Minimising Expected Misclassification Cost when using Support Vector Machines for Credit Scoring

Terry Harris, Curtis Gittens
Dept. of Computer Science, Mathematics & Physics
University of the West Indies - Cave Hill Campus
Bridgetown, Barbados
terry.harris@mycavehill.uwi.edu, curtis.gittens@cavehill.uwi.edu

***Abstract*— With the gradual relaxation of credit around the world, the cost of losses experienced when extending credit is expected to become increasingly important to financial institutions. In this paper, we offer theoretical and empirical evidence to support the argument that the minimisation of this cost should be the primary objective when developing classification models for credit scoring. This cost can be referred to as the Expected Misclassification Cost. In addition, we present and test a system that builds models to minimise this cost when given varying values for its components. Moreover, we show that using differing values for the components of Expected Misclassification Cost can result in improved performance, in terms of Type I or Type II accuracy, when Expected Misclassification Cost is used as the prime evaluation metric by a support vector machine.**

Keywords- Credit Scoring; Decision Support Systems; Expected Misclassification Cost ; Support Vector Machines

I. INTRODUCTION

The assessment of credit risk is a very important task for financial institutions. This is in part due to the need to avoid losses associated with inappropriate credit approval or rejection decisions [1]. In recent years, credit scoring has emerged as one of the primary ways for financial institutions to assess credit risk [2]. Credit scoring entails the classification of potential customers into applicants with good credit and applicants with bad credit. This is done by analysing the applicant's data based on a past pattern of customer behaviour [3].

Since Fisher's [4] seminal paper, numerous models have been proposed, which attempt to differentiate between "good" and "bad" credit applicants. Many of these classification models are based on classical statistical methods such as Discriminant Analysis (DA), Linear or Polynomial Regression (LPR), Logistic Regression (LR), Non-Parametric Models (NPMs), Artificial Neural Networks (ANNs), and Support Vector Machines (SVMs) [5], [6], [7], [8], [9], and [10].

Whatever its form, many existing credit scoring models are built on samples of customer historical data, and their primary objective is to avoid over-fitting while maximising generalisablity from the samples [5]. As a result, improving test accuracy, as in (1), which is the measure of how accurately the model classifies credit applicants from a withheld dataset, known as the test dataset, is of importance [5] and [6]. However, this approach alone can lead to unsatisfactory results if the cost of making one type of error as opposed to another is not considered. We propose that credit scoring models can be improved if they are designed to minimise this type of cost called the Expected Misclassification Cost [11].

$$\text{Test Accuracy} = \frac{\text{True Positive}}{\text{True Positive+False Positive}} + \frac{\text{True Negative}}{\text{False Negative+True Negative}} \quad (1)$$

The remainder of this paper is organised as follows. In Section II, we discuss some of the problems which emerge when using test accuracy as the primary model evaluation metric, before discussing the rationale behind the use of the Expected Misclassification Cost as the model evaluation metric. In Section III, the Support Vector Machine algorithm, which is the classification algorithm implemented in our system, is discussed. The details of the dataset chosen as our case study are presented in Section IV. Described in Section V, is our parameter tuning algorithm and the methodology of the study. Section VI, discusses the results of the study, and Section VII highlights the conclusions and directions for future research.

II. BACKGROUND

A. Skewed Datasets

When a classification model is designed to minimise test accuracy as its main objective, this can prove problematic if the training dataset is skewed in favour of one particular class over another (as is often the case in credit scoring exercises). This is because it becomes difficult to determine if higher test accuracy corresponds to an improved quality classifier. The following example illustrates this point.

Suppose we have a classifier that gives a test accuracy of 99% when determining the creditworthiness of clients. At first glance, this system seems to be a good classification model. However, if the probability of a potential customer being un-creditworthy is 0.5%, it becomes clear that test accuracy tells us nothing about the quality of the classifier because 99.5 % test accuracy can be achieved by classifying

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-202-8

all applicants as creditworthy. Without a doubt, this second approach is unacceptable, because by simply approving all applicants, we are not detecting potentially “bad” clients.

To solve this problem, many researchers often use the Precision, as in (2), and Recall, as in (3), evaluation metrics. Precision is the measure of how accurately we have classified our positive predictions (what fraction is correctly categorised), while Recall measures the proportion of the dataset, which was actually positive, that we predicted as positive. Given our previous scenario, the algorithm that simply predicts that the applicant was creditworthy 100% of the time would continue to score 99.5% on test accuracy; however, it would score 0% accuracy on the Recall evaluation metric. As a result, tailoring classification models to improve Precision and/or Recall can help to improve classifier quality when the dataset is skewed.

$$\text{Precision} = \frac{\text{True Positive}}{\text{\# Predicted as Positive}} = \frac{\text{True Positive}}{\text{True positive+False Positive}} \quad (2)$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{\# Actually Positive}} = \frac{\text{True Positive}}{\text{True Positive+False Negative}} \quad (3)$$

B. *Minimising Type I and Type II errors*

Another issue that arises when using total test accuracy as the performance metric to develop credit scoring models, is the problem of minimising Type I error, as in (4), and Type II error, as in (5). If we let the null hypothesis on any credit approval decision be that the credit applicant is un-creditworthy, then a Type I error occurs when we reject the null hypothesis that the potential customer is un-creditworthy and grants them credit when we should have rejected their application. Conversely, a Type II error occurs when we accept the null hypothesis (that the applicant is un-creditworthy) when we should have rejected it, and grant the client credit. Developing a model to maximise Precision and Recall using the F_1 Score, as in (6), which is a type of average for Precision and Recall, can assist with minimising both of these errors. Furthermore, models could be developed to minimise Type I and Type II errors separately and/or jointly. However, focusing solely on effectively minimising Type I and II errors or maximising Precision and Recall does not take into consideration the misclassification cost to the institution of making one type of error over another [11]. We believe that existing credit scoring models could be enhanced if this expected cost is taken into consideration when developing the model.

$$\text{Type I Error} = \frac{\text{False Positive}}{\text{True Negative+False Positive}} \quad (4)$$

$$\text{Type II Error} = \frac{\text{False Negative}}{\text{True Positive+False Negative}} \quad (5)$$

$$F_1\, Score = 2\frac{Recall*Precision}{Recall+Precision} \quad (6)$$

The Expected Misclassification Cost, as in (7), is comprised of two component costs associated with each type of inappropriate credit granting decision or error. Where the variable Z, represents the Expected Misclassification Cost, X the Default Cost, Y the Opportunity Cost, the variable a, the probability of Type I error, and b the probability of Type II error.

$$Z = Xa + Yb \quad (7)$$

The Expected Default Cost is associated with making Type I errors. This type of error can have the most damaging effect on the institution as it often leads to the loss of credit principal and interest. This cost can be quantified as the net present value of the credit principal and interest (base rate plus margin*principal), multiplied by the probability of Type I error. The second error, Type II is associated with the Expected Opportunity Cost of rejecting a potential client who would have been creditworthy. As a result, this cost is simply the net present value of the interest (net interest rate spread*principal) that could have been made, had credit been granted, multiplied by the probability of Type II error.

C. *Motivation*

Intuitively, for credit-granting decisions, Type I errors should be weighted with higher importance than Type II errors [10]. This belief is due to the fact that when a financial institution grants credit to a customer who later defaults, the financial institution potentially loses 100% of the principal and interest on the investment. This is often a higher cost than the opportunity cost of making a Type II error, which is usually limited to the loss of interest on the investment. However, to seek to minimise Type I error while ignoring its impact on Type II error (as they are inversely related) could lead to increased Expected Misclassification Cost to the institution. This can be seen by the following simplified example.

Suppose an institution seeks to minimise Type I error while ignoring its impact on Type II. One way of achieving this would be to simply cease granting credit. However, if this was done, then the institution would face massive opportunity costs because it would not be earning interest. This means that there must be some optimal value for both Type I and Type II errors such that Expected Misclassification Cost to the institution is minimised.

We present a system that produces credit scoring models which classify credit applicants as either creditworthy or un-creditworthy, such that the Expected Misclassification Cost to a financial institution is minimised. In addition, we present a parameter tuning algorithm which selects the parameters *Gamma* and *C* for the SVM (RBF kernel) such that Type I and/or Type II errors are optimised when weighted according to default cost and opportunity cost. We verify our results by testing our system using the LIBSVM (RBF kernel), which is a state of the art SVM by Chang and Lin [12].

III. SVMs and Credit scoring

The SVM was first developed by Cortes and Vapnik [13] for binary classification. To do this binary classification, SVMs attempt to find the optimal separating hyperplane between classes by maximising the margin (Fig. 1). The points lying on the boundaries are called support vectors, and the middle of the margin is referred to as the optimal separating hyperplane. This margin maximisation characteristic of SVMs is argued to improve the decision boundaries and hence lead to better classifier quality.

A. *SVM use in Credit Scoring*

Over the past decade, SVMs have been successfully used in many credit scoring systems [14], [15], [16], [17], and [18]. However, the superiority of the SVM when compared to other classifiers remains debatable, as Van Gestel et al. [16] found that even though SVMs showed improved performance, there was no significant difference between SVMs, LR and LDAs. This finding supports a widely held view that modern learning algorithms approximate each other's performance when given large datasets [19]. Consequently, although we use SVMs to implement our credit scoring system we suspect that other classification techniques may approximate or even outperform our system once designed to minimise Expected Misclassification Cost.

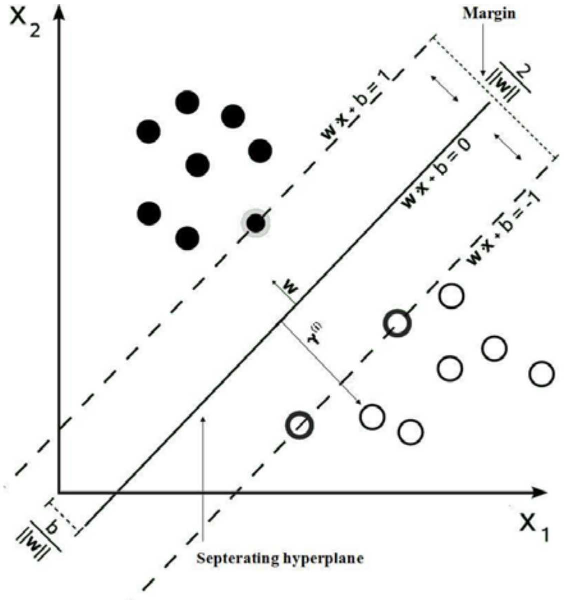

Figure 1: Simplified Depiction of SVM Classification

B. *SVMs Development for Credit Scoring*

When a financial institution is presented with a new credit applicant, in order to make the credit approval decision the institution seeks to classify the applicant as either "good" or "bad" according to the SVM score. In the case of a linear SVM this score can be represented as the linear combination of the applicant's characteristics (features) multiplied by some weights, as in (8).

$$z = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b \qquad (8)$$

Where *n* represents the number of client features, the *w*'s and *b* are learnt parameters, and the *x*'s are client features. Transforming the *w*'s and *x*'s into column vectors, (8) can be written more concisely as;

$$z = w^T x + b. \qquad (9)$$

The SVM learns the parameters *w* and *b* from training examples of historic client data that the financial institution collected over time. This training dataset will normally consist of a number of example clients; as a result, from a geometric perspective, calculating the value of *w* and *b* means looking for a hyperplane which best separates "good" clients from "bad". To do this, the SVM maximises the margin between the two clouds of data. As a result, when given a training example ($x^{(i)}$, $y^{(i)}$), such that $y \in$ {-1,1}, the functional margin $\hat{\gamma}$, of (*w*, *b*) can be defined with respect to the training example as;

$$\hat{\gamma} = y^{(i)}(w^T x + b). \qquad (10)$$

In order to confidently predict the class of the training example the functional margin needs to be large. Thus, if $y^{(i)}$ = 1, then for the functional margin to be large $w^T x + b$ must be a large positive number. As a result, if $y^{(i)}$ = -1, then $w^T x + b$ needs to be a large negative number. Accordingly, given a training set $S = \{(x^{(i)}, y^{(i)}); i = 1,\dots, m\}$, the function margin of (*w*, *b*) with respect to *S* is defined as the smallest of the functional margins of the training examples, as in (11).

$$\hat{\gamma} = \min_{i=1,\dots,m} \hat{\gamma}^{(i)} \qquad (11)$$

To find the geometric margin, γ, consider the case of a positive training example where $x^{(i)}$ corresponds to the label $y^{(i)}$ = 1. The distance from this point to the decision boundary, $\gamma^{(i)}$, is a straight line (vector) orthogonal to the hyperplane (Fig. 1). To find the value of $\gamma^{(i)}$ the corresponding point on the decision boundary is found. This can be easily determined since *w*/||*w*|| is a unit-length vector pointing in the same direction as *w*. Therefore, the corresponding point on the hyperplane is given by the equation $x^{(i)} - \gamma^{(i)} \cdot w/||w||$, and because this point lies on the decision boundary, it satisfies the equation $w^T x + b = 0$ (Fig. 1), as in (12).

$$w^T\left(x^{(i)} - \gamma^{(i)} \frac{w}{\|w\|}\right) + b = 0 \qquad (12)$$

We can simplify (12) as following:

$$w^T x^{(i)} - \gamma^{(i)} \frac{w^T w}{\|w\|} + b = 0\,. \qquad (13)$$

Since, $w^T w/\|w\| = \|w\|^2/\|w\| = \|w\|$, we solve for $\gamma^{(i)}$, as is shown in (14);

$$\gamma^{(i)} = \left(\frac{w}{\|w\|}\right)^T x^{(i)} + \frac{b}{\|w\|}. \qquad (14)$$

Generalising this representation to account for negative training examples, we have;

$$\gamma^{(i)} = y^{(i)} \left[\left(\frac{w}{\|w\|}\right)^T x^{(i)} + \frac{b}{\|w\|}\right]. \qquad (15)$$

Here, if $\|w\| = 1$, then the geometric margin is equal to the functional margin, In addition, the geometric margin is invariant to rescaling of the parameters (w, b). As a result, given a training set $S = \{(x^{(i)}, \; y^{(i)}),\; i = 1,\ldots,m\}$, the geometric margin is the smallest of the geometric margins on the individual training examples (16).

$$\gamma = \min_{i=1,\ldots,m} \gamma^{(i)} \qquad (16)$$

Accordingly, when given a training dataset of past clients, it seems natural that the financial institution would want to find a decision boundary that maximises the geometric margin, since this would reflect a very confident set of predictions on the training data. Specifically, this will result in a SVM classifier that separates “good” and “bad” past clients effectively, thus giving the institution reliable information with which to make judgments about future credit applications. As a result, to find the hyperplane that achieves the maximum geometric margin the following optimisation problem is posed:

$$\max_{\gamma,w,b} \gamma,$$

$$s.t.\; y^{(i)}\left(w^T x^{(i)} + b\right) \geq \gamma, i = 1, \ldots, m, \qquad (17)$$

$$\| w \| = 1.$$

However, because the $\|w\| = 1$ constraint is non-convex, the problem is transformed into one more suited for optimisation, as in (18). Here, if, $\hat{\gamma} = 1$, then $\hat{\gamma}/\|w\| = 1/\|w\|$, and maximising this is the same thing as minimising $\|w\|^2$.

$$\min_{\gamma,w,b} \frac{1}{2} \| w \|^2,$$

$$s.t.\; y^{(i)}\left(w^T x^{(i)} + b\right) \geq \hat{\gamma}, i = 1, \ldots, m. \qquad (18)$$

At this point, a regularisation term ξ, is added to the optimisation problem posed in (18) to modify the algorithm so that it works for non-linearly separable datasets, as is often the case with credit scoring data. The term C is a turning parameter which weights the significance of a classification error to the overall model.

$$\min_{\gamma,w,b} \frac{1}{2} \| w \|^2 + C \sum_{i=1}^m \xi_i,$$

$$s.t.\; y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 - \xi_i, i = 1, \ldots, m, \qquad (19)$$

$$\xi_i \geq 0, i = 1, \ldots, m.$$

Equation (19) represents the primal from of the optimisation problem for finding the optimal margin classifier to separate “good” and “bad” clients. Given that this equation satisfies the Karush-Kuhn-Tucker (KKT) conditions, the condition $g_i(w) \leq 0$ is an active constraint. As a result, the constant to the primal problem can be rewritten as follows:

$$g_i(w) = -y^{(i)}\left(w^T x^{(i)} + b\right) + 1 - \xi_i \leq 0. \qquad (20)$$

To develop the dual form of the problem, the Lagrangian for the optimisation problem is constructed, as in (21). Where the α_i ’s and the r_i’s are Lagrangian multipliers.

$$L(w,b,\xi,\alpha,r) \frac{1}{2} \| w \|^2 - c \sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i \left[y^{(i)}\left(w^T x^{(i)} + b\right) - 1 + \xi_i\right] - \sum_{i=1}^m \mathrm{r}_i \, \xi_i \qquad (21)$$

Equation (21) is minimised with respect to w and b by taking partial derivatives with respect to w and b and setting them to zero. The equations derived are as follows:

$$\frac{\partial}{\partial w} L(w,b,\xi,\alpha,r) = w - \sum_{i=1}^m \alpha_i y^{(i)} x^{(i)} = 0, \qquad (22)$$

$$\frac{\partial}{\partial b} L(w,b,\xi,\alpha,\mathrm{r}) = \sum_{i=1}^m \alpha_i y^{(i)} = 0. \qquad (23)$$

Solving (22) for w produces;

$$w = \sum_{i=1}^m \alpha_i y^{(i)} x^{(i)}. \qquad (24)$$

Therefore, substituting the definitions of w (24) and b (23) in (21) and including the constraints $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^m \alpha_i y^{(i)} = 0$ the dual optimisation problem is derived as;

$$W(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y^{(i)} y^{(j)} \alpha_i \alpha_j < x^{(i)}, x^{(j)} >,$$

$$s.t.\; 0 \leq \alpha_i \leq C, i = 1, \ldots, m, \qquad (25)$$

$$\sum_{i=1}^m \alpha_i y^{(i)} = 0.$$

This dual form (25) can be solved in lieu of the primal problem, in order to derive the parameters α_i's that maximise $W(\alpha)$ subject to the constraints. These parameters can then

ISBN: 978-1-61208-202-8

be used in (24) to find the optimal *w*'s. Having found *w**, the primal problem can be used to find the optimal value for the intercept term *b*.

Accordingly, after the classification model has been trained, when presented with a new credit applicant the equation $w^T x + b$, would calculate and predict $y = 1$ if and only if this quantity is bigger than zero.

$$(w^T x + b) = \left(\sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}\right)^T x + b \qquad (26)$$

Equation (26) can be rewritten as;

$$\sum_{i=1}^{m} \alpha_i y^{(i)} < x^{(i)}, x > +b. \qquad (27)$$

This representation allows for the inclusions of kernels to deal more effectively with datasets which have multiple dimensions. Kernels map attributes to higher order feature spaces, and this is represented by replacing the *x*'s in the equation with the feature vector ϕ(*x*), as shown in (28).

$$\sum_{i=1}^{m} \alpha_i y^{(i)} K(x^{(i)}, x) + b \qquad (28)$$

Where,

$$K\left(x^{(i)}, x^{(j)}\right) =< \phi(x^{(i)}), \phi(x^{(j)}) > . \qquad (29)$$

IV. DATA

A German credit scoring dataset was taken from the UCI Machine Learning Repository [20]. This dataset was provided by Prof. Hofmann of Hamburg University and consists of 700 examples of creditworthy applicants and 300 un-creditworthy applicants. This dataset has been widely used in credit scoring research to evaluate the performance of classification models. The dataset measured twenty (20) features for each credit applicant comprising the following categories: the status of the client's existing checking account, the duration of the credit period in months, the client's credit history, the purpose for the credit, the credit amount requested, the client's savings account/bonds balance, the client's present employment status, the client's personal (marital) status and sex, whether the client is a debtor or guarantor of credit granted by another institution, the number of years spent at present residence, the type of property possessed by the client, the client's age in years, whether the client has other installment plans, the client's housing arrangements (whether they own their home, rent, or live for free), the number of existing credits the client has at the bank, the client's job, the number of people for whom the client is liable to provide maintenance for, whether the client has a telephone, and whether the client is a foreign worker.

The data was pre-processed so as to transform all categorical data into numerical data for analysis. In addition, the data was normalised so as to improve the performance of the SVM.

V. ALGORITHM AND METHODOLOGY

A. *Parameter Tuning Algorithm*

```
Begin
    1.  Randomly sort sample applicant dataset.
    2.  Split sample dataset into 3 sub datasets.
        a.  Sub-dataset 1: Training (60%)
        b.  Sub-dataset 2: Cross Validation (20%)
        c.  Sub-dataset 3: Test (20%)
    3.  For the # of parameters conduct grid-search
            Select the pair of parameters (C and
            Gamma) based on how well they minimise
            expected misclassification cost on the
            Training dataset using the CV dataset.
        End for
    4.  Use the pair of parameters from part 3 to train the
        model using Training dataset.
    5.  Test the model for overall Test, Type I, and Type II
        accuracies using the Test dataset (reported results).
    6.  Re-train the model using the full dataset and the
        pair of parameters selected in part 3.
End
```

B. *Method*

Our empirical testing began by randomly sorting the dataset before splitting it into 3 sub-datasets; the training dataset, the cross validation dataset, and the test dataset. The initial step of randomly sorting the dataset was done in order to increase the probability of an equal distribution of clients across the 3 sub-dataset. To train for the minimisation of the components of the Expected Misclassification, we further subdivided the cross validation dataset into two data-files, each only containing positive or negative examples. To test for Type I and Type II accuracy the test dataset was also subdivided into two data-files, one with all positive and another with all the negative test examples.

We implemented our system in OCTAVE 3.2.4 and used it to repeatedly train models using the LIBSVM package fitted with a RBF Kernel. These models where built using the training dataset and certain values for the parameters *Gamma* and *C*. We used a grid search technique to find the parameters *Gamma* and *C* which minimised Expected Misclassification Cost using the cross validation dataset. When deciding on the search ranges for *C* and *Gamma* care was taken to ensure that ∃ *C* and ∃ *Gamma,* within the search ranges, which produced models that have zero Type I error, and zero Type II error (on two separate models). This was an important step to ensure that each component of Expected Misclassification Cost could be minimised to zero. The usual approach when selecting the parameter ranges is to use known benchmarks. However, these ranges may not be well-suited to every dataset and do not guarantee perfect Type I or Type II accuracy on any of the possible models.

Having found the pair of parameters which minimised the Expected Misclassification Cost on the cross validation dataset, we used them to build our models. Three models were built using varying assumptions for Default Cost and Opportunity Cost. This was done in order to illustrate the dynamic nature of our system. The results are presented in TABLE I.

VI. Results and Analysis

The first model shown in the TABLE I was built weighting Default Cost and Opportunity Cost equally. As a result, the minimisation of Expected Misclassification Cost equated to the minimisation of overall test accuracy. We use this model as a control to illustrate the variations in performance achievable if different weights are used when setting Default Cost and Opportunity Cost. This first model surpassed most contemporary classifiers in terms of Type I accuracy on this dataset (TABLE II). This performance is interesting because many existing SVM classifiers that have reported results on this dataset were highly optimised for performance while our system is not. The reason for our relatively superior performance could be attributed to the fact that we selected the parameter ranges to ensure that errors on both Type I and Type II error metrics could be minimised as low as possible. However, further investigation into this hypothesis needs to be conducted to confirm our intuition.

TABLE I. MODELS AND ACCURACIES

Model	Parameters		Accuracies (%)				
	Gamma	*C*	*Train*	*CV*	*Type I*	*Type II*	*Test*
1	2^{-50}	2^{57}	74.83	73.13	66.66	73.24	71.36
2	2^{-50}	2^{49}	71.16	69.84	75.45	66.20	68.84
3	2^{-50}	2^{41}	76.66	76.12	40.35	90.14	75.88

TABLE II. PERFORMANCE COMPARISONS

Models	Accuracies (%)		
	Type I Accuracy	*Type II Accuracy*	*Total Accuracy*
Model 1	66.66	73.24	71.36
Model 2	75.45	66.20	68.84
Model 3	40.35	90.14	75.88
Yu et al. [10]	53.57	90.33	78.46
Wang et al. [15]	45.62	89.44	76.30
Ahmad et al .[21]	66.66	88.08	81.42

The second model (TABLE I) was built with the objective of reducing Expected Default Cost (weighted Type I error), while placing less emphasis on Expected Opportunity Cost (weighted Type II error). To achieve this, Default Cost was set to one while Opportunity Cost was set to one-half. As a result, when the system selected parameters to minimise Expected Misclassification Cost, the Expected Default Cost was weighted twice as significant as the Expected Opportunity Cost. This process successfully achieved better performance (75.45%). As shown in TABLE II, this result surpassed the performance in terms of Type I accuracy of many of the known published SVM systems on this dataset, while still remaining relatively generalisable at 68.84% test accuracy. We attribute this performance to the fact that when given the input values for Default Cost and Opportunity Cost our system selected parameters for the model which placed more emphasis on the reduction of Expected Default Cost which is calculated based on Type I error. Focus was placed on Expected Default Cost because it was the primary contributor to Expected Misclassification Cost in this model.

The third model presented in TABLE I was built with the intention of reducing Expected Opportunity Cost (weighted Type II error), while weighting the impact of Expected Default Cost (weighted Type I error) with less importance. To achieve this, Default Cost was set to one-half, while Opportunity Cost was set to one. As a result, this model showed a 16.9% improvement in terms of Type II accuracy when compared to the control (Model 1). In addition, this model showed an improvement of 4.52% over the Model 1 in terms of test accuracy (75.88%). However, this model resulted in a 26.31% fall in terms of Type I accuracy. We attributed this occurrence to the fact that the model is weighted to select those parameters for *C* and *Gamma* which minmise the Expected Opportunity Cost since it had a greater impact on Expected Misclassification Cost in this model.

VII. Conclusion and Future work

In this paper, we presented a system for the minimisation of the expected cost to financial institutions when making credit granting decisions. We showed that the minimisation of this cost, which is referred to as the Expected Misclassification Cost, can be achieved by considering its components when building classifier models. In addition, we showed that this approach can lead to performance gains by increasing Type I and Type II accuracy.

Future work will consider the generalisablity of this approach to other classifiers and classification problems. In addition, other studies will investigate the advantages and disadvantages of using Expected Misclassification Cost as the primary model evaluation metric in combination with ensembles, bagging, boosting and other SVM performance enhancing techniques.

Acknowledgment

This research was partially funded by a Graduate Scholarship from the University of the West Indies, Cave Hill Campus. In addition, we would like to acknowledge fellow members of our research group (Sequitur), Mr. Gamal Crichton and Mr. Xavier Caddle, for their valuable comments and suggestions. Finally, we would like to thank the anonymous reviews, who helped to improve the readability of this paper.

REFERENCES

[1] L. Yu, S. Wang, and K. K. Lai, "Credit risk assessment with a multistage neural network ensemble learning approach," *Expert Systems with Applications*, vol. 34, pp. 1434-1444, 2008.

[2] C. L. Huang, M. C. Chen, and C. J. Wang, "Credit scoring with a data mining approach based on support vector machines," *Expert Systems with Applications*, vol. 33, pp. 847-856, 2007.

[3] L. Thomas, R. Oliver, and D. Hand, "A survey of the issues in consumer credit modelling research," *Journal of the Operational Research Society*, vol. 56, pp. 1006-1015, 2005.

[4] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Human Genetics*, vol. 7, pp. 179-188, 1936.

[5] Z. Huang, H. Chen, C. J. Hsu, W. H. Chen, and S. Wu, "Credit rating analysis with support vector machines and neural networks: a market comparative study," *Decision support systems*, vol. 37, pp. 543-558, 2004.

[6] Y. Wang, S. Wang, and K. Lai, "A new fuzzy support vector machine to evaluate credit risk," *Fuzzy Systems, IEEE Transactions on*, vol. 13, pp. 820-831, 2005.

[7] H. Li and J. Sun, "Predicting business failure using multiple case-based reasoning combined with support vector machine," *Expert Systems with Applications*, vol. 36, pp. 10085-10096, 2009.

[8] L. Yu, S. Wang, and J. Cao, "A modified least squares support vector machine classifier with application to credit risk analysis," *International Journal of Information Technology & Decision Making*, vol. 8, pp. 697-710, 2009.

[9] L. Zhou, K. K. Lai, and L. Yu, "Least squares support vector machines ensemble models for credit scoring," *Expert Systems with Applications*, vol. 37, pp. 127-133, 2010.

[10] L. Yu, X. Yao, S. Wang, and K. Lai, "Credit Risk Evaluation Using a Weighted Least Squares SVM Classifier with Design of Experiment for Parameter Selection," *Expert Systems with Applications*, pp. 15392-15399, 2011.

[11] D. J. Hand and W. E. Henley, "Statistical classification methods in consumer credit scoring: a review," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 160, pp. 523-541, 1997.

[12] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, p. 27, 2011.

[13] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273-297, 1995.

[14] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen, "Benchmarking state-of-the-art classification algorithms for credit scoring," *Journal of the Operational Research Society*, vol. 54, pp. 1082-1088, 2003.

[15] G. Wang, J. Hao, J. Ma, and H. Jiang, "A comparative assessment of ensemble learning for credit scoring," *Expert Systems with Applications*, vol. 38, pp. 223-230, 2011.

[16] T. Van Gestel, B. Baesens, J. A. K. Suykens, D. Van den Poel, D. E. Baestaens, and M. Willekens, "Bayesian kernel based classification for financial distress detection," *European journal of operational research*, vol. 172, pp. 979-1003, 2006.

[17] Y. C. Lee, "Application of support vector machines to corporate credit rating prediction," *Expert Systems with Applications*, vol. 33, pp. 67-74, 2007.

[18] T. Bellotti and J. Crook, "Support vector machines for credit scoring and discovery of significant features," *Expert Systems with Applications*, vol. 36, pp. 3302-3308, 2009.

[19] E. Brill, J. Lin, M. Banko, S. Dumais, and A. Ng, "Data-intensive question answering," 2001.

[20] A. Frank and A. Asuncion. UCI Machine Learning Repository [Online]. Available: http://archive.ics.uci.edu/ml [retrieved: January 21, 2012]

[21] G. Ahmad, R. Manoj, P. Dhirendra, S. Ugrasen, G. Neelesh, G. Roopam, K. Verma, K. K. Brajesh, S. Raghuvir, and S. Pushpa, "A Hybrid Support Vector Machine Ensemble Model for Credit Scoring," 2011.

ISBN: 978-1-61208-202-8