이학석사 학위논문

Default Risk Modeling and Machine Learning

2014 년 8 월

서울대학교 대학원

수리과학부

Jianyu Kang

Master's degree dissertation

Default Risk Modeling and Machine Learning

August 2014

Seoul National University

Graduate School of Mathematics

Jianyu Kang

Abstract

This paper will be focused on applying machine learning to predict the possibilities for firms to default. The data selected for this modeling are firms from United States between 2008 and 2012. We will use logistic regression and support vector machines, two major classification model from machine learning to forecast the risk of default. The result will be compared when different features are selected. Furthermore, we will discuss the strength of each method by comparing the result.

Key components: default risk prediction, machine learning, logistic regression, support vector machines, stock prices

Student Number: 2012-23906

Contents

Chapter 1

Introduction

The financial crisis that first broke out in the US around the summer of 2007 and crested around the autumn of 2008 had destroyed $34.4 trillion of wealth globally by March 2009. The lost wealth, $34.4 trillion, is more than the 2008 annual gross domestic product (GDP) of the US, the European Union and Japan combined. This wealth deficit effect would take at least a decade to replenish even if these advanced economies were to grow at mid-single digit rate after inflation and only if no double dip materializes in the markets. At an optimistic compounded annual growth rate of 5%, it would take over 10 years to replenish the lost wealth in the US economy.

Therefore, if there is a method that can provide a more accurate forecast of the bankruptcy of those companies, then hundreds of billions dollars can be saved from such tragedy. This study was conducted for the

purpose of applying machine learning methods to forecast the risk of default targeted the stock listed companies in United States.

The core of machine learning deals with representation and generalization. Representation of data instances and functions evaluated on these instances are part of all machine learning systems. Generalization is the property that the system will perform well on unseen data instances; the conditions under which this can be guaranteed are a key object of study in the subfield of computational learning theory. This paper will be focus on applying machine learning to default risk of firms.

Chapter 2

Machine Learning Methods

Machine learning is a branch of artificial intelligent. Concerns the construct data and study of system that can learn from data. From the most commonly used field such as distinguish spam and non-spam emails. Up to glut and productive capacities in varies industries. Such as ship building, semi conduct, petro chemical and etc.

In this Chapter, we will introduce the methods selected from machine learning: logistic regression and support vector machines (SVM). Furthermore, we will explain the detail methodology of how patterns are recognized, then applied them to the test data.

2.1 Logistic regression

Logistic regression is a type of probabilistic statistical classification model. It is used for predicting the outcome of a categorical dependent variable (i.e., a class label) based on one or more predictor variables (features). The probabilities describing the possible outcomes of a single trial are modeled, as a function of the explanatory (predictor) variables, using a logistic function. "Logistic regression" is used to refer specifically to the problem in which the dependent variable is binary.

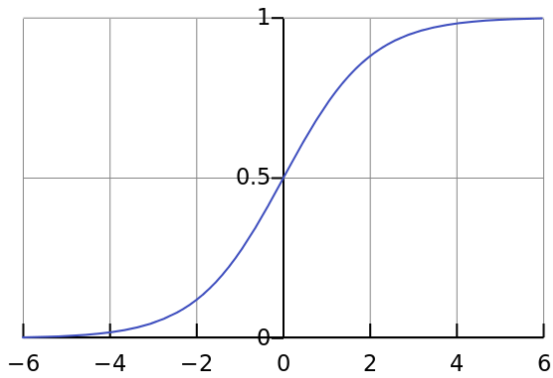
First, we set the classification variable y to be either 0 or 1 due to the binary classification nature of logistic regression. Then x is selected to denote a feature vector. Next, the training sample is defined to be $D = \{(x^{(i)}, y^{(i)})\}_{i=1,\cdots,n}$. Given any $D = \{(x^{(i)}, y^{(i)})\}_{i=1,\cdots,n}$ logistic regression will construct the classification model and predicts the y to be either 0 or 1. In order to do so, we assume there exists a weighted vector $w = (w_0, w_1, \cdots, w_d)^t$ and a function $h_w(x)$ such that

$$h_w(x) = P\ (\ y = 1|\ x;\ w)$$

The key to logistic regression is to define the logistic function $g(t)$ to be

$$g(t) = \frac{1}{1 + e^{-t}}$$

The image below shows a graph of $g(t)$. As we can see from the graph that $g(t)$ is has the domain range of (0, 1).



Next, we apply weighted vector and x into $g(t)$ and obtain

$$h_w(x) = g(w^T \cdot x) = \frac{1}{1 + e^{-w^T \cdot x}}$$

With the data $D = \{(x^{(i)}, y^{(i)})\}_{i=1,\cdots,n}$ given, we need to find $\boldsymbol{w}$ that is able to minimize the following cost function:

$$J(w) = \sum_{i=1}^{n} (y^{(i)} - h_w(x^{(i)}))^2$$

Through the basic definition of likelihood, by maximizing the likelihood of following function $L(w)$ will obtain the optimal parameter $\boldsymbol{w}$

$$L(w) = \prod_{i=1}^{n} p(y^{(i)} | x^{(i)}; w)$$

Since the regression coefficients are usually estimated using maximum likelihood estimation. Therefore, the function above will be used to calculate the parameter $\boldsymbol{w}$**.**

Formally, the outcomes y are described as being Bernoulli-distributed data, where each outcome is determined by an unobserved probability that is specific to the outcome at hand, but related to the explanatory variables. Therefore, we assume

$$p(\,y = 1 \mid\, x; w) =\, h_w(x) =\, g(\,w^T \cdot x\,)$$

$$p(\,y = 0 \mid\, x; w) = 1 -\, h_w(x) = 1 -\, g(\,w^T \cdot x\,)$$

Which can be converted into

$$p(\,y \mid\, x; w) =\, h_w(x)^y \cdot (\,1 -\, h_w(x))^{1-y} \sim Bernoulli\,\big(h_w(x)\big)$$

Applying the function above back to the likelihood function and we can obtain

$$L\,(w) = \prod_{i=1}^{n} h_w\big(x^{(i)}\big)^{y^{(i)}} \cdot (\,1 -\, h_w\big(x^{(i)}\big))^{1-y^{(i)}}$$

Now, in order to simplify the calculation, we apply the log likelihood to the equation above and result in

$$l\,(w) =\, log\, L\,(w)$$

$$= \prod_{i=1}^{n} \Big\{\, y^{(i)}\, log\, h_w\big(x^{(i)}\big) + \big(1 - y^{(i)}\big) \log\Big(1 - h_w\big(x^{(i)}\big)\Big)\Big\}$$

$$= \prod_{i=1}^{n} \big\{y^{(i)} \log g(\,w^T \cdot x\,) + \big(1 - y^{(i)}\big) \log\big(1 - g(\,w^T \cdot x\,)\big)\big\}$$

Then we take the derivative of $l\,(w)$ and we will obtain.

$$\frac{\partial l\,(w)}{\partial w_k} = \sum_{i=1}^{n} \left\{ y^{(i)} \frac{g'(\,w^T \cdot x^{(i)}\,)}{g(\,w^T \cdot x^{(i)}\,)} {x_k}^{(i)} + (1 - y^{(i)}) \frac{g'(\,w^T \cdot x^{(i)}\,)}{1 - g(\,w^T \cdot x^{(i)}\,)} {x_k}^{(i)} \right\}$$

$$= \sum_{i=1}^{n} \Big\{ y^{(i)} \Big(1 - g\big(\,w^T \cdot x^{(i)}\,\big)\Big) + (1 - y^{(i)}) g\big(\,w^T \cdot x^{(i)}\,\big) \Big\} {x_k}^{(i)}$$

$$= \sum_{i=1}^{n} \{ y^{(i)} - g(w^T \cdot x^{(i)}) \} x_k{}^{(i)}$$

And the second derivative of *l (w)* is

$$\frac{\partial^2 l\,(w)}{\partial w_k \partial w_r} = - \sum_{i=1}^{n} g'(w^T \cdot x^{(i)}) x_k{}^{(i)} x_r{}^{(i)}$$

Due to the nature of logistic regression, iterative methods such as gradient descent algorithm is used to find *w*. Gradient descent algorithm constructs a loop that is able to update parameter to achieve optimal value.

$$w\ (new) = w\ (old) + \ \alpha\ \nabla\ l(w)|_{w=w(old)}$$

The key to obtain *w* in this equation is α, the learning rate. Small α will cause the algorithm converge too slowly. On the other hand, if α is too large, it may result in not converge and oscillate.

Applying the derivative of log likelihood to *w (new)* and we will obtain

$$w\ (new) = w\ (old) + \alpha \sum_{i=1}^{n} \{ y^{(i)} - g(w^T (old) \cdot x^{(i)}) \} x_k{}^{(i)}$$

Since the Hessian matrix *H (w)* is given by

$$H\ (w) = \ - \sum_{i=1}^{n} g'(w^T \cdot x^{(i)}) x^{(i)} (x^{(i)})^T$$

For any $\mathbf{v} \in \ \mathbb{R}^{\boldsymbol{d}}$,

$$\mathbf{v}^{\boldsymbol{T}}\ x^{(i)} (x^{(i)})^T\ \mathbf{v} = \left\| (x^{(i)})^T\ \mathbf{v} \right\|^{\mathbf{2}}\ \geq 0$$

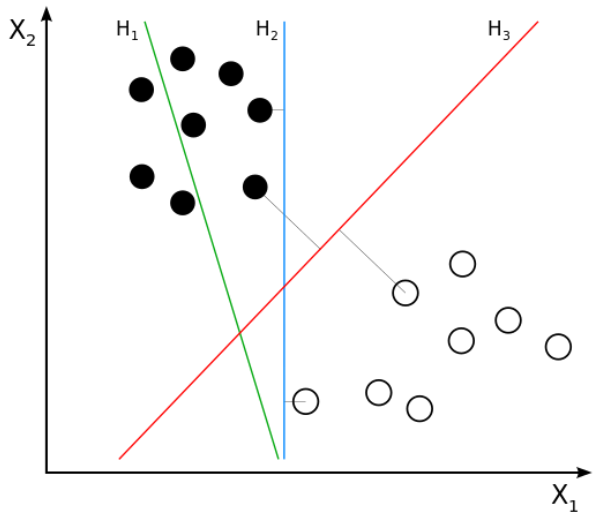
Since $g'(t) = \frac{e^{-t}}{(1+e^{-t})^2} > 0$, $-H(w)$ is positive semi-definite, the log likelihood function $l(w)$ is a concave function, which means there exit a maximum. Through the gradient descent algorithm we are able to compute the maximum, which is the optimal parameter that minimizes the cost function.

For $i = 1, 2, \cdots, n$. we are able to forecast the classification vector y to be

$$y^{(i)} = \begin{cases} 1 \text{ if } g\left(w^T \cdot x^{(i)}\right) \geq 0.5 \\ 0 \text{ otherwise} \end{cases}$$

2.2 Support Vector Machines

Support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification analysis. Given a set of training data, each is classified as category one or two, a SVM training algorithm constructs a model that assigns new data into one category or the other, making it a non-probabilistic binary linear classifier. A SVM model is a representation of the data as points in space, mapped so that the data of the separate categories are separated by a clear gap that is as wide as possible. New data are then mapped into that same space and classified to a category based on which side of the gap they fall on.



For given data $D = \{(x^{(i)}, y^{(i)})\}_{i=1,\cdots,n}$ where the $y^{(i)}$ is either 1 or -1, indicating the class $x^{(i)}$ belongs. We aim to find the maximum-margin hyperplane that separates the points within class $y^{(i)} = 1$ from those within class $y^{(i)} = -1$. Any hyperplane can be written as the set of points $x^{(i)}$ satisfying $w \cdot x - b = 0$. The parameter $\frac{b}{\|w\|}$ determines the offset of the hyperplane from the origin along the normal vector $\boldsymbol{w}$.

When the training data are linearly separable, we select two hyperplanes that separate the data with no points between them, and then try to maximize the gap in between. The region bounded by the hyperplanes is called "the margin". These hyperplanes can be described by $w \cdot x - b = 1$ or $w \cdot x - b = -1$.

By using geometry, we find the distance between these two hyperplanes is $\frac{2}{\|w\|}$, therefore, the next step is to minimize $\| w \|$ in order to maximize the gap. To prevent data points from falling into the margin, we add the following constraint: for each i

$$\begin{cases} w \cdot x^{(i)} - b \geq 1 & \text{for } x^{(i)} \text{ belongs to the first class} \\ w \cdot x^{(i)} - b \leq -1 & \text{for } x^{(i)} \text{ belongs to the second} \end{cases}$$

This can be written as

$$y^{(i)}\left(w \cdot x^{(i)} - b\right) \geq 1, \text{ for all } 1 \leq i \leq n$$

The optimization problem is difficult to solve due to the dependence on $\| w \|$, the norm of w, which a square root is involved. Fortunately by substituting $\| w \|$ with $\frac{1}{2} \| w \|^2$ the equation is altered without changing the solution. Then it will be a quadratic programming optimization problem. More clearly:

$$\arg\min_{(w,b)} \frac{1}{2} \| w \|^2 \text{ for all } 1 \leq i \leq n$$

The Soft Margin method can be applied when the data cannot be clearly categorized. The Soft Margin will choose a hyperplane that splits the data as cleanly as possible, while still maximizing the distance to the nearest cleanly split data. There exits a slack variables, $\xi^{(i)} \geq 0$ that measure the degree of misclassification of the data $x^{(i)}$

$$y^{(i)}\left(w \cdot x^{(i)} - b\right) \geq 1 - \xi^{(i)}, \ 1 \leq i \leq n \qquad (2.2.1)$$

The objective function is then increased by a function which penalizes non-zero $\xi^{(i)}$, and tradeoff a large margin with a small error penalty. If the penalty function is linear, the optimization problem becomes:

$$\arg\min_{w,\xi,b} \left\{ \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \xi^{(i)} \right\}$$

$$\text{For } i = 1, 2, \cdots, n, \qquad \xi^{(i)} \geq 0$$

The constraint in (2.2.1) along with the objective of minimizing$\| w \|$ can be solved through Lagrange multipliers as done above. We has then to solve the following problem, with $\alpha^{(i)}, \beta^{(i)} \geq 0$

$$\arg\min_{w,\xi,b} \max_{\alpha,\beta} \left\{ \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \xi^{(i)} - \alpha^{(i)} \left[y^{(i)} \left(w \cdot x^{(i)} - b \right) - 1 + \xi^{(i)} \right] - \sum_{i=1}^{n} \beta^{(i)} \xi^{(i)} \right\}$$

Through standard quadratic programming techniques and programs the problem can now be solved. The "stationary" Karush–Kuhn–Tucker condition implies that the solution can be expressed as a linear combination of the training vectors

$$w = \sum_{i=1}^{n} \alpha^{(i)} y^{(i)} x^{(i)}$$

The corresponding $x^{(i)}$ are exactly the support vectors, which lie on the margin and satisfy $y^{(i)} \left(w \cdot x^{(i)} - b \right) = 1$.

With the information obtained, the dual form of the SVM reduces to the following optimization problem:

$$\max_{\alpha^{(i)}} L(\alpha) = \sum_{i=1}^{n} \alpha^{(i)} - \frac{1}{2} \sum_{i,j=1}^{n} \alpha^{(i)} \alpha^{(j)} y^{(i)} y^{(j)} k\left(x^{(i)} x^{(j)} \right)$$

For $i = 1, 2, \cdots, n$, $\alpha^{(i)} \geq 0$ and to the constraint from the minimization in b

$$\sum_{i=1}^{n} \alpha^{(i)} y^{(i)} = 0$$

The key advantage of Soft Margin is that the slack variables vanish from the dual problem, with the constant C appearing only as an additional constraint on the Lagrange multipliers.

For non-linear classification, we applies kernel function to transform. V. Vapnik suggested a way to create nonlinear classifiers by applying the kernel trick to maximum-margin hyperplanes in 1992. The resulting algorithm is similar, except a nonlinear kernel function k is applied rather than dot product. This allows the feature mapping $\varphi(x)$ maps into the feature space.

The kernel is related to the transform $\varphi x^{(i)}$ by the equation

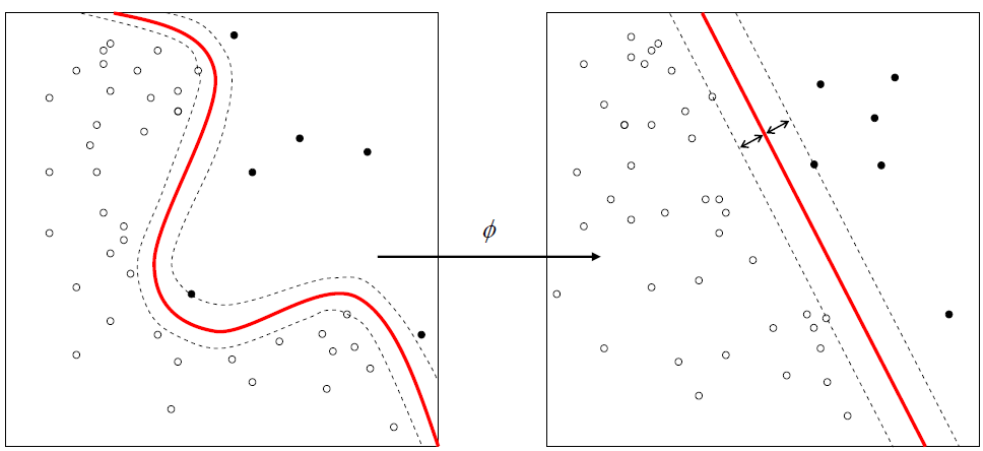
$$k\left(x^{(i)}, x^{(j)}\right) = \varphi x^{(i)} \cdot \varphi x^{(j)}$$

w will also in the transformed space, with

$$w = \sum_{i=1}^{n} \alpha^{(i)} y^{(i)} \varphi x^{(i)}$$

Again, we use the kernel trick instead of dot products with w for classification,

$$\text{i. e.} \qquad w \cdot \varphi(x) = \sum_{i=1}^{n} \alpha^{(i)} y^{(i)} k\left(x^{(i)}, x\right)$$



Kernel machines

If we use a Gaussian radial basis function from kernel, the corresponding feature space is a Hilbert space of infinite dimensions. Maximum margin classifiers are well regularized, so the results do not spoil by the infinite dimensions. There are several common kernels such as a homogeneous polynomial kernel function $k\left(x^{(i)}, x^{(j)}\right) = \left(x^{(i)} \cdot x^{(j)}\right)^d$, a Gaussian radial basis kernel function $k\left(x^{(i)}, x^{(j)}\right) = \exp\left(-\gamma\left\|x^{(i)} - x^{(j)}\right\|^2\right)$, for $\gamma > 0$. Sometimes parameterized using $\gamma = {}^1\!/_{2\sigma^2}$. Finally, we can classify the test examples as

$$y^{(i)} = \begin{cases} 1 & \text{if } w\varphi\left(x^{(i)}\right) - b \geq 1 \\ -1 & \text{otherwise} \end{cases}$$

For $i = 1, 2, \cdots, n$

Chapter 3

Data, Features selection and Measurements

In this chapter, we will first introduce the detail information on data set and features selection. Then we will present the parameters that can help us have a better understanding of the analysis.

3.1 Data selection

The data selected for this analysis are stock listed companies in United States during the period of 2008 to 2012. We will examine the results of the analysis in two subset, the large data subset and small data subset. The large data subset is a combination of 1421 non-defaulted companies from Standard & Poor 1500 and 144 defaulted companies. Therefore the non-defaulted to defaulted ratio is 10:1. Small data subset is a combination of 523 non-defaulted companies from Standard & Poor

500 and 144 defaulted companies. Hence, the non-defaulted to defaulted ratio is about 4:1.

3.2 Features selection

In this study, we will introduce five features for the analysis. Three of them are historical quotes related and two are financial ratios during the period from 2008 to 2012. Furthermore, the study will examine and compare the correlation of each feature and varies combination of features to achieve the best way of forecasting the default likelihood.

FEATURES	DISCRIPTION
X1	Stock performance
X2	Stock performance compare to market performance
X3	Stock performance compare to index/sector performance
X4	Current ratio
X5	Debt to equity ratio

Table 1. Features selected for the analysis

Each feature is constructed by 2 sub-features, which is 1 year prior performance and 2 year prior performance. For example, Franklin Bank defaulted on November 7th, 2008. For feature X1, we will collect the end of year quote from 2005 to 2007

Year	End of year quote
2007	4.31
2006	20.54
2005	17.99

Then the first sub-feature of X1 will be $(4.31 - 20.54) / 20.54 = -0.7902$, and second sub-feature of X1 will be $(4.31 - 17.99) / 17.99 = -0.7604$. Similarly, the relative performance are applied to the rest of the features. Therefore, a total of 10 sub-features are used during the analysis.

3.3 Measurements

In this study, we first set up following the confusion matrix as the basic statistical measurements.

		True classification	
		0 (negative)	1 (positive)
Test classification	0 (negative)	TN	FN
	1 (positive)	FP	TP

In the table above, 0 the 'negative' is representing non-default and 1 the 'positive' represent default. Furthermore, TN stands for *true negative*, which true classification matches test classification of being non-default. FN stands for *false negative*, which true classification is defaulted, but test class predicted to be non-defaulted. FP stands for *false positive*, which true classification is non-defaulted, but test class predicted to be defaulted. TP stands for *true positive*, which true classification matches test classification of being default.

Next, with the confusion matrix elements we will introduce the following statistical measurements in order to have a better understanding of the analysis.

Accuracy = (TN + TP) / (TN + FN + FP + TP)

Recall = TP / (FN + TP)

Specificity = TN / (FP + TN)

Positive predictive value (precision) = TP / (FP + TP)

Negative predictive value = TN / (FN + TN)

F-measure = (Recall × Precision × 2) / (Recall + Precision)

Type 1 error = 1 - Specificity

Type 2 error = 1 - Recall

Accuracy measures the correctness of prediction. *Recall* is the ratio of correctly predicted default companies among all default companies.

Specificity shows the percentage of correctly predicted non-default companies among all non-default companies.

Positive predictive value is the ratio of correctly predicted default companies among all predicted default companies.

Negative predictive value is the ratio of correctly predicted non-default firms among all predicted non-default companies.

F-measure is the Harmonic mean of Recall and Precision.

Type 1 error shows the percentage of falsely predicted the non-default company as a default company.

Then most important of all, *Type 2 error* gives the percentage of falsely predicted defaulted companies as non-default company.

Chapter 4

Results and Analysis

In this chapter, the empirical results and analysis will be presented. First, we will evaluate the results when different features are applied. Then we will divide the data into smaller subsets by the defaulting year and apply it to practical annual forecasting.

4.1 General result and analysis

Features	Accuracy	Recall	Specificity	Positive predictive value
X1-X5	0.8981	0.6667	0.9444	0.7059
X1-X3	0.9352	0.6111	1.0000	1.0000
X1-X2	0.9722	0.9444	0.9778	0.8947
	Negative predictive value	F-measure	Type 1 error	Type 2 error
X1-X5	0.9341	0.6857	0.0556	0.3333
X1-X3	0.9278	0.7586	0.0000	0.3889
X1-X2	0.9888	0.9189	0.0222	0.0556

Table 1. General analysis of logistic regression

Features	Accuracy	Recall	Specificity	Positive predictive value
X1-X5	0.8889	0.8333	0.9000	0.6250
X1-X3	0.9167	0.9444	0.9111	0.6800
X1-X2	0.8889	1.0000	0.8667	0.6000
	Negative predictive value	F-measure	Type 1 error	Type 2 error
X1-X5	0.9643	0.7143	0.1000	0.1667
X1-X3	0.9880	0.7907	0.0889	0.0556
X1-X2	1.0000	0.7500	0.1333	0.0000

Table 2. General analysis of support vector machines

We can see from the tables above that when all the features are included in the analysis, both methods shows an accuracy around 0.89. However, the type 2 error for logistic regression is 0.33, nearly twice as much as 0.17 from support vector machines.

After both financial ratios features are eliminated from the analysis, the accuracy of logistic regression has increased dramatically from 0.90 to 0.94 while the type 2 error also increased from 0.33 to 0.39. In the meantime, the accuracy of support vector machines also increased from 0.89 to 0.92 with the type 2 error cut to only one third amount.

Finally, when only X1 and X2 are included in the feature, the accuracy of logistic regression has reached 0.97 and type 2 error has dropped down to only 0.06. The accuracy of support vector machines also moved back to 0.89 while the type 2 error reached 0.

Overall, both methods had forecasted with higher accuracy and lower type 2 error when only stock performance related features are selected for the analysis.

4.2 Practical Analysis

In this section, in order to examine the reliability of stock performance related features in the practical, we will apply both methods to both large and small data subsets. For example, with the information analysis we have obtained from the 2008 data, we will try to forecast the risk of default in 2009. Then we compare our results with empirical data from 2009 to test the accuracy.

Train	True	Accuracy	Recall	Specificity	Positive predictive value
2008	2009	0.9148	0.9318	0.9125	0.5942
2009	2010	0.9707	0.8095	0.9841	0.8095
2010	2011	0.8049	0.9130	0.7955	0.2800
2011	2012	0.9586	0.6757	0.9934	0.9259
		Negative predictive value	F-measure	Type 1 error	Type 2 error
2008	2009	0.9898	0.7257	0.0875	0.0682
2009	2010	0.9841	0.8095	0.0159	0.1905
2010	2011	0.9906	0.4286	0.2045	0.0870
2011	2012	0.9614	0.7813	0.0066	0.3243

Table 3. Logistic regression analysis for large data subset

Train	True	Accuracy	Recall	Specificity	Positive predictive value
2008	2009	0.8984	0.8636	0.9031	0.5507
2009	2010	0.9121	0.9524	0.9087	0.4651
2010	2011	0.9443	0.9130	0.9470	0.6000
2011	2012	0.9024	0.9730	0.8937	0.5294
		Negative predictive value	F-measure	Type 1 error	Type 2 error
2008	2009	0.9797	0.6726	0.0969	0.1364
2009	2010	0.9957	0.6250	0.0913	0.0476
2010	2011	0.9921	0.7241	0.0530	0.0870
2011	2012	0.9963	0.6857	0.1063	0.0270

Table 4. Support vector machines analysis for large data subset

Train	True	Accuracy	Recall	Specificity	Positive predictive value
2008	2009	0.9266	0.9318	0.9248	0.8039
2009	2010	0.9634	0.8571	1.0000	1.0000
2010	2011	0.9468	0.7826	1.0000	1.0000
2011	2012	0.9315	0.9444	0.9273	0.8095
		Negative predictive value	F-measure	Type 1 error	Type 2 error
2008	2009	0.9762	0.8632	0.0752	0.0682
2009	2010	0.9531	0.9231	0.0000	0.1429
2010	2011	0.9342	0.8780	0.0000	0.2174
2011	2012	0.9808	0.8718	0.0727	0.0556

Table 5. Logistic regression analysis for small data subset

Train	True	Accuracy	Recall	Specificity	Positive predictive value
2008	2009	0.7966	0.9545	0.7444	0.5526
2009	2010	0.9512	0.9524	0.9508	0.8696
2010	2011	0.9574	0.8261	1.0000	1.0000
2011	2012	0.8904	0.9722	0.8636	0.7000
		Negative predictive value	F-measure	Type 1 error	Type 2 error
2008	2009	0.9802	0.7000	0.2556	0.0455
2009	2010	0.9831	0.9091	0.0492	0.0476
2010	2011	0.9467	0.9048	0.0000	0.1739
2011	2012	0.9896	0.8140	0.1364	0.0278

Table 6. Support vector machines analysis for small data subset

Overall, both methods has performed well during this practical experiment. Logistic regression has shown a 0.93 accuracy and 0.15 type 2 error. Support vector machines method has obtained a 0.91 accuracy and 0.07 type 2 error.

During the experiment of large data subset, both methods has shown a 0.91 accuracy. However, the type 2 error from logistic regression is 0.17, more than twice of 0.07, the type 2 error obtained from support vector machines. Hence, when large amount of data are being analyzed, support vector machines method is able to provide lower error while remain the same accuracy.

For small data subset, logistic regression method achieved an average accuracy of 0.94. It has outperformed support vector machines accuracy by 0.04. Also compare to large data subset, logistic regression has lowered the type 2 error to

0.12. On the other hand, support vector machines has shown the stability maintaining the type 2 error at 0.07.

Chapter 5

Conclusion

The ultimate goal of this research is to forecasting the risk of default, which targeting result with higher accuracy and lower type 2 error. The result has shown that support vector machine is able to provide a more stable accuracy regardless the size of data or the quantity of features. On the other hand, although logistic regression shows higher accuracy, the accuracy varies upon the features selection. Support vector machines appears to be more reliable in practical use, since lower type 2 error means less risk of investing in a company that will default but was classified as non-defaulted.

Furthermore, we can see from the accuracy is higher and type 2 error is lower when only stock performance and related comparisons are selected as the features of the analysis. This suggested that the stock performance and risk of default has a very high correlation. In other words, the financial status of companies in US can be best reflected by the stock performance.

Bibliography

[1] C. Cortes and V. Vapnik, *Support-Vector Networks*, Machines Learning, 20,273-297, 1995

[2] C. W. Hsu and C. J. Lin, *A comparison of methods for multiclass support vector machines*, IEEE transactions on neural networks, 13(2), 415-425, 2002

[3] J.H. Min and Y.C. Lee, *Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters*, Expert Systems with Applications, 28, 603-614, 2005

[4] K.S. Shin, T.S. Lee and H. Kim, *An application of support vector machines in bankruptcy prediction model*, Expert System with Applications, 28, 127-135, 2005

[5] A. Lee, *Default Predication by Using Machine Learning Methods*, MTL, SNU, 2012

[6] Wikipedia contributors, "Support vector machine," *Wikipedia, The Free Encyclopedia*, http://en.wikipedia.org/wiki/Support_vector_machine

.