evident as the performance of RBF kernel is firstly increasing when $\lambda \leq 4^7$ but showing a decrement when $\lambda = 4^8$.



(a)



(b)

Figure 2: Type I and Type II Accuracy between Mahalanobis RBF Kernel (2a) and RBF Kernel (2b) in Japan Data

Considering the class-specific accuracy, we can get some information from Figure 2. Figures 2a and 2b, respectively, report for two kernels on the type I

and II accuracy for Japan data for different values of λ.

In Figure 2a for Mahalaonbis RBF kernel, the distribution of Type I and Type II accuracy is quite unbalanced when $\lambda < 1$ where Type II accuracy dominates Type I accuracy. However, when $\lambda \geq 1$, Type II accuracy increases drastically and two types of accuracy becomes balanced. Similar patterns can be detected for RBF kernel except that the critical point is $\lambda = 4^4$ rather than 1. This shows the robustness of the Mahalanobis RBF kernel. Comparing the two kernels, we can conclude that Mahalanobis RBF kernel is dominantly better than RBF kernel in both Type I and Type II accuracies.

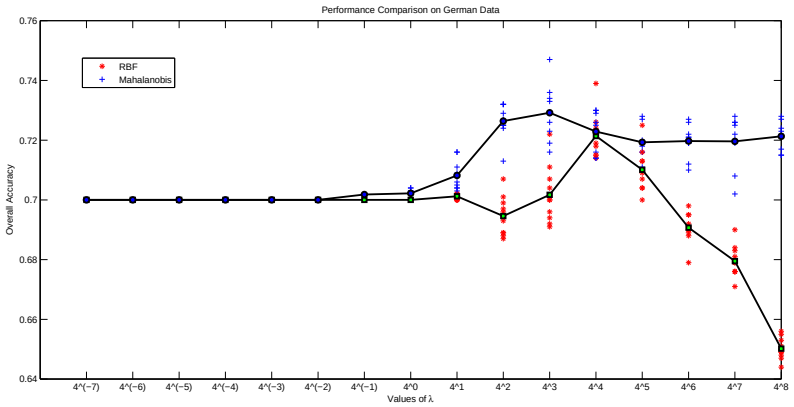
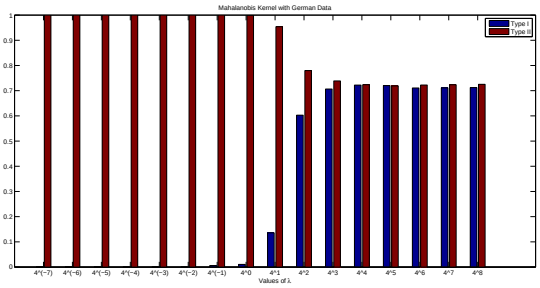
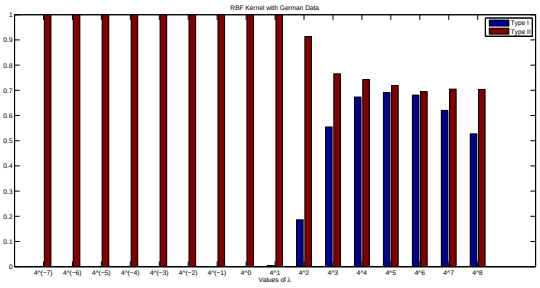


Figure 3: Overall Accuracy between Mahalanobis RBF Kernel and RBF Kernel in German Data

The results on German Data are summarized in Figures 3 and 4. With the increment of λ, we can see that the accuracy keeps steadily at 70% which is actually the percentage of good credit clients ratio in German data, meaning that the classifier is trained to overfit the data set. When $\lambda > 1$, the overall accuracy is increasing for Mahalanobis RBF kernel and achieves the best performance when $\lambda = 4^3$. While for RBF kernel, the overall accuracy is firstly decreasing and then increase to achieve the best performance when $\lambda = 4^4$. When $\lambda > 4^4$, however, the performance is decreasing steadily, showing the sensitivity of RBF kernel on λ. Meanwhile, we can see the superiority of Mahalanobis RBF kernel over RBF kernel with the respected λ. Regarding the type specific accuracy, we can see similar patterns with Figure 2 for Japan Data. One more conclusion we can draw is that RBF kernel is not very stable when λ is in a large range. The Type I accuracy is increasing first and then decreasing, while Type II accuracy decreases steadily. The performance of Mahalanobis RBF kernel is more stable with respect to the values of λ.



(a)



(b)

Figure 4: Type I and Type II Accuracy between Mahalanobis RBF Kernel (4a) and RBF Kernel (4b) in German Data

4.2. Mahalanobis Log Kernel Compared with Log Kernel

In Figures 5 to 6, we compare Mahalanobis Log kernel with Log kernel in terms of overall accuracy, Type I and Type II accuracy on the considered data sets. Blue color represents the average overall accuracy, green color represents the average Type I accuracy and red color represents the average Type II accuracy.

In Figure 5 for Japan Data, we can see that Mahalanobis Log Kernel overwhelmingly outperforms Log Kernel in overall accuracy, Type I and Type II accuracy: the three considered measures. The overall accuracy for MLog Kernel is around 86% while Log kernel only achieves 78% on average. Type I accuracy for Mlog Kernel is around 89% but Log kernel only gets 71% in average. The difference on Type II accuracy between MLog kernel and Log kernel is so large where on average MLog kernel achieves 84.25% and Log Kernel achieves 83.6%.

In Figure 6 for German Data, we can see that Mahalanobis Log Kernel out performs Log Kernel in terms of overall accuracy and Type I accuracy. The overall accuracy of MLog kernel is better than Log kernel (76.77% to 84.81% on average). Type I accuracy for MLog kernel is 56.94% on average but Log kernel can only get 48.26%. Here Type I accuracy actually refers to the ability

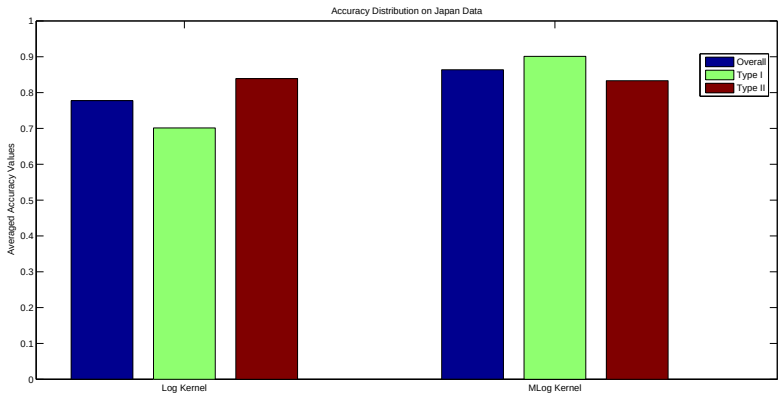


Figure 5: Accuracy Distribution on Japan Data with Log Kernel

of correctly classify a bad credit client to the 'bad' class, we can see that MLog kernel is more robust as Log kernel misclassified a number of bad credit clients which may bring burden to bank corporations. Type II accuracy for MLog kernel is 85.17% and 86.28% for Log kernel. Though the Type II accuracy for MLog kernel is slightly inferior to Log kernel, we can see that overall MLog kernel can provide a more robust and practical evaluation model.

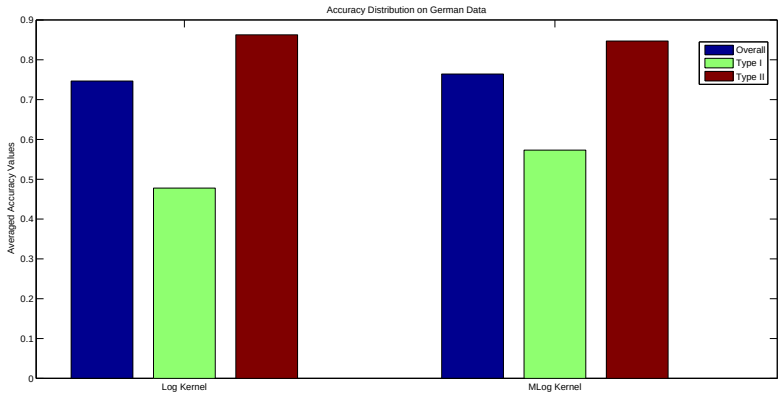


Figure 6: Accuracy Distribution on German Data with Log Kernel

4.3. Mahalanobis Power Kernels vs Power Kernels

In Figures 7 to 8 we compare Mahalanobis Power kernel with Power kernel in terms of overall accuracy, Type I and Type II accuracy on the considered data sets. Blue color represents the average overall accuracy, green color represents

the average Type I accuracy and red color represents the average Type II accuracy. Left bars represent Power kernel and right bars represent Mahalanobis Power kernel.

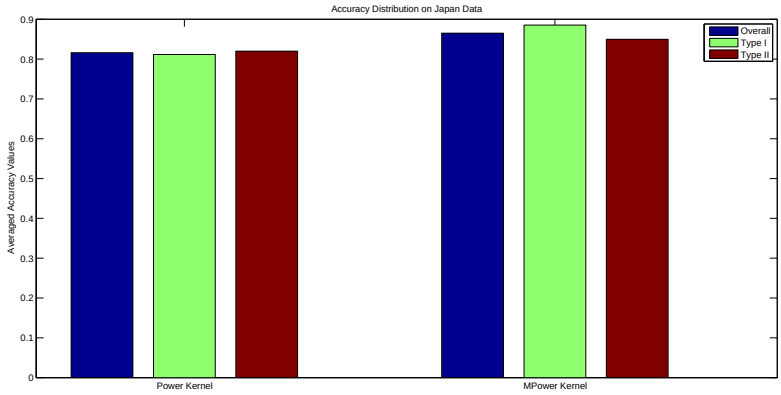


Figure 7: Accuracy Distribution on Japan Data with Power Kernel

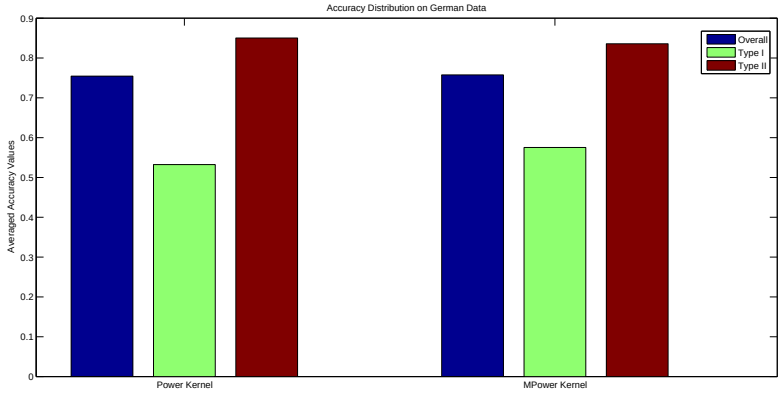


Figure 8: Accuracy Distribution on German Data with Power Kernel

In Figure 7 for Japan Data, we can see that Mahalanobis Power Kernel out performs Power Kernel on all the considered measures. Overall accuracy for MPower kernel is 86.12% on average but it is 81.90% for Power kernel. Type I and Type II accuracies for MPower kernel are 87.95% and 84.73%, respectively while 81.55% and 82.34% for Power kernel, respectively. In Figure 8 for German Data, we can see that Mahalanobis Power Kernel out performs Power Kernel in terms of providing a more robust classifier. Overall accuracy for MPower Kernel and Power Kernel is similar to each other with MPower kernel being slightly

better. Type I accuracy for MPower kernel is 57.48% on average, Power kernel can achieve 52.56%. Similar to the previous analysis, we can conclude that MPower kernel is more robust since misclassification of bad credit clients may do harm to bank corporations. Type II accuracy for MPower kernel is 83.89% and 85.31% for Power kernel. Though the Type II accuracy for MPower kernel is slightly inferior to Power kernel, we can see that MPower kernel overall can provide a more robust and practical evaluation model.

4.4. Model Comparisons: Mahalanobis Distance Induced Stationary Kernels vs Others

In this section, we compare Mahalanobis distance induced stationary kernels with SVM to a number of the state-of-the-art credit evaluation models, results are illustrated in Table 7. On the left hand side of the table, MLog, MPower, MRBF represent Mahalanobis Distance induced Log kernel, power kernel, and RBF kernel, respectively. On the right hand-side, NB stands for Naive Bayes, DT stands for Decision Tree, LDA means Linear Discriminant Analysis classifier, $\text{Knn}_1, \text{Knn}_5, \text{Knn}_{10}$ represent K-nearest neighborhood with $K = 1, 5, 10$, respectively. In the table, average accuracies in 10 times 5-fold cross validation on the data sets are recorded. Best performance for each considered measure is marked in bold face.

We can see that best overall accuracy is achieved by MLog kernel for all considered data sets. Two other Mahalanobis distance induced kernels also perform well. In Japan Data, we can see that MPower kernel achieves 86.72% and MRBF kernel achieves 86.10%. They both rank in the top performing models. In German Data, we can see that MPower kernel achieves 75.92% and MRBF kernel achieves 72.38%. Together with MLog kernel, they rank in the top 3 performing models. These results demonstrate that Mahalanobis Distance Induced Kernel can compete with other models.

Regarding the Type I accuracy, we can see that LDA shows best performance compared with all the other models. In Japan data, the Type I accuracy for LDA is 94.04% on average. In German Data, the Type I accuracy for LDA is 71.12% on average. When we focus on the LDA classifier, we find that LDA is the best model among the state-of-the-art models. However, Type II accuracy in LDA classifier is not quite satisfying compared to Mahalanobis kernels.

It is interesting to see that best Type II accuracy is shown in different models for different data sets. In Japan Data, the best Type II accuracy is shown in Naive Bayes model, achieving 90.57% in average. However the Type I accuracy in Naive Bayes model for Japan data is 67.22%, which is among the worst performing list of models. In German Data, the best Type II accuracy is shown in K-nearest neighborhood model with $K = 10$, achieving 89.58% on average. But the Type I accuracy correspondingly is only 25.34% on average. This illustrates the overfitting problem has occurred in model training, whereby rendering a relatively poor classifier from the perspective of class specific accuracy.

Naive Bayes model in the state-of-the art models is in the top performing range in terms of overall accuracy. The overall accuracy in Japan Data is 80.16% which ranks the third in state-of-the art models. In German Data the overall

accuracy is 72.33 on average, ranking the first in the state-of-the-art models. However, they still cannot compete with Mahalanobis kernel models.

Decision Tree model is the top 3 model in state-of-the art models. The overall accuracy on average in Japan data is 83.22%and in German data 69.61%. We find that Type I accuracy in DT model is not satisfactory when compared to the other top ranking models. One possible reason for the poor performance in DT model is the overfitting problem where the model is more fit for describing the behavior of good credit clients.

In the K-nearest neighbor model, we can see that the performance varies when different K is chosen. In Japan Data, with the increment of K, the overall accuracy is increasing then decreasing. The Type II accuracy is at most 68.04%, not competitive with other models except decision tree model. In German Data, a steady increment can be observed in overall accuracy with the increment of K. We note that Type II accuracy in German data is at most 43.63%, which is not good satisfactory.

Table 2: A Comparison with State-of-the-art Models

		MLog	Mpower	MRBF	NB	DT	LDA	Knn_1	Knn_5	Knn_{10}
Japan	Overall	**86.81**	86.72	86.10	80.16	83.22	86.10	71.86	75.78	75.32
	Type I	90.12	88.62	94.00	67.22	81.34	**94.04**	68.04	67.23	64.76
	Type II	84.25	85.32	79.74	**90.57**	84.71	79.91	74.95	82.67	83.78
German	Overall	**76.30**	75.92	72.38	72.33	69.61	72.06	66.81	68.59	70.23
	Type I	57.39	57.37	70.40	61.63	48.05	**71.12**	43.63	30.10	25.34
	Type II	84.47	83.92	73.22	76.92	79.05	72.46	76.80	85.17	**89.58**

4.4.1. Performance of Indefinite Mahalanobis Stationary Kernels

In particular, we find that MLog kernel and Mpower kernel tend to perform stably satisfactory compared to all the other models even when Mahalanobis RBF kernel is considered. Hence in the following we would like to check the performance of the two indefinite Mahalanobis stationary kernels with varying degree.

Table 3 is related to MLog kernel. It can be seen from the tale that overall the performance is stable in Japan data, maintaining the overall accuracy in around 86%. The best performance is achieved when $d = 2$. The Type I accuracy is generally increasing and Type II accuracy is decreasing with the degree d decreases. In German data set, the overall accuracy of MLog kernel is gradually increasing and then decreases slightly, with the best performance achieved when $d = \frac{1}{6}$. However, the tendency of Type I accuracy and Type II accuracy is completely different from that in Japan data. Type I accuracy is generally decreasing and Type II accuracy is increasing with the decrement of degree d. To seek a balanced performance of MLog kernel, we can see that $d = \frac{1}{2}$ is preferred in Japan Data and $d = 1$ for German data.

Table 4 is related to MPower kernel. It can be seen from the tale that overall the performance is stable (except for $d = 3$) in Japan data, maintaining the overall accuracy in around 86%. The best performance is achieved when

Table 3: Performance of MLog Kernel with different degree

	Deg / Acc	3	2	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{7}$	$\frac{1}{8}$	$\frac{1}{9}$	$\frac{1}{10}$
Japan	Overall	86.25	**86.59**	86.16	86.29	86.09	85.96	85.99	85.96	86.09	86.03	86.03	85.86
	Type I	86.62	88.51	89.25	91.05	91.56	91.76	92.26	92.30	92.65	92.39	**92.67**	92.48
	Type II	**85.92**	85.14	83.75	82.57	81.76	81.33	80.98	80.94	80.99	80.98	80.69	80.54
German	Overall	75.66	76.03	76.88	76.43	76.82	76.99	77.30	**77.33**	77.03	77.24	76.57	76.17
	Type I	54.62	56.39	**58.41**	56.44	55.14	52.93	51.42	48.54	45.53	43.90	40.44	37.16
	Type II	84.73	84.51	84.77	85.04	86.17	87.41	88.40	89.78	90.60	91.64	92.25	**93.04**

$d = 2$. Type I accuracy is generally increasing and Type II accuracy is decreasing with the degree d decreases. In German data set, the overall accuracy of MPower kernel is gradually increasing and then decreases slightly, with the best performance achieved when $d = \frac{1}{6}$. However, the tendency of Type I accuracy and Type II accuracy is completely different from that in Japan data. Type I accuracy is generally decreasing and Type II accuracy is increasing with the decrement of degree d. Exception happened when $d = 3$ for the considered 2 data sets. The Overall accuracy when $d = 3$ is not satisfying. Mpower kernel with $d = 2$ shows the best Type I accuracy but the Type II accuracy is not satisfying. To seek a balanced performance of MPower kernel, we can see that $d = \frac{1}{3}$ is a preferred choice for both Japan and German data.

Table 4: Performance of MPower Kernel with different degree

	Deg / Acc	3	2	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{7}$	$\frac{1}{8}$	$\frac{1}{9}$	$\frac{1}{10}$
Japan	Overall	57.80	86.00	86.26	86.39	**86.43**	86.28	86.10	85.99	85.96	86.13	86.20	85.91
	Type I	57.47	**94.10**	87.95	89.79	90.81	91.25	91.52	91.63	91.68	91.87	92.25	91.96
	Type II	58.05	79.48	**85.01**	83.76	82.98	82.34	81.79	81.53	81.53	81.59	81.35	81.07
German	Overall	68.91	71.72	76.51	76.37	76.98	77.00	77.25	**77.45**	77.42	77.34	77.28	76.85
	Type I	48.83	**71.51**	58.67	56.32	56.05	54.50	53.37	51.80	49.91	48.30	46.33	44.10
	Type II	77.57	71.79	84.16	85.03	86.01	86.76	87.49	88.54	89.27	89.91	90.72	**91.04**

4.5. Statistical Significance

In order to show the significance for the analysis of the results, we conduct 20 runs of 5-fold cross validations on the given data sets. The performances of the considered algorithms were compared to our proposed method through t-test. We made null hypothesis that other algorithms yield larger evaluation accuracy values compared to our Mahalanobis stationary kernel methods. The p-values of the statistical tests are reported in Table 5. It is clearly shown that we should reject the null hypothesis and accept the alternative hypothesis that our Mahalanobis stationary kernel methods yield larger accuracy values compared to other classical algorithms.

Table 5: Statistical Test for the Analysis of the Results

	p-values / Methods	DT	NB	LDA	Knn_1	Knn_5	Knn_{10}
Japan	MPOWER	9.0151e-13	2.0207e-20	0.0191	7.0697e-22	1.7426e-20	3.2450e-20
	MLOG	1.9017e-13	1.1845e-20	1.3714e-04	3.0905e-22	1.6864e-20	7.9165e-21
	MRBF	1.3788e-12	4.4483e-20	0.0013	9.0125e-22	6.2057e-20	4.6148e-20
German	MPOWER	1.2806e-15	9.2100e-17	4.5143e-18	9.2560e-25	1.0890e-18	6.5484e-19
	MLOG	1.4201e-14	3.0688e-16	1.0295e-16	6.5396e-26	7.5786e-18	1.1776e-18
	MRBF	6.0878e-07	4.7333e-04	1.9281e-04	3.1433e-17	1.2344e-12	1.3842e-11

We further checked if the Mahalanobis distance induced kernels perform better than other distance-measure based kernels. We introduce Euclidean, Cosine, Correlation and Chebychev distances for comparison of the methods via 10 runs 5-fold cross validations. Averaged AUC values with standard deviations are reported in Table 6. Results show that Mahalanobis distance provides a proper description on the relationship of data.

Table 6: Kernels with Mahalanobis Distance vs Various Distances

	Distance / Methods	Mahalanobis	Cosine	Euclidean	Correlation	Chebychev
Japan	POWER	**0.8639 ±0.0038**	0.7951±0.0072	0.7864±0.0105	0.7972±0.0087	0.6852±0.0076
	LOG	**0.8643 ±0.0047**	0.7807±0.0044	0.7693±0.0098	0.7848±0.0042	0.6959±0.0092
	RBF	**0.8599 ±0.0029**	0.7184±0.0044	0.7372±0.0103	0.7228±0.0021	0.6633±0.0079
German	POWER	**0.7698 ± 0.0049**	0.7298 ± 0.0034	0.7496 ± 0.0071	0.7325 ± 0.0046	0.5403 ± 0.0205
	LOG	**0.7634± 0.0046**	0.6673± 0.0024	0.7455± 0.0082	0.6726± 0.0035	0.5860± 0.0196
	RBF	**0.7234±0.0061**	0.5908±0.0036	0.7168±0.0077	0.5903±0.0032	0.5343±0.0086

4.6. Large-scale Credit Evaluation

To test the algorithm on large-scale problems, we introduced a dataset of default payments in Taiwan [29] and compared the predictive accuracy of probability of default among six data mining methods. This research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable. The dataset contains 30000 credit card clients, 23 attributes ranging from gender, education, age and amount of bill statement, etc. We compared our algorithm with Euclidean distance based algorithms and other classical machine learning algorithms. The experiments were conducted through 5-fold cross validations and performances were measured using averaged AUC values. Results in Table 7 show that our algorithm is better than the other methods. In particular, the Type I and Type II accuracies in our methods are more balanced. In comparison, RBF kernel with Euclidean distances and decision tree algorithms tend to train over-fitted models.

Table 7: Large Scale Credit Evaluation: Taiwan Credit Default

	MLog	Mpower	MRBF	Log	Power	RBF	NB	DT	LDA	Knn_1	Knn_5	Knn_{10}
Overall	69.12	69.76	69.51	62.83	61.77	51.01	62.56	59.67	67.04	57.09	58.86	60.09
Type I	74.24	75.50	76.46	63.40	61.98	98.72	61.42	35.46	72.84	56.88	58.95	64.28
Type II	63.99	64.00	62.55	62.25	61.53	3.29	63.70	83.83	61.22	57.29	58.75	55.73

5. Conclusions

In this paper, we have proposed Mahalanobis distance induced kernels in conjunction with SVM for credit risk evaluation, with a focus on stationary kernels. The stationary property when Mahalanobis distance incorporated is also illustrated. Through comparison with Euclidean distance and various other distances based stationary kernels, Mahalanobis distance based kernels are more robust and more fit for describing the behavior of the credit risk data

sets. In order to illustrate the power of Mahalanobis stationary kernel models, we introduce a number of state-of-the-art models for comparison. Results show that Mahalanobis stationary models can compete with state-of-the-art models. In particular, the indefinite Mahalanobis kernels tend to perform stably satisfactory. The newly constructed stationary kernels may shed some light on SVM based models for credit risk evaluations.

Acknowledgments

The authors would like to thank the anonymous referees for their helpful comments and suggestions. This work is supported by National Natural Science Foundation of China NSFC Nos. 11626229 and 11671158, Research Grants Council of Hong Kong under grant number 17301214, IMR and RAE Research Fund from Faculty of Science, the University of Hong Kong, and Natural Science Foundation of SZU (grant no. 2017058).

References

[1] Vapnik, V.N. (1995) The nature of statistical learning theory. Springer, New York.

[2] Carrizosa, E., Morales, D.R.(2013) Supervised classification and mathematical optimization. Computers & Operations Research. 40: 150–165.

[3] Pan, B., Lai, J., Shen, L., (2014) Ideal regularization for learning kernels from labels. Neural Networks. 56: 22–34.

[4] Pan, B., Chen, W., Xu, C., Chen. B. (2016) A Novel Framework for Learning Geometry-Aware Kernels. IEEE Transactios on Neural Networks and Learning Systems. 27: 939–951.

[5] Chen. W., Zhao, Y., Pan, B., Chen. B. (2016) Supervised kernel nonnegative matrix factorization for face recognition. Neurocomputing. 205: 165–181.

[6] Scholkopf B., Smola A.J. (2001) Kernels. Learning with kernels, MIT Press, London, England, Ed.1, 29–38.

[7] Van Gestel T., Baesens B., Garcia J., Van Dijcke P. (2003) A support vector machine approach to credit scoring. Bank en Financiewezen 2: 73–82.

[8] Huang Z., Chen H.C., Hsu C.J., Chen W.H., Wu S.S. (2004) Credit rating analysis with support vector machines and neural networks: A market comparative study. Decision Support Systems, 37: 543–558.

[9] Lai K.K., Yu L., Huang W., Wang S.Y. (2006) A novel support vector machine metamodel for business risk identification. Lecture Notes in Artificial Intelligence. 4099: 480–484.

[10] Yu L., Lai K.K., Wang S.Y., Zhou L.G. (2007) A least squares fuzzy svm approach to credit risk assessment. Fuzzy Information and Engineering (ICFIE), 40: 865–874.

[11] Yao P., Lu Y.H. (2011) Neighborhood rough set and SVM-based hybrid credit scoring classifier. Expert Systems with Applications, 38(9): 11300–11304.

[12] Yao X. ,Yu L.(2012) A fuzzy proximal support vector machine model and its application to credit risk analysis. Systems Engineering: Theory & Practice, 32(3): 549–554.

[13] Yi B.H., Zhu J.J., Li J. (2016) Imbalanced data classification on micro-credit company customer credit risk assessment using improved SMOTE support vector machine. Chinese Journal of Management Science, 24(3): 24–30.

[14] Zhang M., Zhou Z.F. (2009) An evaluation model for credit risk of enterprise based on multi-objective programming and support vector machines. China Soft Science, (4): 185–190.

[15] Zhang Q., Hu L.Y.,Wang J. (2015) Study on credit risk early warning based on logit and SVM. Systems Engineering: Theory & Practice, 35(7): 1784–1790.

[16] Tsang I.W., Kwok J.T., Li S.T. (2006) Learning with kernels in Mahalanobis one-class support vector machines. International Joint Conference on Neural Networks. Vancouver, BC, Canada. 1169–1175.

[17] Santiago P., Jayro, Torres R., et al. (2015) Using Generalized Entropies and OC-SVM with MahalanobisKernel for Detection and Classification of Anomalies in Network Traffic. Entropy,17(9): 6239-6257.

[18] Zhang Y., Xie F.D., Huang D., Ji M. (2010) Support vector classifier based on fuzzy c-means and Mahalanobis distance. Journal of Intelligent Information Systems. 35: 333–345.

[19] James N.K.L., Hu Y.X. (2013) Support vector regression with kernel Mahanobis measure for financial forecast. Time Series Analysis, Models.& Applications, 47: 215–227. Springer.

[20] Lee Y.C., Teng H.L. (2009) Predicting the financial crisis by Mahanobis-Taguchi system - Examples of Taiwan's electronic sector. Expert Systems with Applications. 36: 7649–7478.

[21] Cortes C.,Vapnik V.N. (1995) Support-vector networks. Machine Learning, 20: 273–297.

[22] Aizerman M.A., Braverman E.M.,Rozonoer L.I. (1964) Theoretical foundations of the potential function method in pattern recognition learning, Automation and Remote Control, 25: 821–837.

[23] Vedaldi A., Zisserman A. (2012) Efficient additive kernels via explicit feature maps. IEEE Transaction on Pattern Analysis and Machine Intelligence, 34: 480–492.

[24] Mercer, J. (1909) Functions of positive and negative type and their connection with the theory of integral equations, Philosophical Transactions of the Royal Society A, 209 (441C458): 415–446.

[25] Boughorbel S., Tarel J.P., Boujemaa N. (2005) Conditionally positive definite kernels for svm based image recognition, IEEE International conference on Multimedia and Expo, 113-116.

[26] Sahbi H., Fleuret F. (2002) Scale-invariance of support vector machines based on the triangular kernel. Research Report, RR-4601, INRIA.

[27] Quinlan J.R. (1993) C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc.

[28] Jiang H., Ching W. (2012) Correlation kernels for support vector machines classification with applications in cancer data, Computational and Mathematical Methods in Medicine, Vol. 2012, Article ID 205025, 7 Pages, http://dx.doi.org/10.1155/2012/2050252012.

[29] Yeh, I. C., Lien, C. H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2), 2473-2480.