UNIVERSITY OF GOTHENBURG
SCHOOL OF BUSINESS, ECONOMICS AND LAW

Modelling rare events using non-parametric machine learning classifiers

—Under what circumstances are support vector machines preferable to conventional parametric classifiers?

by

Lukas Ma[*]

Bachelor's Thesis in Statistics (15 credits ECTS)

Department of Economics,
School of Business, Economics and Law,
University of Gothenburg

February 2021
Supervisor: Mattias Sundén

[*] Email: lukas.ma96@yahoo.com

Acknowledgements

I want to express my sincere thanks to my supervisor, Mattias Sundén, who guided me through the entire writing phase with continued support and patience. He has provided me with invaluable assistance in constructing the simulation study and offered insightful advice on my work. My gratitude is also extended to my fellow students, Tim Emanuelsson, Sofia Hartelius, Omid Raisi, and Yohanes Wolde-Senbet, for their constructive feedback on my thesis.

Abstract

Rare event modelling is an important topic in quantitative social science research. However, despite the fact that traditional classifiers based upon general linear models (GLM) might lead to biased results, little attention in the social science community is devoted to methodological studies aimed at alleviating such bias, even fewer of them have considered the use of machine learning methods to tackle analytical problems imposed by rare events.

In this thesis, I compared the classification performance of the SVMs – a group of machine learning classification algorithms – with that of the GLMs under the presence of imbalanced classes and rare events. The results of this study shows that the standard SVMs have no better classification performance than the traditional GLMs. In addition, the standard SVMs also tend to have low sensitivity, rendering it inappropriate for rare event modelling. Although the cost-sensitive SVMs could lead to more rare events be identified, these methods tend to suffer from overfitting as the events become rarer. Finally, the results of the empirical analysis using the *Military Interstate Dispute* (MID) data imply that the probabilistic outputs produced by Platt scaling are not reliable. For the above reasons, a wider application of SVMs in rare event modelling is not supported by the results of this study.

Abbreviations

AUC	Area under ROC curve
BA	Balanced accuracy
cdf	Cumulative distribution function
DEC	Different error costs
KNN	k-nearest neighbours
LR	Logistic regression
MID	Military interstate dispute
ML	Maximum likelihood
MLE	Maximum likelihood estimators
MSE	Mean square errors
pdf	Probability density function
pmf	Probability mass function
ROC	Receiver operating characteristic
RVM	Relevance vector machines
SMOTE	Synthetic minority oversampling technique
SVM	Support vector machines

Contents

Chapter 1

Introduction

There are many types of events in our world that, despite being very rare in terms of occurrence, could induce a substantial impact on human society once they actually take place. Examples of such events can be wars, pandemics, financial crises, economic depressions, etc. Statisticians often use the term *rare events*[1] to describe such highly improbable outcomes, which usually appear in data as a binary outcome variable characterised by an overwhelming number of zeros representing the non-events, and a tiny fraction of ones representing the events [King & Zeng, 2001a].

In various social science disciplines, rare events are considered as a topic of great importance because of its potential impact on our society, and hence are extensively studied. For instance, researchers in business administration and innovation studies might be interested in building a model to predict technical breakthroughs *ex ante* among millions of registered patents [Hain & Jurowetzki, 2020], while scholars in international conflict studies might want to construct a predictive model for military conflicts [King & Zeng, 2001a]. However, the quantitative modelling of rare events is a challenging task. As noted by [King & Zeng, 2001a, 2001b], the use of logistic regression (LR), a method commonly used by quantitative social science researchers, might systematically underestimate the probability of rare event occurrence, yielding bias results, also known as *rare event bias* (more on this issue in Section 3.1 below). Although the correction proposed by King & Zeng [2001a, 2001b] could alleviate the rare event bias to some extent, the need for alternative statistical procedures specifically targeted at rare event modelling is still paramount.

Following from the technological development in computer science, non-parametric statistical procedures based upon machine learning algorithms emerged and now receive increasing attention among researchers outside the computer science community. Comparing with the traditional classification approaches, machine learning classifiers are more capable of handling complex data and demonstrate better predictive accuracy in general [Hain & Jurowetzki, 2020]. This argues strongly in favour of a wider application of machine learning methods in social science research, since quantitative research in this field often involve the modelling of complex real-world relationships [Hain & Jurowetzki, 2020]. Additionally, the improvement in predictive accuracy provided by machine learning methods also makes these methods attractive candidates for the rare event modelling. However, as noted by James et al. [2013] and Hain & Jurowetzki [2020], results obtained from non-parametric machine learning methods are often hard to interpret, some of these methods, such as support vector machines (SVM), do not even produce probabilistic outputs. This highlights the need for more research concerning the application of machine learning methods in quantitative social science studies. My thesis is intended to contribute to this kind of research.

[1] Or *black swan events*, a term used in Taleb's [2007] bestseller *The Black Swan: The Impact of the Highly Improbable.*

1.1 Aims and objectives

The aim of this study is to explore the possibility of a wider application of non-parametric machine learning classification algorithms in quantitative social science research, especially research in those fields where rare events are common, such as economics, political science, and conflict studies. This can be done by conducting a comparative study between one such machine learning classifier and classifiers that are commonly used in quantitative social science research. In this thesis, I chose to compare different variants of SVM with methods from the general linear model (GLM) family such as logistic regression (or GLM with logit link). This comparative study was performed in two parts: a simulation study and an empirical analysis. The objective of the simulation study is to evaluate the classification performance of different methods, trying to determine in which type of data the use of a particular classification method is preferred, while the empirical analysis has the objective to extend the results from the simulation study to the context of social science, paving the way for a discussion about the applicability of different methods in social science research concerning rare events.

Therefore, this thesis seeks to answer the following research question:

1. *Under the presence of rare events, do SVMs generally outperform the conventional parametric classifiers in terms of classification performance?*

In addition, in quantitative social science research, we are also interested in constructing a probability model for rare event prediction [King & Zeng, 2001b]. This means that we not only should devote more attention to out-of-sample accuracy than to in-sample accuracy [James et al., 2013], but also focus on whether the classifier is able to produce reliable probability estimates, which can then be used in rare event prediction. Since the outputs from SVM is are not probabilistic and must be converted to probabilities using calibration techniques such as Platt scaling (more on this issue in Section 2.3.4 below), it is also the objective of this thesis to investigate whether the use of such probability output can be motivated. Therefore, this thesis also seeks to address the following question:

2. *Can we motivate the use of the probabilistic outputs produced by Platt scaling in rare event modelling?*

1.2 Related work

To date, there already exist a considerable amount of studies in different levels dedicated to the modelling of class imbalance and rare events. According to Fernández et al. [2018], it is well documented in statistical literature that the presence of class imbalance and rare events would hamper the performance of nearly all classifiers in their standard form, including those that based upon machine learning. This is because all standard classifiers are designed to maximise the overall accuracy. In such setting, the classification accuracy of the majority class is prioritised at the cost of the accuracy of the minority class. Special techniques – such as modifications in the standard classifiers – are required for rare event modelling [Fernández et al., 2018].

For classifiers based upon the GLM, such as logistic and probit regression, the common modification approach is to either adjust the regression output *ex post* or to apply another link function. An example for the former case is the *ReLogit* proposed by King & Zeng [2001a, 2001b]. Their strategy for alleviating rare event bias was to first estimate the bias before running logistic regression, and then subtract this estimated bias from the intercept of the regression output. As for the link function approach, Van der Paal

[2014] had compared the ability of different link functions in modelling rare events. Using four real-world data sets from the *UCI Machine Learning Repository*, Van der Paal [2014] provided empirical evidence that GLMs with skew link function tend to perform better than GLMs with symmetric link functions. Additionally, he also compared the classification performance of the GLMs with two machine learning classifiers – Random forest and SVM – and found that both machine learning classifiers outperformed the GLMs.

In quantitative social science research, comparisons between the GLMs and machine learning methods of the kind mentioned above are, to my knowledge, very limited – not to mention that existing studies on this topic focus predominantly on the use of neural networks. For instance, Zeng [1999] compared the classification performance of two GLM models (logit and probit) with that of a ten-hidden-unit neural network, using both synthetic data generated from a non-linear model with various noise levels and empirical data from previous research in international relations. His conclusion is that the neural network model outperforms the GLMs. Beck et al. [2000] did a similar comparison using the *Militarised Interstate Dispute* (MID) data, which has a structure typical for a rare event data set. Using the militarised conflicts in 1947–85 as the training set and conflicts in 1986–89 as the test set, Beck et al. [2000] concluded not only that the neural network model outperforms the conventional logistic regression model, but also achieved 16.7 percent accuracy in predicting militarised conflicts and 99.42 percent accuracy in non-conflicts in the test set. Finally, in a recent working paper, [Hain & Jurowetzki, 2020] explored the use of autoencoder, a type of neural network, in detecting breakthrough patents. Among the 2,722 breakthrough patents in their test set (corresponding to a rarity equal to 0.6 percent), the trained autoencoder correctly classified 1,402 of them. However, this comes with the cost of large number of false positives, yielding a precision score equal to 0.0328.

1.3 Scope and constraints

The problem of rare event bias can be addressed in many ways. This thesis is limited to the *internal methods*, which is a set of methods created from modifying the the formulation or algorithm of certain statistical procedures so that they become more suitable for rare event data. The *sampling methods*, i.e., techniques that alleviate rare event bias by altering the data structure, are not included in this study. However, some of the sampling methods were briefly described in Section 3.3 for informational purposes.

Additionally, this study focuses only on a few number of classifiers from the SVM and the GLM family, respectively. Other popular classifiers, such as Artificial neural networks, elastic nets, and random forest, are not included in this study. Moreover, other modifications in the SVM and the GLM family than those listed in Table 4.1 are not included in this study.

Finally, in evaluating the output from different statistical procedures, I focused only on their classification performance and probabilistic output. Other subjects for evaluation, such as sparsity and model selection, are outside the scope of this study.

1.4 Outline of chapters

This thesis is organised as followed: Chapter 2 provides the mathematical details behind the statistical methods evaluated in this study; Chapter 3 describes the rare event bias using the mathematical details behind LR introduced in the previous chapter, with focus on why such bias is so problematic and what remedies are available to alleviate it; Chapter 4 presents the evaluation metrics used in this thesis and describes how the simulation study and the empirical analysis was constructed and performed; Chapter

5 and 6 report the results of simulation study and empirical analysis, respectively; Chapter 7 is devoted to the discussion of the results in previous chapters and finally, Chapter 8 concludes.

Chapter 2

Theory

2.1 Modelling binary response

2.1.1 Parametric and non-parametric methods

In the field of statistics, the term supervised learning refers to a set of methods that are used to model a certain outcome, or *output*, based on the values of a number of features, or *inputs* [James et al., 2013]. More formally, let $X = (X_1, X_2, \ldots, X_h)$ be a vector consisting h independent input variables – also known as *predictors* – and Y be the response variable. Both X and Y are real-valued random variables, whose values are determined by a certain probability distribution. Since our goal is to learn about Y given the information provided by X, we try to find a function, $f(\cdot)$, that transform the values of X into Y. That is, we assume the following relationship between our predictors and response

$$Y = f(X) + \epsilon \tag{2.1}$$

where ϵ is called the *error term*, i.e. the error between Y, the true response, and $f(X)$, the mapping based on the information provided by the predictors [Friedman et al., 2009; James et al., 2013]. The reason why the error term exists is that we do not expect to gain full knowledge about the response through the values of a set of predictor selected *a priori*. The mapping $f(X)$ is not the same as Y, it only represents the systematic information provided by X on Y. In other words, $f(X)$ is merely a *prediction* of Y. In addition, it can be shown that the best prediction of Y at any point $X = x$ is the conditional mean of the response Y given X (see Friedman et al. [2009, p.18] for more details), that is

$$f(x) = \mathbb{E}[Y|X = x] \tag{2.2}$$

which also means that

$$f(X) = \mathbb{E}[Y|X]. \tag{2.3}$$

In reality, the mapping $f(\cdot)$ is unknown to us. The essence of supervised learning is therefore to estimate this function and then use the estimation $\hat{f}(\cdot)$ to either make prediction of the future outcome when encountering a new set of values of X or draw inference about the relationship between the response and the predictors, as described above, or both [James et al., 2013].

According to James et al. [2013], there are two sets of methods for estimating $f(\cdot)$ – *parametric* models and *non-parametric* models. When estimating $f(\cdot)$ using the parametric approach, we first assume that $f(\cdot)$ has certain functional form and determine a fixed number of parameters to this function. After that, we estimate $\hat{f}(\cdot)$ by fitting or training a model to our observed data. These observed data points we use to estimate $\hat{f}(\cdot)$ are also called *training set*. The non-parametric approach, on the other hand, does not make such an assumption on $f(\cdot)$ as in the parametric case. For this reason, the non-parametric models are more flexible and hence more accurate than the parametric ones. However, the superiority of the non-parametric methods in predictive accuracy does not come without a price: the more flexible one method is, the harder the interpretation of its resulted output will be. Therefore, we have a *trade-off between prediction accuracy and model interpretability* [James et al., 2013]. As illustrated in figure 2.1, parametric approaches, such as LR, are easier to interpret compared to the non-parametric approaches, such as SVM. Meanwhile, the relatively high flexibility in methods from the non-parametric family might result in higher prediction accuracy in comparison to those from the parametric family. The advantages and disadvantages of LR and SVM, the two groups of methods that are the focus of this study, are discussed further in the methods' respective sections, i.e., Sections 2.2 and 2.3.

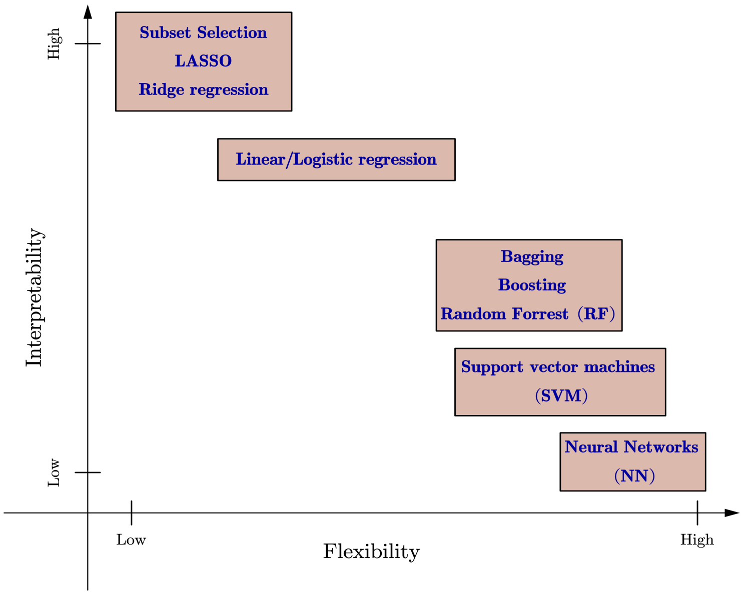


Figure 2.1: The trade-off between model flexibility and model interpretability. This figure is a modified version based on two similar figures included in James et al. [2013] and Hain & Jurowetzki [2020], respectively.

2.1.2 The linear function

Regarding the parametric methods, one of the most common approaches is to model the response Y as a *linear function* of the inputs $X = (X_1, X_2, \ldots, X_h)$. The linear function is defined as followed[2]

$$f(X) = \langle W, X \rangle + b \tag{2.4}$$

where the term b denotes the *intercept* of the linear model [Friedman et al., 2009]. In some occasions, especially in machine learning literature, the quantity b is termed as the *bias* since it can be considered as the residual error between the linear model's prediction and the true response [Friedman et al., 2009; Murphy, 2012]. Meanwhile, the term $\langle W, X \rangle$ is the *inner products* between the weight vector[3]

[2] The notation expressing the linear function varies among different literature. In Murphy [2012], for instance, the following notation is used

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + \epsilon$$

where the T above $\mathbf{w}$ indicates the transpose. In this thesis, I choose to use the version with angle brackets $\langle W, X \rangle$ for the purpose of highlighting the element of inner product in this equation.

[3] In some literature, such as Friedman et al. [2009] and James et al. [2013], the weight vector is often express as the vectors of model parameters and denoted as β (sometimes even θ). In these works, the linear function is denoted as

$$Y = X^T\beta$$

$W = (w_1, w_2, \ldots, w_h)$ and the inputs X. Using the definition of an inner product[4], we can rewrite Eq.(2.4) as

$$f(X) = \sum_{j=1}^{h} w_j X_j + b. \tag{2.5}$$

As we will see in Sections 2.2 and 2.3, both the LR and SVM can be expressed in the same manner as Eq.(2.5) above or closely related to it.

2.1.3 Binary response

By *binary response* we mean a discrete random variable Y that only has two outcomes, for instance $Y \in \{0, 1\}$. In classification setting, this means that we are dealing with a qualitative response such that the number of classes is equal to 2 (i.e., $K = 2$). The random variable Y is said to have a *Bernoulli distribution* with success probability p, which can be denoted as $Y \sim \mathit{Bernoulli}(p)$[5] where $p = \Pr(Y = 1)$ [Murphy, 2012].

When we model the binary response, often with the aim of predicting the probability that a certain event occurs (i.e., when $Y = 1$), we want to find a probability function $p(X)$ that returns the *conditional probability* of the random variable Y given the values of X, that is [James et al., 2013; Murphy, 2012]

$$p(X) = \Pr(Y = 1|X) = \mathbb{E}[Y|X]. \tag{2.6}$$

More formally, assume that we have a sample consisting n independent observations. Let $y_i \in \{0, 1\}$ denote the value of ith observation for the response variable Y_i and x_{ij} the value of ith observation for predictor X_j, where $i = 1, \ldots, n$, and $j = 1, \ldots, h$. We have

$$(Y_i = y_i | X_j = x_{ij}) \sim \mathit{Bernoulli}(p_i)$$

as followed from the fact that Y is binary [Friedman et al., 2009; Wasserman, 2004]. The probability model that predicts the the binary response Y at every points $X = x$ will be

$$p(x; \theta) = \Pr(Y = y_i | X = x; \theta) \tag{2.7}$$

where $\theta = (\theta_0, \theta_1, \ldots, \theta_h)$ is the vector of parameters for the probability model. In addition, as followed from Eq.(2.6) and Eq.(2.7), we have

$$p(X; \theta) = \Pr(Y = 1|X; \theta) \tag{2.8}$$

$$1 - p(X; \theta) = \Pr(Y = 0|X; \theta). \tag{2.9}$$

Note that the mapping $p(\cdot)$ corresponds to $f(\cdot)$ above, only that the former is a function such that $p : X \to [0, 1]$ given a set of model parameters θ. In the classification setting, the model $p(X; \theta)$ is a probabilistic classifier. We can obtain $p(X; \theta)$ either by first constructing a joint probability model of the

[4] Let a and b be two vectors, the inner product of these two vectors, $\langle a, b \rangle$ are defined as

$$\langle a, b \rangle = \sum_{i=1}^{n} a_i b_i$$

[5] In some literature, the response variable is indexed as $Y = (Y_1, \ldots, Y_n)$ and represents a data set composed of n independent Bernoulli trials. In such case, Y is said to have a binomial distribution $Y \sim \mathit{Bin}(n, p)$.

form $p(Y, X)$ and then deriving the conditional probability from it (*generative* approach), or by fitting a model of the form $p(X; \theta)$ to our data directly (*discriminative* approach) [Murphy, 2012]. Regarding the subsequent classification procedures, we used to impose a decision rule by selecting some threshold τ that assign the inputs X to one class (e.g. $Y = 1$) when $p(X; \theta) = \Pr(Y = 1|X; \theta) > \tau$ and to the other class when $p(X; \theta) \leq \tau$. Such procedure is called *latent variable threshold model* [Agresti, 2015]. More detail about the threshold model in the case of logistic regression is provided in Section 2.2.4.

2.2 Logistic Regression (LR)

2.2.1 Model formulation

The probability model $p(X; \theta)$ can have many candidates, one of them is the so-called *logistic* or *sigmoid* function shown in Figure (2.2). The sigmoid function is defined as

$$sigm\,(z) \triangleq \frac{1}{1 + \exp(-z)} = \frac{e^z}{1 + e^z} \tag{2.10}$$

where z is some arbitrary variable. By replacing z with the combination of inputs X and the model parameters θ, we obtain the model formulation for the *logistic regression* (LR)

$$p(X; \theta) = sigm(X; \theta) = \frac{\exp\left(\theta^T X\right)}{1 + \exp\left(\theta^T X\right)} \tag{2.11}$$

where the quantity $\theta^T X$ is assumed to be linear, that is [Agresti, 2015; Friedman et al., 2009]

$$\theta^T X = \sum_{j=0}^{h} \theta_j X_j. \tag{2.12}$$

In addition, by replacing the intercept θ_0 with b and the rest of model parameters $(\theta_1, \ldots, \theta_h)$ with $W = (w_1, \ldots, w_h)$, we can rewrite the model formulation of LR in the same manner as in Eq.(2.5), that is, the inner product of the weights and the predictors

$$p(X) = sigm(X) = \frac{\exp\left(\sum_{j=1}^{h} w_j X_j + b\right)}{1 + \exp\left(\sum_{j=1}^{h} w_j X_j + b\right)}. \tag{2.13}$$

Alternatively, we can formulate the LR model in terms of *odds*, i.e. the quantity $p(X)/[1 - p(X)]$. Note that given the definition of the sigmoids function in Eq.(2.10), we can rewrite Eq.(2.13) as

$$p(X) = \frac{1}{1 + \exp\left(-\left(\sum_{j=1}^{h} w_j X_j + b\right)\right)}.$$

Then we have

$$
\begin{aligned}
p(X) + \exp\left(-\left(\sum_{j=1}^{h} w_j X_j + b\right)\right) p(X) &= 1 \\
\frac{p(X)}{\exp\left(\sum_{j=1}^{h} w_j X_j + b\right)} &= 1 - p(X) \\
\frac{p(X)}{1 - p(X)} &= \exp\left(\sum_{j=1}^{h} w_j X_j + b\right).
\end{aligned}
$$

Taking the log of the last equation above, we obtain the *logit function*, also known as the *log-odds* [James et al., 2013; Wasserman, 2004]

$$
logit\,(p(X)) = \log\left(\frac{p(X)}{1 - p(X)}\right) = \sum_{j=1}^{h} w_j X_j + b. \tag{2.14}
$$

In other words, the LR is essentially a linear model, where the linearity lies in the log-odds.

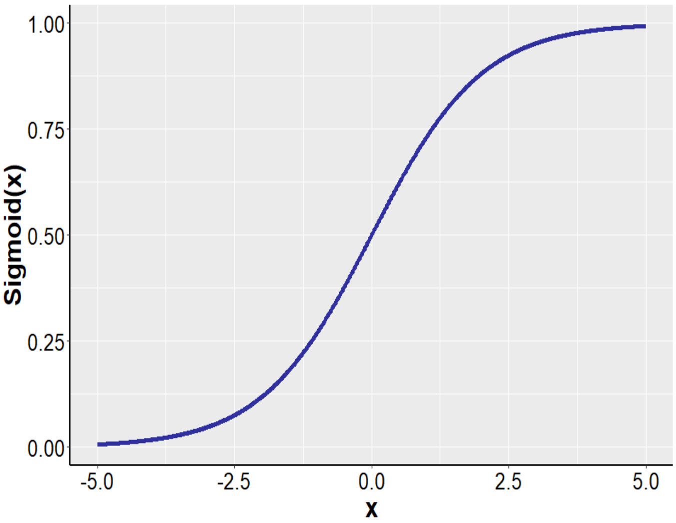


Figure 2.2: The Sigmoid function

2.2.2 Inference and estimation of the model parameters

As shown in Eq.(2.13) and Eq.(2.14), the formulation of the logistic regression model is entirely depending on the quantity $\sum_{j=1}^{h} w_j X_j + b$ (or $\theta^T X$ in the matrix form). The sigmoid function $sigm(\cdot)$ is merely a function that projects the the value of $\theta^T X$ – which could range between $(-\infty, \infty)$ – into the interval [0,1] and hence ensures that the probabilities generated from the model sum to 1 [Friedman et al., 2009]. In classification setting, the above feature means that the quantity $\theta^T X$ also determines the class to which we assign an observation given to the observation's values in X. In other words, the set of model parameters[6] θ reveals the relationship between the outputs and the inputs. For this reason, the inference and estimation of the model parameters θ are of paramount importance.

As implied by Eq.(2.14), for each predictor $X_i, i = 1, \ldots, h$, the magnitude of its corresponding parameter θ_i is the instantaneous effect on the log-odds followed by a one-unit change in the particular X_i in question [James et al., 2013]. However, the instantaneous effect of some predictor X_i on the

[6] For convenience, we use $\theta = (\theta_0, \theta_1, \ldots, \theta_h)$ to denote the model parameters instead of w_j, $j = 1, \ldots, h$, and b

probability $p(X)$ is not fixed and might vary depending on the value of X_i. But regardless of the value of X_i, if $\theta_i > 0$, then increase in X_i will also increase the probability $p(X)$; In the case where $\theta_i < 0$, an increase in X_i will induce a decrease in $p(X)$ [James et al., 2013]. Furthermore, if $\theta_i = 0$, then it would imply that the response Y is conditionally independent of X_i, given the other predictors [Agresti, 2015]. In addition, note that θ_0 is the model parameter for X_0, a column vector of ones. The quantity corresponds to the intercept (or bias) discussed in Section 2.1.2 and equals to $\mathbb{E}\left[Y|X=0\right]$. In other words, we can interpret θ_0 as our expectation about the log-odds in absence of information provided by the predictors [James et al., 2013].

In practice, the set of model parameters θ is unknown to us and has to be estimated. One way of doing that is to find the set of estimated parameters $\hat{\theta}$ that maximises the *likelihood function* of our sample. Such approach is called *maximum likelihood estimation* (MLE). More formally, suppose we have collected a sample of size n. Let $y_1, y_2, \ldots, y_n$ be the observed values of the corresponding random variables $Y_i, i = 1, \ldots, n$. Since we only consider the discrete case in this thesis, the joint distribution of Y_i is given by a *probability mass function* (pmf) $p(y_1, y_2, \ldots, y_n)$. Assuming the probability distribution of the observations y_i can be model by some unknown parameters θ, we specify the likelihood function of our observation as

$$\begin{aligned} L\left(\theta|y_1, y_2, \ldots, y_n\right) &= p\left(\theta|y_1, y_2, \ldots, y_n\right) \\ &= p\left(\theta|y_1\right) p\left(\theta|y_2\right) \ldots p\left(\theta|y_n\right) \\ &= \prod_{i=1}^{n} p\left(\theta|y_n\right). \end{aligned} \tag{2.15}$$

Thus, the likelihood function, if formulated as above, provides us with the probability or *likelihood* of observing the events $\{Y_i = y_i\}$, given the values of θ [Wackerly et al., 2008]. That is to say, the most reasonable value – the best estimate – of the unknown parameter θ is the one that maximises the probability of observing the same events $\{Y_i = y_i\}$ as in our sample [Friedman et al., 2009]

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \ \Pr\Big(\text{Observing the events } \{Y_i = y_i\}\Big).$$

We consider the case with h predictors as in Section 2.1.3. Let $\mathbf{x}_i = (x_{i0}, x_{i1}, \ldots, x_{ih})$ be a vector of observed values of the h inputs in ith sample observation. Then the (conditional) likelihood function for sample observations $(Y_i = y_i|\mathbf{x}_i) \sim Bernoulli(p_i)$, where $p_i = p(\mathbf{x}_i)$, is

$$L(\theta|y_i) = \prod_{i=1}^{n} p(\mathbf{x}_i;\theta)^{y_i} \left(1 - p(\mathbf{x}_i;\theta)\right)^{1-y_i}. \tag{2.16}$$

Instead of maximising Eq.(2.16) directly, it is usually easier to maximise the logarithm of it, the *log-likelihood function*. The log-likelihood function $\ell\left(\theta|y_i\right)$ has the same global maximum as the original function while still depends on θ, as shown in Eq.(2.17) below [Friedman et al., 2009; Greene, 2018]

$$\ell(\theta|y_i) = \sum_{i=1}^{n} y_i \log p(\mathbf{x}_i;\theta) + \sum_{i=1}^{n} (1 - y_i) \log(1 - p(\mathbf{x}_i;\theta)). \tag{2.17}$$

As followed from Eq.(2.10) and the fact that $1 - p(X;\theta) = 1/[1 + \exp(\theta^T X)]$, the log-likelihood for the logistic regression model is equal to

$$\ell(\theta|y_i) = \sum_{i=1}^{n} \Big[y_i \theta^T \mathbf{x}_i - \log\left(1 + \exp\left(\theta^T \mathbf{x}_i\right)\right) \Big]. \tag{2.18}$$

To find the set of $\hat{\theta}_{MLE}$ that maximises Eq.(2.18), we derive the partial derivative $\frac{\partial \ell(\theta)}{\partial \theta}$ and set it equal to 0, that is [Agresti, 2012]

$$\frac{\partial \ell(\theta)}{\partial \theta} = \sum_{i=1}^{n} \mathbf{x}_i y_i - \sum_{i=1}^{n} \mathbf{x}_i \frac{\exp\left(\theta^T \mathbf{x}_i\right)}{1 + \exp\left(\theta^T \mathbf{x}_i\right)} = 0 \tag{2.19}$$

The Eq.(2.19) above has a numerical solution, which can be obtained by using the Newton-Raphson algorithm, an iterative process for solving non-linear equations. The mathematical details behind the method are beyond the scope of this study. We therefore refer readers who are interested to the works of Agresti [2012] and Friedman et al. [2009] for further reading.

2.2.3 LR from the perspective of Generalised Linear Models (GLM)

The logistic regression model discussed previously is, in fact, a special case of the *generalised linear models* (GLM). The GLM has three essential parts: 1) random component, 2) linear predictor, and 3) link function. LR differs from the other members of the GLM family, such as linear regression and Poisson regression, in the distribution of the random component and the link function – a function that connects the random component to the linear predictor [Agresti, 2015].

We begin with defining the *exponential family distribution*, which encompasses some of the well-known probability distributions such as Gaussian, Poisson, and Bernoulli (or Binomial). Let $y_i, i = 1, \ldots, n$, be the observed values of the random variable Y and assume that the quantity y_i, which is also the random component of a GLM, has the following conditional distribution

$$f(y_i|\theta_i, \phi) = \exp\left(\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\right) \tag{2.20}$$

where θ_i is the *natural parameter*[7] and ϕ the *dispersion* or *scale parameter*. In addition, the quantity $c(y_i, \phi)$ is a normalising constant and $a(\cdot)$ and $b(\cdot)$ are some functions of the parameters ϕ and θ_i, respectively. By choosing certain combination of $a(\cdot)$ and $b(\cdot)$, we can rewrite Eq.(2.20) into the pmf or pdf of different distributions [Agresti, 2015]. The quantities $b(\theta_i)$ and $a(\phi)$ are also essential to the GLM in the sense that they determine the expected value and variance of the random component, that is

$$\mathbb{E}\left[Y_i\right] = b'\left(\theta_i\right) \tag{2.21}$$

$$\mathrm{Var}\left(Y_i\right) = b''\left(\theta_i\right) a\left(\phi\right). \tag{2.22}$$

In GLM, we assume the observed values of inputs $\mathbf{x}_i = (x_{i0}, x_{i1}, \ldots, x_{ih})$ relate to the expected value of random component, $\mathbb{E}\left[Y_i\right] = \mu_i$, through a *link function* $g(\cdot)$ such that

$$g(\mu_i) = \eta_i \tag{2.23}$$

where the quantity η_i is called the *linear predictor* and is defined in a similar way as the linear function

[7] Note that θ_i is indexed here, that is because this parameter is observation specific in GLM setting. In other words, it can be expressed as a function of the observed inputs $\theta_i = \theta(x_i)$[Agresti, 2012].

in Eq.(2.5), i.e. as the linear combination of the unknown parameters w_j and their related predictors X_j

$$\eta_i = \sum_{j=1}^{h} w_j x_{ij} \tag{2.24}$$

where x_{ij} is the observed values for predictors $X_j, j = 1, \dots, h$. Eq.(2.23) also suggests that the functions used in linear models – such as $sigm(\cdot)$ mentioned above – are equivalent to the inverse of the link function $g^{-1}(\cdot)$, which maps the values of a linear combination of inputs and model parameters to the expected response [Agresti, 2015].

As a member of the exponential family, the pmf of a Bernoulli distributed random variable can be easily transformed into the format specified in Eq.(2.20). Suppose that we have a sample which $(Y_i = y_i|\mathbf{x}_i) \sim Bernoulli(p_i)$ with the pmf defined as

$$f(y_i; p_i) = p_i^{y_i}(1-p_i)^{1-y_i} \tag{2.25}$$

which is the same as

$$\begin{aligned} f(y_i; p_i) &= \exp\left(\log f\left(y_i; p_i\right)\right) \\ &= \exp\left(y_i \log p_i + (1-y_i)\log\left(1-p_i\right)\right) \\ &= \exp\left(y_i \log\left(\frac{p_i}{1-p_i}\right) + \log\left(1-p_i\right)\right). \end{aligned} \tag{2.26}$$

The equation above is the same as Eq.(2.20) with $a(\phi) = 1$, $c(y_i, \phi) = 1$ and natural parameter θ and the function $b(\theta)$ take the following form

$$\theta_i = \log\left(\frac{p_i}{1-p_i}\right) \tag{2.27}$$

$$\begin{aligned} b\left(\theta_i\right) &= -\log\left(1-p_i\right) = -\log\left(1 - \frac{\exp\left(\theta_i\right)}{1+\exp\left(\theta_i\right)}\right) \\ &= \log\left(1+\exp\left(\theta_i\right)\right). \end{aligned} \tag{2.28}$$

In addition, as followed from Eq. (2.21) and (2.22) above, we have

$$\mu_i = \frac{\exp\left(\theta_i\right)}{1+\exp\left(\theta_i\right)} = p_i \tag{2.29}$$

$$\text{Var}\left[Y_i\right] = \frac{\exp(\theta_i)}{\left(1+\exp(\theta_i)\right)^2} = p_i(1-p_i) \tag{2.30}$$

As mentioned previously, the logistic regression model is linear in log-odds, meaning that the linear predictor $\eta_i = logit(p_i)$. Hence, using the result of Eq.(2.29), we can define the link function $g(\cdot)$ as

$$g\left(\mu_i\right) = \eta_i = \log\left(\frac{p_i}{1-p_i}\right) = \theta_i \tag{2.31}$$

The link function defined above is also called the *logit link*. In other words, the LR is equivalent to a GLM that uses logit link to model a Bernoulli distributed random component. As a sidenote, Eq.(2.31) also indicates that the logit link is also a *canonical link*, i.e. a link function $g(\cdot)$ that transform the mean μ_i to the natural parameter θ_i [Agresti, 2015].

2.2.4 Classification using LR

Concepts such as linear predictor and link function discussed in the previous section are closely related to the latent variable model for the classification of binary responses. In such model, we assume that there is an unobserved continuous response y_i^* that for each observation $i = 1, \ldots, n$ satisfies [Agresti, 2015]

$$y_i^* = \sum_{j=1}^{h} w_j x_{ij} + \epsilon_i = \eta_i + \epsilon_i \tag{2.32}$$

where η_i is the linear predictor, while ϵ_i is the error term (or bias) described in Eq.(2.1) and has a distribution with zero mean and the cumulative distribution function (cdf) F_ϵ. By imposing a threshold τ as decision rule, we can construct a linear classifier with the following formulation

$$y_i = \begin{cases} 1 \text{ if } y^* > \tau \\ 0 \text{ if } y^* \leq \tau \end{cases} \tag{2.33}$$

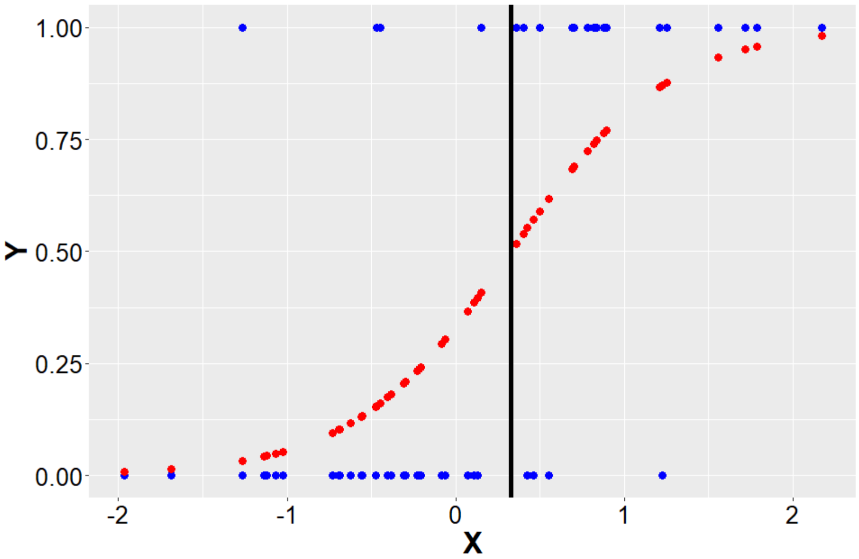


Figure 2.3: Data of size $n = 50$ simulated from a logistic model; Blue dots are the original values; Red dots are the predicted values; Black solid line is the decision boundary with the threshold set to $\hat{p} = sigm(X; \hat{\theta}) = 0.5$, or equivalently, $\tau = 0$.

Graphically, imposing τ means that we are drawing a vertical decision boundary at the point that $y^* = \tau$. We then classify observations to the left of this line as 0 and those to the right as 1 [Murphy, 2012], as shown in Figure 2.3. The choice of the threshold τ is arbitrary. One common approach is to set the predicted probability cut-off, $\hat{p}_0$, to 0.5. In the case of LR, this is equivalent to setting $\tau = 0$. Another common approach is to set $\hat{p}_0 = \bar{y}$, i.e. the fraction of 1 in the data [Agresti, 2015]. In the following chapters of this thesis, the term $\bar{y}$ is also referred as *rarity*.

Meanwhile, the classifier in Eq.(2.33) connects to the probability model $p(\mathbf{x}_i) = \Pr(y_i = 1|\mathbf{x}_i)$ in the sense that

$$\begin{aligned}\Pr(y_i = 1|\mathbf{x}_i) = \Pr(y_i^* > \tau|\mathbf{x}_i) &= \Pr\left(\eta_i + \epsilon_i > \tau|\mathbf{x}_i\right) \\ &= 1 - \Pr\left(\epsilon_i \leq \tau - \eta_i|\mathbf{x}_i\right) \\ &= 1 - F_\epsilon\left(\tau - \eta_i|\mathbf{x}_i\right) \\ &= 1 - F_\epsilon\left(\tau - \sum_{j=1}^{h} w_j x_{ij}\middle|\mathbf{x}_i\right). \end{aligned} \tag{2.34}$$

Since the choice of τ is arbitrary, this quantity is unrelated to our observed data. The result does not lose its generality if we set $\tau = 0$. Meanwhile, because $F(z) = 1 - F(-z)$, as followed from the definition of cdf, we have [Agresti, 2015]

$$\Pr(y_i = 1) = F_\epsilon\left(\sum_{j=1}^{h} w_j x_{ij}\right), \quad \text{and} \quad F_\epsilon^{-1}\Big(\Pr\left(y_i = 1\right)\Big) = \sum_{j=1}^{h} w_j x_{ij} = \eta_i. \tag{2.35}$$

This shows that the inverse of the cdf F_ϵ^{-1} returns the linear predictor η_i, making it equivalent to the link function $g(\cdot)$ described in previous section. Hence, the latent variable model for classification can be treated as a version of GLM. And LR is the case when F_ϵ is the cdf of the standard *logistic distribution*, that is

$$F_\epsilon(z) = \frac{e^z}{1 + e^z} = sigm(z), \quad \text{and} \quad F_\epsilon^{-1}(z) = g(z) = logit(z). \tag{2.36}$$

Followed from the result above, we can obtain the pdf of the standard logistic distribution as $f(z) = e^z/(1 + e^z)^2$. As Figure 2.4 shows, the logistic density is symmetry. Hence, one important feature of the LR is that, following a change in the inputs x_i, the probability p_i approaches to 1 *at the same rate* as it approaches to 0 [Agresti, 2015].

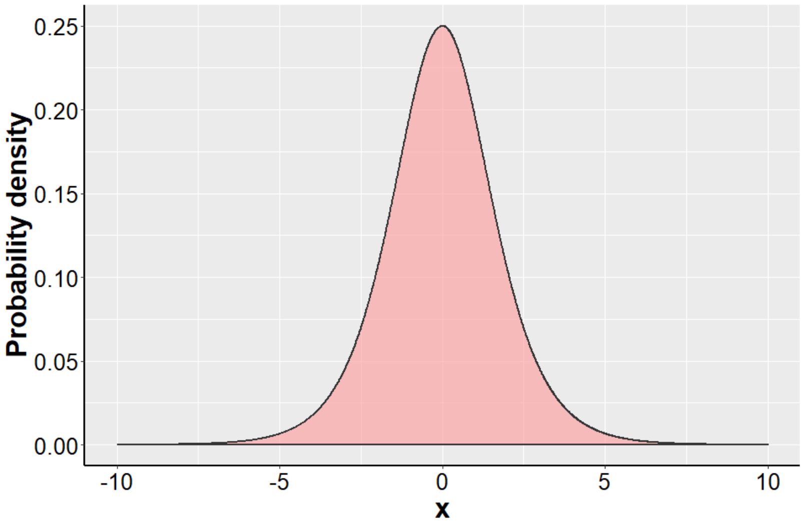


Figure 2.4: Probability density of standard logistic distribution

2.3 Support Vector Machine (SVM)

2.3.1 Classification using a separating hyperplane

In this section, we discuss the classification method that uses a *separating hyperplane* as decision boundary. The *hyperplane* is defined as a flat affine subspace of dimension $p-1$ in a p-dimensional space. In other words, if we have a plane (i.e., $\mathbb{R}^2$), then the hyperplane is a straight line (i.e., $\mathbb{R}^1$) that not necessarily passes through the origin [James et al., 2013]. The concept of a separating hyperplane is essential to the support vector machines (SVM). Although the approach of using a hyperplane for classification might be seen completely different from the previously mentioned LR at first glance, these two methods are – as shown in the following sections – closely related in many aspects.

Suppose that we have a training data set S consisting n ordered pairs $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$, where the dimension of the feature spaces $\mathbf{x}_i$ is equal to the number of predictors h. Assume that our data S is linearly separable, that is to say, there exists at least one separating hyperplane that perfectly separates the training observations (i.e., classification with zero error). We can then model such separating hyperplane using a linear function similar to Eq.(2.5), hence defining *a separating hyperplane with respect to training set S* by the following equation [Cristianini & Shawe-Taylor, 2000; Friedman et al., 2009]

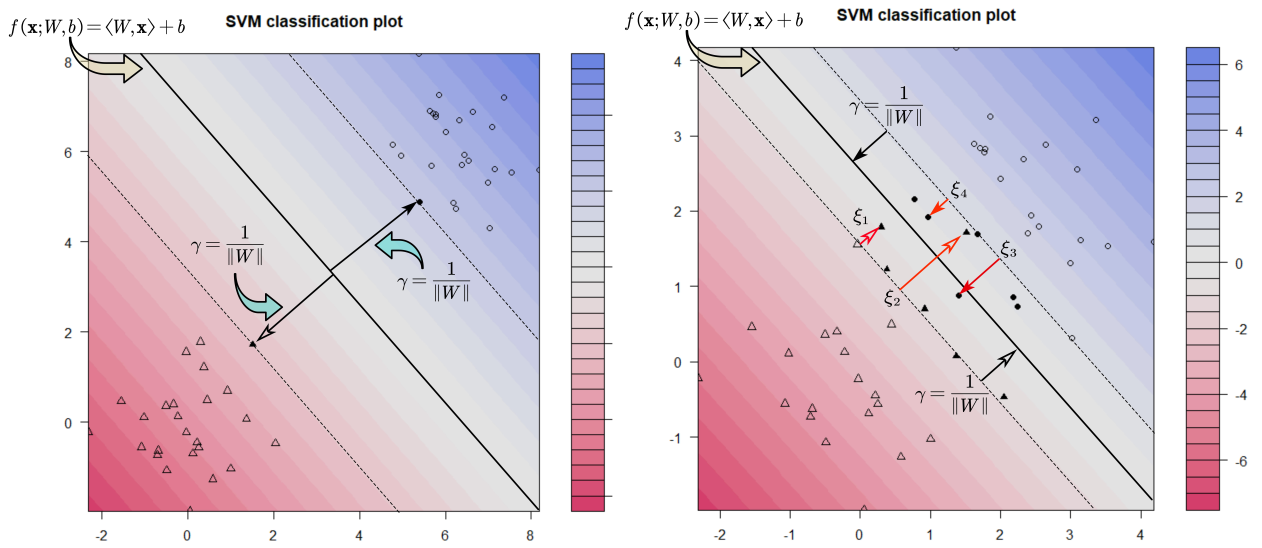
$$f(\mathbf{x}; W, b) = \langle W, \mathbf{x} \rangle + b = \sum_{j=1}^{h} w_j x_{ij} + b = 0. \tag{2.37}$$

The equation above means that the separating hyperplane is basically a set of data points $\mathbf{x}_i$ on the feature space whose linear combination – defined by the parameters (W, b) – equals to zero. Hence we can denote a separating hyperplane by the set of parameters (W, b) that defines it. From here we can see that the definition of separating hyperplanes is similar to the model formulation of LR shown in Eq.(2.14), although the inference of parameters (W, b) in the former case has a more geometrical nature: the weight vector $W = (w_1, \ldots, w_h)$ is a direction orthogonal to the hyperplane and the bias parameter b is a vector controlling the distance between the hyperplane and the origin, i.e. it has the ability of making parallel "shift" in the hyperplane [Murphy, 2012]. In addition, since the hyperplane is formulated as an equation that equals to zero, it would be convenient to redefine our binary outcome variable as $y_i \in \{-1, 1\}$ [James et al., 2013]. Note that despite change in notation, the binary outcome defined as $y_i \in \{-1, 1\}$ is not different from the one defined as $\{0, 1\}$, since the value of these numbers has no real meaning – it is merely a label created to distinguish between two classes. With that said, we can construct a classifier based upon a separating hyperplane as follows [Wasserman, 2004]

$$\operatorname{sgn}(z) = \begin{cases} -1 & \text{if } z < 0 \\ 0 & \text{if } z = 0 \\ 1 & \text{if } z > 0 \end{cases} \tag{2.38}$$

where $\operatorname{sgn}(\cdot)$ is the *sign function* and $z = f(\mathbf{x}; W, b) = \sum_{j=1}^{h} w_j x_{ij} + b$ corresponds to the latent variable y^* in Eq.(2.33) above.

The classifier defined in Eq.(2.38) works as followed: the separation hyperplane $f(\mathbf{x}; W, b) = 0$ divides the feature space in one positive side and one negative side, so for a new observation $\mathbf{x}^*$ whose linear combination $f(\mathbf{x}^*) > 0$, we classify it to the positive side of the feature space; if $f(\mathbf{x}^*) < 0$, we classify it to the negative side. This is also the reason why $\operatorname{sgn}(\cdot)$ is used. In addition, we can use the *magnitude* of $f(\mathbf{x}_i; W, b)$ as a measure for the confidence we have for our classification. If $f(\mathbf{x}^* \gg 0)$, i.e this new observation lies far away from the separating hyperplane, then we say that we are "confident" about that our classification is correct [James et al., 2013].



(a) Maximum margin classifier, separable case

(b) Support vector classifier, overlapping classes

Figure 2.5: Illustration of separating hyperplane, geometric margin and slack variable. In subfigures (a) and (b), the solid line is the separating hyperplane $f(\mathbf{x}_i;W,b)$; the black arrows are the geometric margins γ; the filled-in points/triangles are the support vectors. The red arrows in subfigure (b) are the slack variables ξ.

Since there might exist more than one (or indefinitely many) separating hyperplanes for a given linearly separable data set, as subfigure 2.5a shows, we have to find a way to determine which of these hyperplane is the best one. It reveals that the choice of the optimal hyperplane is closely related to the magnitude of $f(\mathbf{x}_i;W,b)$ we discussed previously. Note that Eq.(2.38) implies that an input in the sample $\mathbf{x}_i$ is assign to the correct class if and only if its linear combination $f(\mathbf{x}_i;W,b)$ has the same sign as y_i. In other words, for a given training set S, we want to ensure that [Friedman et al., 2009]

$$y_i f(\mathbf{x}_i;W,b) > 0, \quad \forall i = 1,\dots,n. \tag{2.39}$$

The quantity $y_i f(\mathbf{x}_i;W,b)$ in the constraint above is also known as the *functional margin of a hyperplane* (W,b) *with respect to the sample observation* $(\mathbf{x_i},y_i)$. We denote such a functional margin as $\tilde{\gamma}_i$, in order to distinguish it with the *geometrical margin* defined in Eq.(2.41). In addition, the *functional margin of a hyperplane with respect to the entire training set* S is defined as the minimum of the functional margins across all data points in the sample, that is [Cristianini & Shawe-Taylor, 2000]

$$\tilde{\gamma} = \min_{i=1,\dots,n} \tilde{\gamma}_i \tag{2.40}$$

As the definition above implies, the magnitude of the functional margin for a single observation, $\tilde{\gamma}_i$, reflects the degree of confidence we have towards our classification of that observation. It is therefore reasonable to consider $\tilde{\gamma}$ – the functional margin of the observation which we are least confident about its classification – to be the quality measure for our classifier. Hence, of all possible candidates, the hyperplane that maximises $\tilde{\gamma}$ must be the optimal one. However, this approach is problematic since scaling up the parameters (W,b) by some factor k will undoubtedly increase the functional margin, but the hyperplane in question will remain unchanged because it is defined to be an equation equal to zero. Normalisation of the parameters (W,b) is therefore needed. This leads us to the *geometric margin of a* *hyperplane* (W, b) *with respect to the sample observation* $(\mathbf{x_i}, y_i)$ [Herbrich, 2002]

$$\gamma_i = \frac{y_i f(\mathbf{x}_i; W, b)}{\|W\|} = \frac{\tilde{\gamma}_i}{\|W\|} \tag{2.41}$$

where the norm $\|W\| = \sqrt{\langle W, W \rangle}$. As in the case of functional margin, the *geometric margin of* (W, b) *with respect to the training set* S is

$$\gamma = \min_{i=1,\ldots,n} \gamma_i. \tag{2.42}$$

Thus, we are facing the following optimisation problem [Friedman et al., 2009]

$$\begin{aligned} \max_{\gamma, W, b} \quad & \gamma \\ \text{subject to} \quad & y_i \left(\sum_{j=1}^{h} w_j x_{ij} + b \right) \geq \gamma, \; i = 1, \ldots, n \\ & \|W\| = 1. \end{aligned} \tag{2.43}$$

In other words, we want to find the separating hyperplane that maximises the geometric margin with respect to our training data, subject to each of the observations having a functional margin at least the size of the geometric margin. Unfortunately, this optimisation problem is hard to solve, since the objective function is neither linear nor quadratic. And the constraint $\|W\| = 1$, which ensures the functional margin equals to the geometric margin, is non-convex [Herbrich, 2002]. Luckily, we can use the definition of geometric margin in Eq.(2.41) to eliminate the constraint $\|W\| = 1$ and rearrange Eq.(2.43) above as

$$\begin{aligned} \max_{\tilde{\gamma}, W, b} \quad & \frac{\tilde{\gamma}}{\|W\|} \\ \text{subject to} \quad & y_i \left(\sum_{j=1}^{h} w_j x_{ij} + b \right) \geq \gamma, \; i = 1, \ldots, n. \end{aligned} \tag{2.44}$$

The objective function above is still non-convex. But we can get rid of this non-convexity by imposing a scaling constraint $\tilde{\gamma} = 1$. This is possible because the norm $\|W\|$ ensures that the distance of any point to the hyperplane will not change after the re-scaling of parameters (W, b) [Murphy, 2012]. As followed from Eq.(2.41), the geometric margin which we aim to maximise can be expressed as $\gamma = 1/\|W\|$. Since maximising the quantity $1/\|W\|$ is the same as minimising $\|W\|^2$, we have[8]

$$\begin{aligned} \min_{W, b} \quad & \frac{1}{2}\|W\|^2 \\ \text{subject to} \quad & y_i \left(\sum_{j=1}^{h} w_j x_{ij} + b \right) \geq 1, \; i = 1, \ldots, n. \end{aligned} \tag{2.45}$$

To solve the (primal) optimisation problem above, we have to first transform it to an equivalent dual problem, using the *Lagrange duality*, and then perform dual optimisation[9]. The Lagrange function for

[8] Note that the quantity $1/2$ is added for the purpose of computational convenience and does not change the optimisation result [Murphy, 2012]

[9] Readers are referred to the works of Bishop [2006] and Friedman et al. [2009] for a more detailed description about this subject.

the (primal) optimisation problem in Eq.(2.46) is [Friedman et al., 2009]

$$\mathcal{L}_P(W,b,\alpha) = \frac{1}{2}\|W\|^2 - \sum_{i=1}^{n}\alpha_i\left(y_i\left(\sum_{j=1}^{h}w_j x_{ij}+b\right)-1\right) \tag{2.46}$$

where α_i is the Lagrange multiplier. Setting the partial derivatives of the parameters W and b to zero, we obtain

$$\frac{\partial\mathcal{L}_P}{\partial W} = 0 \;\Rightarrow\; W = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i \tag{2.47}$$

$$\frac{\partial\mathcal{L}_P}{\partial b} = 0 \;\Rightarrow\; b = \sum_{i=1}^{n}\alpha_i y_i = 0. \tag{2.48}$$

By plugging the results above back to the Lagrange function in Eq.(2.46), we can obtain the dual form (so-called *Wolfe dual*) which is defined by α_i only

$$\mathcal{L}_D(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_i\alpha_k y_i y_k \mathbf{x}_i^T\mathbf{x}_k. \tag{2.49}$$

Note that the quantity $\mathbf{x}_i^T\mathbf{x}_k = \langle\mathbf{x}_i,\mathbf{x}_k\rangle$, i.e. the inner product of two training observations. Finally, we have the following dual optimisation problem

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_i\alpha_k y_i y_k \langle\mathbf{x}_i,\mathbf{x}_k\rangle \\ \text{subject to} \quad & \alpha_i \geq 0 \text{ and } b = \sum_{i=1}^{n}\alpha_i y_i = 0 \\ & i = 1,\ldots,n. \end{aligned} \tag{2.50}$$

In other words, we first find the Lagrange multiplier α that maximises the objective function $\mathcal{L}_D(\alpha)$, then use this α to compute the optimal set of parameters (W,b), thereby also the optimal separating hyperplane. The solution of the optimisation problem in Eq.(2.50) can be obtained by implementing the *sequential minimal optimisation* algorithm (SMO) developed by Platt [1998]. In addition, this solution must also satisfy the *Karush-Kuhn-Tucker conditions*, in which the following constraint is included [Friedman et al., 2009]

$$\alpha_i\left(y_i\left(\sum_{j=1}^{h}w_j x_{ij}+b\right)-1\right) = 0, \forall i. \tag{2.51}$$

This constraint implies that the optimal hyperplane (W,b) will only be determined by those training observations $\mathbf{x}_i$ with functional margin $\tilde{\gamma}_i = 1$. Such observations are called *support vectors*. Observations that are not support vectors will have $\alpha_i = 0$ and hence can not influence our choice of the optimal set of (W,b) [Friedman et al., 2009].

Once we have solved the dual optimisation problem, we obtain a separating hyperplane defined as the following, using the relationship between W and α_i stated in Eq.(2.47)

$$\begin{aligned} \hat{f}(\mathbf{x}; \hat{W}, \hat{b}) = \left\langle \hat{W}, \mathbf{x} \right\rangle + \hat{b} &= \left\langle \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \ , \ \mathbf{x} \right\rangle + \hat{b} \\ &= \sum_{i=1}^{n} \alpha_i y_i \left\langle \mathbf{x}_i, \mathbf{x} \right\rangle + \hat{b} = 0 \end{aligned} \tag{2.52}$$

which leads us to the *maximum margin classifier* defined by parameters $(\hat{W}, \hat{b})$. In practice, however, it is nearly impossible to obtain such classifier. This is because real world data usually contain overlapping classes and hence are not linearly separable. In order to still use separating hyperplanes for classification in such cases, we have to make some compromises and allow some of the observations to be at the wrong side of the decision boundary. This leads us to the *soft margin* and hence *support vector classifier* [James et al., 2013].

2.3.2 Soft margin and support vector classifier

As mentioned previously, the maximum margin classifier can only apply to data that are linearly separable[10]. To extend its application to data with overlapping classes, we have to allow some observations to have $\tilde{\gamma} < 1$ and perhaps even violate the constraint in Eq.(2.39). With such relaxations, the optimisation problem in Eq.(2.45) becomes

$$\begin{aligned} \min_{W, b, \xi_1, \ldots, \xi_n} \quad & \frac{1}{2}\|W\|^2 + C \sum_{i=1}^{n} \xi_i \qquad (2.53) \\ \text{subject to} \quad & y_i \left(\sum_{j=1}^{h} w_j x_{ij} + b \right) \geq 1 - \xi_i, \ i = 1, \ldots, n \\ & \xi_i \geq 0, \ i = 1, \ldots, n \end{aligned}$$

where the quantity ξ_i in the objective function is called the *slack variable* and the constant C is a pre-specified, non-negative tuning parameter [James et al., 2013]. The size of the slack variable ξ_i determines whether an observation is misclassified: if $0 < \xi_i < 1$, then the observation are still at the correct side of the decision boundary even if it has a functional margin less than 1; if $\xi_i > 1$, then the observation appears at the wrong side of decision boundary. For this reason, the sum $\sum_{i=1}^{n} \xi_i$ sets the upper bound on the number of misclassifications. And the constraint that includes the slack variable, $y_i f(\mathbf{x}_i) \geq 1 - \xi_i$, is called the *soft margin constraint* [Murphy, 2012]. Meanwhile, the tuning parameter C controls the trade-off between the goal of maximising the margin (i.e., minimising $\|W\|^2$) and minimising training errors $\sum_{i=1}^{n} \xi_i$ [Bishop, 2006]. Violations are less tolerated when C is small, meaning that we fit the training data closely and it will result in a model that has low bias but large variance. On the contrary, large value of C means that we allow more violations to the margin, making the fitting procedure less hard and resulting in a model of low variance and high bias [James et al., 2013]. Hence, the tuning parameter C is closely connected with the *bias-variance trade-off*. For this reason, we usually use *k-fold cross validation* to choose the optimal value of C [Friedman et al., 2009].

As in the case of the maximal margin classifier, the optimisation problem above can also be rewritten to dual form using Lagrange function[Friedman et al., 2009]. The dual optimisation problem using soft margin is expressed as

[10] More specifically, if the training set is not linearly separable, then the algorithm that we use to compute such separating hyperplane – the perceptron algorithm developed by Rosenblatt [1958] – will not converge.

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_i \alpha_k y_i y_k \langle \mathbf{x}_i, \mathbf{x}_k \rangle \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C \text{ and } b = \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & i = 1, \ldots, n. \end{aligned} \tag{2.54}$$

And the constraint defined in Eq.(2.51) became

$$\alpha_i \left(y_i \left(\sum_{j=1}^{h} w_j x_{ij} + b \right) - (1 - \xi_i) \right) = 0, \forall i. \tag{2.55}$$

Hence, the support vectors in the linearly non-separable case are defined to be those that satisfy

$$y_i \left(\sum_{j=1}^{h} w_j x_{ij} + b \right) - (1 - \xi_i) \geq 0, \forall i \tag{2.56}$$

These support vectors define the separating hyperplane $(\hat{W}, \hat{b})$. And the classifier based on this hyperplane are called *support vector classifier*. Subfigure 2.5b shows an illustration of the relationship between the hyperplane $(\hat{W}, \hat{b})$, the slack variable ξ_i and the support vectors. We can see that some of the support vectors are lying on the line where $\tilde{\gamma} = 1$ (i.e., $\xi_i = 0$) with its Lagrange multiplier falling in the interval $0 < \alpha_i < C$. Those support vectors that are inside the functional margin (i.e., $\xi_i > 0$) are all characterised by $\alpha_i = C$ [Friedman et al., 2009].

2.3.3 Extension to non-linear cases using kernels

Like the maximum margin classifier, the support vector classifier described in the previous section is a linear classifier. For this reason, the support vector classifier is most suitable for data in which the relationship between the response and the predictors is – or assumed to be – linear. In general, linear classifiers perform poorly in non-linear relationships, where the decision boundary could be polynomial or circular. If we still want to use the linear classifier to model non-linear relationships, what we can do is to enlarge our data to a higher dimension. One (informal) way of doing that is to add quadratic or cubic terms to our linear model [James et al., 2013].

More formally, we introduce some function $\phi(\cdot)$ that maps the data points in our sample to a higher dimension. Since the the non-linear relationship will become linear in the higher dimension, we can apply our intended linear classifier in the enlarged feature space and compute the decision boundary. After this is completed, we map the result back to the original feature space [James et al., 2013]. Transforming the training set to a higher dimension using $\phi(\cdot)$ implies that Eq.(2.49), the objective function for the dual optimisation problem, becomes

$$\mathcal{L}_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_i \alpha_k y_i y_k \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_k) \rangle \tag{2.57}$$

and the solution function $\hat{f}(\mathbf{x}; \hat{W}, \hat{b})$ in Eq.(2.52) becomes

$$\begin{aligned} \hat{f}(\mathbf{x}; \hat{W}, \hat{b}) &= \left\langle \hat{W}, \phi(\mathbf{x}) \right\rangle + \hat{b} \\ &= \sum_{i=1}^{n} \alpha_i y_i \left\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \right\rangle + \hat{b}. \end{aligned} \tag{2.58}$$

Although the solution above can lead us to an non-linear decision boundary, the inner product $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle$ will be very difficult to compute explicitly. As the number of predictors increases, the dimension of the feature space will also increase. In the end, we might have to deal with an feature space of infinite dimension [Friedman et al., 2009]. The solution to this dimensional problem is the so-called *kernel trick*, which allows us to replace the $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle$ with a *kernel* $K(\mathbf{x}_i, \mathbf{x})$, making the computation more comprehensible for us [Murphy, 2012].

More formally, a kernel or *kernel function* is defined as a function K that takes two arguments $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, where $\mathcal{X}$ is some input space, and maps $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ satisfying the following condition[11]

$$K\left(\mathbf{x}, \mathbf{x}'\right) \triangleq \left\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \right\rangle. \tag{2.59}$$

In the most cases, the kernel function is symmetric (i.e., $K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x})$ and non-negative [Murphy, 2012]. Furthermore, in order to be a valid kernel, the mapping $K(\cdot, \cdot)$ must also satisfy the requirement that its corresponding kernel matrix (so-called *Gram matrix*), defined as

$$\mathbf{K} = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & \cdots & K(\mathbf{x}_n, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x}_1, \mathbf{x}_n) & \cdots & K(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}$$

must be positive semi-definite for all sets of inputs $\mathbf{x}_i, i = 1, \ldots, n$. Those $K(\cdot, \cdot)$ that fulfill this requirement are also called *Mercer kernel* [Murphy, 2012]. One of the most popular mercer kernel is the *Gaussian kernel*, defined as

$$K\left(\mathbf{x}, \mathbf{x}'\right) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right). \tag{2.60}$$

In addition to that, polynomial kernels and hyperbolic tangent (Sigmoid) kernel[12] are also commonly used [Bishop, 2006; Friedman et al., 2009]

$$d\text{-th Degree polynomial kernel:} \quad K\left(\mathbf{x}, \mathbf{x}'\right) = (1 + \langle \mathbf{x}_i, \mathbf{x} \rangle)^d \tag{2.61}$$

$$\text{Sigmoid kernel:} \quad K\left(\mathbf{x}, \mathbf{x}'\right) = \tanh(a \langle \mathbf{x}_i, \mathbf{x} \rangle + b). \tag{2.62}$$

Because of the space limit, I will not describe these kernels in more detail in this thesis. The readers are therefore referred to the works of Friedman et al. [2009] and Murphy [2012] for a more thorough description about different kernel functions as well as the proof of their validity.

To sum up, by replacing the inner product component in Eq.(2.58) with some of the valid kernel

[11] Note that in from Eq.(2.49) above, we hinted the expression of the Lagrange dual function using inner product. This is the reason why: the inner product notation creates a natural way for us to incorporate the kernels into our optimisation problem.

[12] Note that the Gram matrix of the Sigmoid kernel is not positive (semi) definite. The reason why we still use this kernel is because it is closely connected to the neural network, an another very popular method in machine learning [Bishop, 2006].

functions listed above, we can write our solution function as [Friedman et al., 2009]

$$\hat{f}(\mathbf{x}; \hat{W}, \hat{b}) = \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + \hat{b} \tag{2.63}$$

and classifiers that are based upon a separating hyperplane defined as this equation are called *support vector machines*. Note that the use of identity mapping $\phi(\mathbf{x}) = \mathbf{x}$ yields $K(\mathbf{x}_i, \mathbf{x}) = \langle \mathbf{x}_i, \mathbf{x} \rangle$, which will return the same linear decision boundary specified in Eq.(2.58). Such kernel function are called *linear kernel*. In practice, the linear kernel is useful when our original data is already high dimensional and hence the data is linearly separable in this high dimensional feature space, or when we want to compare the SVM output generated from different kernel functions. In the latter case, the SVM output from linear kernel can serve as a reference point [Bishop, 2006; Murphy, 2012].

2.3.4 Obtaining calibrated probabilistic outputs for SVM

As indicated in Eq.(2.38), for a new observation $\mathbf{x}^*$, the SVM classifier will first return the value of linear combination $f(\mathbf{x}^*)$ – a real number that is proportional to the distance of $\mathbf{x}^*$ to the separating hyperplane – and then assign $\mathbf{x}^*$ to one of the two classes based on the sign of $f(\mathbf{x}^*)$ [James et al., 2013; Pedregosa et al., 2020b]. This means that the SVM classifier, unlike LR, does not produce any probabilistic outputs [Murphy, 2012; Tipping, 2001]. In addition, recall that the weights W is interpreted as a vector orthogonal to the separating hyperplane. Hence, the coefficients of estimated parameters $(\hat{W}, \hat{b})$ that define the decision boundary can not be interpreted in the same way as the coefficients of a logistic model. The lack of probabilistic output is regarded as one of the major drawbacks of SVM, since a probability model expresses the uncertainty (or confidence) in the prediction and hence is essential to real-world classification tasks, especially those that involve asymmetric misclassification cost and varying class proportions[13] [Tipping, 2001].

Luckily, there are methods that we can use to obtain probabilistic outputs from SVM. The simplest way of doing that is to map the SVM outputs into the range $[0, 1]$ using the following normalisation formula [Niculescu-Mizil & Caruana, 2005]

$$\hat{p}_i = \frac{s_i - s_{min}}{s_{max} - s_{min}} \tag{2.64}$$

where $s_i = \hat{f}(\mathbf{x_i}; \hat{W}, \hat{b})$ denotes the score predicted by a SVM classifier for the observation $\mathbf{x_i}$ in the data set and is proportional to the distance between $\mathbf{x_i}$ and the seperating hyperplane. Under the assumption that we can interpret the SVM scores as probabilities[14], the normalisation above would give us the posterior probabilities for SVM. However, such ”probabilities” are often considered as *uncalibrated*. According to Kull et al. [2017], a probability model for classification is considered as *well-calibrated* if its probabilistic predictions for a certain class match the observed distribution of that class in the data. More formally, let $s(\cdot)$ denote the *scoring function* of a classifier and $s(\mathbf{x}) = s$ the score produced by this scoring function[15] for each of the observations. To keep things simple, we restrict ourselves to the binary classification case and the case where $s \in [0, 1]$. In this setting, the classifier is considered as

[13] i.e., imbalanced data and rare events, which is the subject of this thesis.

[14] This assumption might be doubtful since the argmax of the SVM scores is not always the same as argmax of the probabilities. In other words, the fact that a outcome receives the highest score, s_{max}, does not necessarily imply that this outcome would have $\hat{p} = 1$ [Pedregosa et al., 2020b].

[15] For instance, $s(\mathbf{x}) = sigm(\mathbf{x}; \hat{\theta})$ for LR and $s(\mathbf{x}) = \hat{f}(\mathbf{x}; \hat{W}, \hat{b})$ for SVM. To avoid confusion, I will henceforth use the term $s(\mathbf{x})$ to denote the scores produced by an arbitrary classifier, while the term $\hat{f}(\mathbf{x})$ refers specifically to the scores produced by a SVM.

well-calibrated if the conditional probability $\Pr\left(y=1|s\left(\mathbf{x}\right)=s\right)$ converges to the scores $s\left(\mathbf{x}\right)=s$, as the number of observations classified approaches infinity [Zadrozny & Elkan, 2002]. In other words, if the classifier gives the score $s\left(\mathbf{x}\right)=0.8$ to a number of observations, then approximately 80 percent of these observations should belong to the class $y=1$ [Kull et al., 2017; Zadrozny & Elkan, 2002].

The probability calibration of different classifiers can be visualised with *reliability diagrams*, also known as *calibration curves*. With the predicted probabilities for the positive class on its x-axis and the observed frequency of the actual positive cases given the predictions on its y-axis, the reliability diagram illustrates how close a classifier's probability predictions are to the actual probabilities observed in the data [Fernández et al., 2018; Prati et al., 2011]. Note that the variables on the reliability diagram's x and y-axis correspond to the score $s(\mathbf{x})$ and the conditional probability $\Pr\left(y=1|s\left(\mathbf{x}\right)=s\right)$, respectively, as mentioned above. For this reason, the convergence of $\Pr\left(y=1|s\left(\mathbf{x}\right)=s\right)$ and $s\left(\mathbf{x}\right)=s$ – hence also the ideal reliable prediction – is represented by an upward diagonal line [Prati et al., 2011]. To plot the reliability diagram for a certain classifier, one must first discretise the predicted probabilities into bins[16] and then for each bins compute the mean predicted value. Plotting these values against the true faction of the positive cases in each of the bins yields the calibration curve of the evaluated classifier [Fernández et al., 2018; Niculescu-Mizil & Caruana, 2005].

Figure 2.6 shows two reliability diagrams, which illustrate the out-of-sample performance of a linear SVM trained on synthesised datasets. As shown in subfigure 2.6a, the purple line, which represents the probabilistic predictions computed via the normalisation formula in Eq.(2.64), does not converge to the diagonal line of the reliability diagram. This is more clear in the case of imbalanced classes. As shown in subfigure 2.6b, the entire purple line is under the diagonal line, indicating a poor agreement between uncalibrated probabilities and their mean observed frequencies.

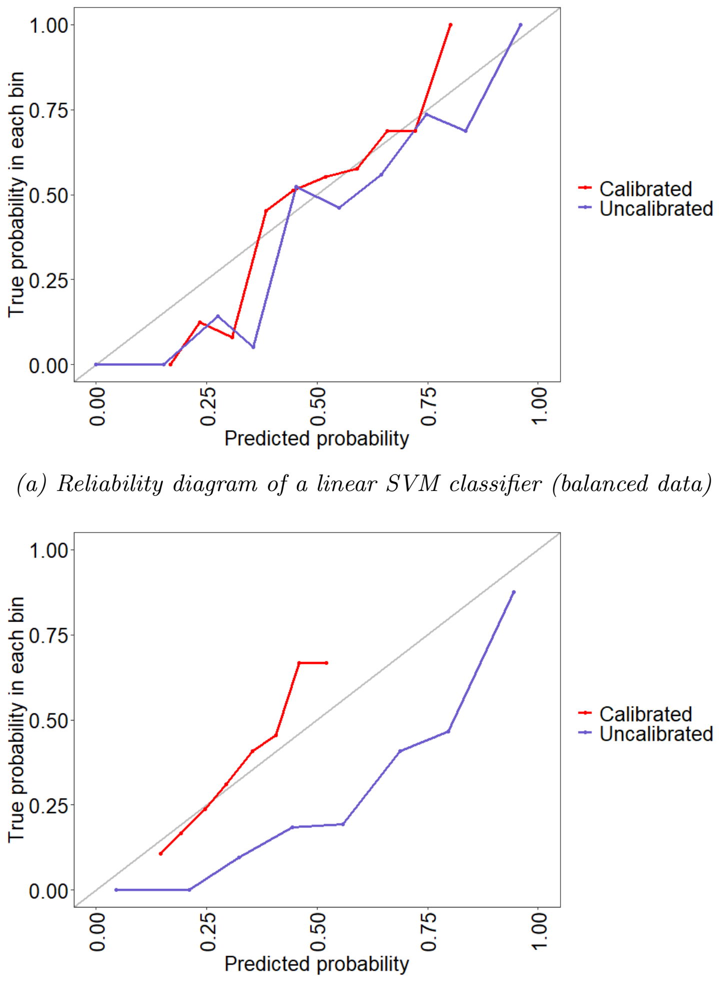
In both subfigures, the red line representing the calibrated probabilities shows a better convergence to the diagonal line than does the purple line. To produce calibrated probabilities, we generally apply a calibration function (or a *calibrator*) to the scores $s(\mathbf{x})$ so that it returns a set of calibrated probability between 0 and 1 [Kull et al., 2017; Pedregosa et al., 2020a]. The calibrated probabilities in subfigures 2.6a and 2.6b were generated by a calibration method called *Platt scaling*, which uses the sigmoid function as the calibrator. For this reason, the Platt scaling is also called *logistic calibration* [Kull et al., 2017; Pedregosa et al., 2020b]. Platt scaling was originally introduced as a method for obtaining probabilistic outputs from SVM. The intuition behind Platt scaling is to interpret the SVM scores as the log-odds defined in Eq.(2.14) and thereafter map the scores into probabilities using the following sigmoid function [Murphy, 2012; Platt, 2000]

$$\Pr(y=1|\hat{f}(\mathbf{x}))=\frac{1}{1+\exp(A\hat{f}(\mathbf{x})+B)} \tag{2.65}$$

where $\hat{f}(\mathbf{x})$ is the output of SVM, while A and B are parameters to the sigmoid function. According

[16] The convention is to discretise the predicted probabilities into 10 bins, that is, the observations with scores (or predicted probabilities) $0\leq s<0.1$ form the first bin, and those with scores $0.1\leq s<0.2$ form the second bin, and so on [Fernández et al., 2018; Niculescu-Mizil & Caruana, 2005]. The reliability diagrams in this study, however, depart from this convention and were plotted on the basis of different numbers of bins. There are two reasons for that: 1) the `evalm()` function from the `MLeval` package, the function I used to generate reliability plots, does not support the discretisation I mentioned above and is only able to divide scores into evenly sized bins; and 2) data sets used in this study are mainly data with imbalanced classes, meaning that the probabilities predicted by the classifiers included in this study would be concentrated below 0.5 (more about this in the following chapters). The second reason also implies that the calibration curve of a classifier trained on imbalanced data would be relatively "short". Therefore, discretising such predicted probabilities with 10 bins would result in a short calibration curve that fluctuate largely between 0 and 1. Such reliability diagram is misleading in my opinion. And to avoid this outcome, I used less than 10 bins to plot the reliability diagrams for classifiers trained on imbalanced data.

to Niculescu-Mizil & Caruana [2005] and Kull et al. [2017], the sigmoid transformation above can also be applied to other classifiers, such as boosted trees and Naive Bayes. In such cases, the term $\hat{f}(\mathbf{x})$ in Eq.(2.65) above will be replaced with scores $s(\mathbf{x})$ that correspond to the outputs of these classifiers. In addition, Kull et al. [2017] has also shown that the sigmoid function is exactly the right calibrator to use, assuming that the scores produced by a classifier are Gaussian distributed within each class around the class means and with equal variance.



(a) Reliability diagram of a linear SVM classifier (balanced data)

(b) Reliability diagram of a linear SVM classifier (imbalanced data)

Figure 2.6: Reliability diagrams showing the out-of-sample performance of a linear SVM. In both subfigures (a) and (b), the purple line is the uncalibrated probabilistic predictions computed by normalising the scores of the linear SVM, while the red line is the calibrated probabilistic predictions produced by Platt scaling. The datasets used to obtain the diagrams were both simulated from a logistic model. The only thing that these two simulated data sets differ is the balance of classes: in subfigure (a), the data set used has 50 percent of its observations belong to the positive class; in subfigure (b), only 25 percent of the observations in the data set are positive instances. Additionally, both datasets contains 500 observations, but they both were split to a training and a test set using the 50:50 ratio.

In practice, using Platt scaling to obtain probabilistic outputs from SVM means that we first train a SVM for our data, and then fit the sigmoid function defined in Eq.(2.65) to the SVM output. More formally, let $\hat{f}_i = \hat{f}(\mathbf{x}_i)$ denote the SVM outputs, which together with the outcomes y_i in the data on which the SVM is trained form the training set $(\hat{f}_i, y_i)$. Note that the outcome variable in this training set is still labelled as $y_i \in \{-1, 1\}$. Because our objective is to obtain probabilities, we replace the outcome variable y_i with a variable for target probabilities, t_i, which defined as [Platt, 2000]

$$t_i = \frac{y_i + 1}{2}. \tag{2.66}$$

It is not difficult to show that $t_i = 1$ when $y_i = 1$ and $t_i = 0$ when $y_i = -1$. We can therefore proceed to define a new training set $(\hat{f}_i, t_i)$, from which we find the logistic parameters A and B in Eq.(2.65) above using MLE. More specifically, we find the value of A and B that minimise the following function, which is the negative log likelihood of the training set $(\hat{f}_i, t_i)$ [Niculescu-Mizil & Caruana, 2005; Platt, 2000]

$$\underset{A,B}{\arg\min} \left(-\sum_i t_i \log(p_i) + (1 - t_i) \log(1 - p_i) \right) \tag{2.67}$$

where

$$p_i = \frac{1}{1 + \exp(A\hat{f}_i + B)}. \tag{2.68}$$

Note that the parameters A and B in the equation above must be estimated in a separate validation set. Since if we estimate (A, B) in the same data set we used to train the SVM, we will have a severe overfitting problem [Murphy, 2012; Platt, 2000]. The solution suggested by Platt [2000] is to apply a k-fold cross-validation in the training set[17]. In particular, we first split the training data into k evenly sized folds and then keep the first fold for validation of the SVM scores $\hat{f}_i$ (i.e., to evaluate the classification performance of the classifier), while train the SVM on the remaining $k - 1$ folds. After repeating this procedure k times so that each of the k folds had been treated as validation set once, we form the training set $(\hat{f}_i, t_i)$ for the estimation of parameters (A, B) by calculating the union of the k sets of $\hat{f}_i$ [James et al., 2013; Platt, 2000]. As a sidenote, it is also worth to point out that we must have $A < 0$ to guarantee that Eq.(2.65) is a monotonically non-decreasing function. This means that we also assume the observations with higher SVM scores have a higher probability of belonging to the positive class [Kull et al., 2017; Platt, 2000].

However, as a method of obtaining calibrated posterior probabilities from SVM, Platt scaling does have some drawbacks. Firstly, according to Niculescu-Mizil & Caruana [2005], Platt scaling is most effective when the distortion in the predicted probabilities[18] have a shape similar to the sigmoid function. If this is not the case, such as the uncalibrated probabilities shown in subfigure 2.6a, the improvements provided by Platt scaling will be minimal. Furthermore, if the shape of the uncalibrated probabilities differ considerably from that of the sigmoid, the Platt scaling might even produce a set of "calibrated" probabilities that are more ill-fitted to the true probabilities than the uncalibrated one [Kull et al., 2017].

Secondly, as mentioned above, the sigmoid function is the right calibrator to use if the scores within each class are Gaussian distributed with equal variance[19]. This would imply that Platt scaling might deliver suboptimal probabilities if the SVM was trained on imbalanced data – a circumstance where the assumption of equal variance is violated [Kull et al., 2017; Pedregosa et al., 2020a]. As shown in subfigure 2.6b, the Platt scaling had failed to provide probability estimates above the 0.5 level.

Thirdly, according to Tipping [2001], Platt's transformation of SVM score could also be poorly calibrated because there are nothing in the SVM that could justify the interpretation of the SVM scores

[17] Platt [2000] sets the number of folds to $k = 3$ in his article about Platt scaling.

[18] That is, the error between the uncalibrated probability estimates and the fraction of positive instances observed in the test set.

[19] Or at least in those cases where the ratio of scores distribution on the positive class to that on the negative class "behaves similarly to the ratio of Gaussians with equal variance" [Kull et al., 2017, p.5058].

as log-odds. For this reason, Tipping [2001] developed the *relevance vector machine* (RVM), a method based on Bayesian learning framework and has same functional form as SVM. In other words, the RVM can be treated as a Bayesian treatment of Eq.(2.63). According to Xu et al. [2007], the RVM not only produces probabilistic outputs, but also has nearly equal training efficiency and classification accuracy as the SVM. Since the RVM involves Bayesian statistics, which is out of the scope of this thesis, I will only use the Platt scaling to obtain probabilistic outputs for SVM.

Chapter 3

Rare events bias

In statistical literature, the term *rare events* refers to those binary random variables that the outcome $Y = 1$ occurs much less frequent than $Y = 0$. In other words, the binary data is heavily skewed towards one class – the class of $Y = 0$ or non-events [King & Zeng, 2001b]. This type of rareness is also called *relative rareness*. There is another type of rareness – *absolute rareness* – which is basically a problem induced by small sample [Van der Paal, 2014]. The small sample problem in binary modelling is presented when our sample consists fewer than 200 observations. In this case, fitting a logistic model to the data will yield biased parameter estimates [King & Zeng, 2001b]. Likewise, the presence of rare events can also cause such biased estimates. We describe this *rare event bias* in more detail in the following section.

3.1 Problem description

In order to understand the rare event bias intuitively, consider the following bivariate logistic model [King & Zeng, 2001b]

$$\Pr\left(Y = 1|X; \hat{\theta}\right) = \hat{p}(X; \hat{\theta}) = \frac{\exp\left(\hat{\theta}^T X\right)}{1 + \exp\left(\hat{\theta}^T X\right)} \tag{3.1}$$

where $\hat{\theta} = (\hat{\theta}_0, \hat{\theta}_1)$ is the MLE of the logistic parameter $\theta = (\theta_0, \theta_1)$. It would be convenient to express them using the notation of weights and intercept, i.e. $\hat{\theta}_0 = \hat{b}$ and $\hat{\theta}_1 = \hat{w}_1$. We further assume that the predictor X is positively correlated to the outcome, i.e. $w > 0$. The conditional densities[20] $\pi_{X|Y}(X|Y = 0)$ and $\pi_{X|Y}(X|Y = 1)$, respectively, will appear in the way illustrated in Figure 3.1 below. Since we have $w > 0$, the conditional density $\pi_{X|Y}(X|Y = 1)$ is located to the right of $\pi_{X|Y}(X|Y = 0)$[21] [King & Zeng, 2001b].

In binary classification, the ideal decision boundary should be set somewhere between the the right tail of the conditional density $\pi_{X|Y}(X|Y = 0)$ and the left tail of $\pi_{X|Y}(X|Y = 1)$. If rare events are presented, then the estimation of the conditional density $\pi_{X|Y}(X|Y = 1)$ – hence the left tail of it – would be a difficult task for the reason that such estimation is based upon only a few observed events. On the contrary, the estimation of $\pi_{X|Y}(X|Y = 0)$ and its right tail would be relatively easy because of the abundance of non-events. As a consequence, the decision boundary would be set in the vicinity of those

[20] For readers that are not familiar with probability theory, the conditional density (or more generally, the *conditional probability distribution*) of X given Y is the probability distribution of X, a continuous random variable, given that we have observed $Y = y$. A term related to the conditional density is *joint probability distribution*, which is the probability distribution of observing a certain pair of values for the random variables X and Y, denoted as $\pi_{X,Y}(X = x, Y = y)$ [Wasserman, 2004].

[21] This is because X and Y is positively correlated, meaning that increase in X will increase $\Pr(Y = 1)$. Since the observations are ordered in the X-axis, the conditional density $\pi_{X|Y}(X|Y = 1)$ must be to the right of $\pi_{X|Y}(X|Y = 0)$.

observed events with largest or second largest value of X (we can see this phenomenon in both Figure 3.1 and Figure 3.2), causing the probability of event occurrence, $\Pr(Y=1|X)$, to be systematically underestimated[22] [King & Zeng, 2001a, 2001b].

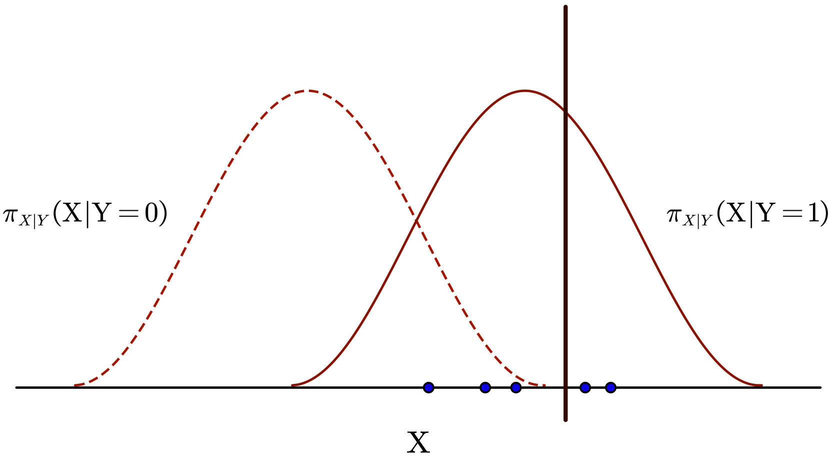

$\pi_{X|Y}(\mathrm{X}|\mathrm{Y}=0)$
$\pi_{X|Y}(\mathrm{X}|\mathrm{Y}=1)$
X

Figure 3.1: An illustration of rare event bias. The dotted red line is the conditional density $\pi_{X|Y}(X|Y=0)$, and the solid red line is the conditional density $\pi_{X|Y}(X|Y=1)$; the blue dots lying on the x-axis are the observations with outcomes $Y=1$, i.e. events that occurred; the vertical black solid line illustrates the decision boundary in Figure (3.2)

Regarding the computation of the probability defined in Eq.(3.1), the presence of rare events would result in: 1) that the $\hat{\theta}=(\hat{w},\hat{b})$ become biased estimates of the parameters of the true model $\theta=(w,b)$; and 2) that even if $\hat{\theta}=(\hat{w},\hat{b})$ are unbiased, the probability computed on the basis of such MLE, $\Pr(Y=1|\hat{\theta})$, would still be an inferior estimator of the true $\Pr(Y=1|\theta)$ [King & Zeng, 2001b].

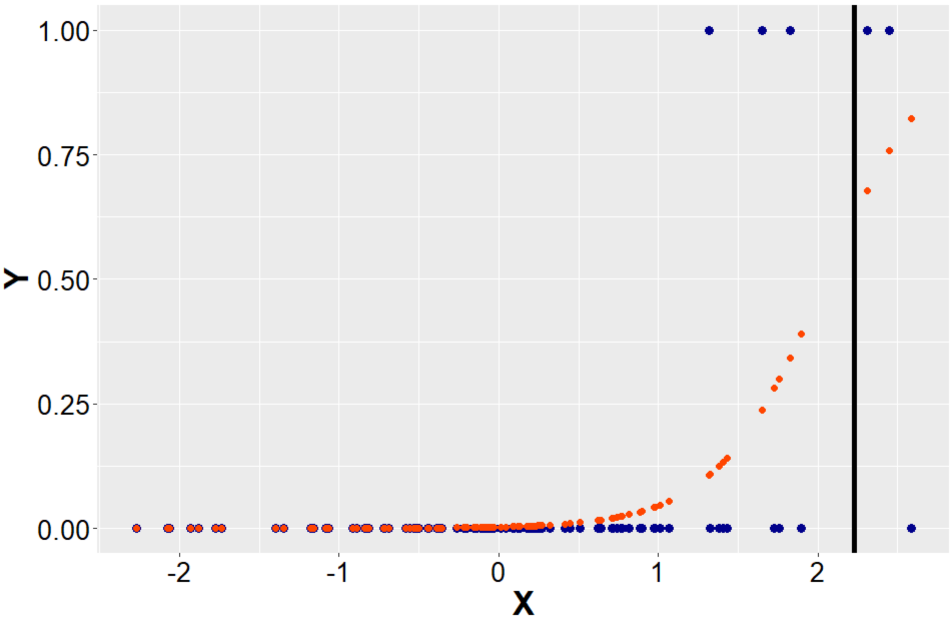


Figure 3.2: An illustration of rare event bias from another perspective. The blue dots are data of size n=100 simulated from a logistic model; the orange dots are the predicted values generated by fitting a LR to the simulated data; the solid black line is the decision boundary with the threshold set to $\tau=0$, or equivalently, $sigm(X;\theta)=0.5$. The overlap between the data and the predicted values in left bottom corner shows that estimation of $\pi_{X|Y}(X|Y=0)$ is easier and more accurate than estimation of $\pi_{X|Y}(X|Y=1)$.

The first kind of problem, i.e., the bias in the MLE, can be expressed using the general definition of

[22] Note that the systematic *underestimation* of $\Pr(Y=1|X)$ is equivalent to the systematic *overestimation* of $\Pr(Y=0|X)$ [King & Zeng, 2001b].

bias [Wackerly et al., 2008]

$$bias(\hat{\theta}) = \mathbb{E}[\hat{\theta} - \theta] = \mathbb{E}[\hat{\theta}] - \theta. \tag{3.2}$$

According to King & Zeng [2001b], the bias term defined above tends to directly affect $\hat{b}$, the estimate of the intercept parameter, which then passes the effect on to other parameters in the logistic model. For this reason, the bias correction purposed by King & Zeng – as we can see in next section – concerns mainly the correction of $bias(\hat{b})$.

The second kind of problem is induced by the uncertainties in the estimation of true θ in general. Such uncertainty can be caused by – for instance – sampling errors, which are then then amplified by the presence rare events. As a consequence, even if the MLE of θ is corrected for possible bias (King & Zeng denote such bias-corrected MLE as $\tilde{\theta}$), the probability function based upon such estimators, $\tilde{p}(\mathbf{x}_i; \tilde{\theta})$, will still be suboptimal [King & Zeng, 2001b]. More specifically, falling to take into account the uncertainty in estimation will result in the conditional distribution of the latent variable y^*, which is defined in Eq.(2.33) above, has smaller variance than what it should has if we incorporated the uncertainty into our estimation. This is illustrated in the Figure 3.3, where the right tail of the distribution of y^* with uncertainty (the blue area) is much heavier than the one without uncertainty (the red area marked with $\Pr(Y = 1|\tilde{\theta})$). Since the area to the right of the decision boundary (dotted line located at $y^* = \tau = 0$) is the probability of $Y = 1$, we can conclude that ignoring the uncertainty in the estimation would systematically underestimate the probability that rare events occur [King & Zeng, 2001b].

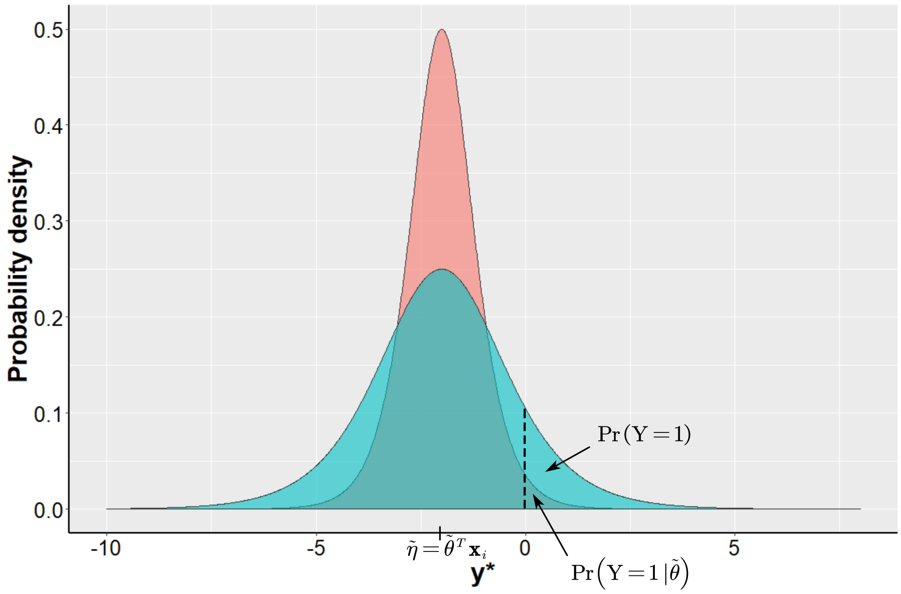


Figure 3.3: An illustration showing the uncertainty in estimation. The red area is the estimated probability using the unbiased parameter estimates $\tilde{\theta}$, while the blue area is the true probability.

3.2 Conventional solutions

3.2.1 ReLogit

As mentioned above, King & Zeng [2001b] found that the rare event bias affects the intercept directly. Therefore, one correction strategy is to first estimate the bias in $\hat{b}$, expressed as $\mathbb{E}[\hat{b} - b]$, and then subtract the estimated bias term from $\hat{b}$. Successful application of this strategy will give us the bias-corrected estimate $\tilde{b} = \hat{b} - bias(\hat{b})$. King & Zeng [2001b] have also shown that $bias(\hat{b})$ can be approximated as

$$bias(\hat{b}) = \mathbb{E}[\hat{b} - b] \approx \frac{\bar{p} - 0.5}{n\bar{p}(1-\bar{p})} \tag{3.3}$$

where $\bar{p} = 1/n \sum_{i=1}^{n} p_i$. Because of the presence of rare events, we have an average probability $\bar{p} < 0.5$, hence must Eq.(3.3) be negative. A negative bias means that the MLE $\hat{b}$ is too small. As a consequence, logistic models that use such intercept estimate tend to underestimate the true $\Pr(Y = 1)$. This is consistent with the theory mentioned in previously section [King & Zeng, 2001b]. Furthermore, Eq.(3.3) also implies that increase in sample size n reduces the bias. This seems to support McCullagh & Nelder's [1989] claim that the bias would become negligible with a larger sample size. However, the presence of $\bar{p}(1-\bar{p})$ shows the bias-reducing effect of larger sample diminishes as $\bar{p} \to 0$, i.e. as the event becomes more rare [King & Zeng, 2001b].

Concerning the issue of underestimation caused by estimation uncertainty, King & Zeng [2001b] proposed a Bayesian solution, which eliminates uncertainty by computing the following integral[23]

$$\Pr\left(Y_i = 1\right) = \int \Pr\left(\mathrm{Y}_i = 1|\theta\right)\pi_\theta\left(\theta\right)d\theta \tag{3.4}$$

where $\pi_\theta(\theta)$ is the density of parameter θ. We could also interpret Eq.(3.4) as the expected value of $\tilde{p} = \Pr(Y_i = 1|\tilde{\theta})$ by replacing θ with $\tilde{\theta}$[24]. Although Eq.(3.4) above can be computed using iterative simulations, King & Zeng [2001b] showed that it can also be approximated as

$$\Pr\left(\mathrm{Y}_i = 1\right) \approx \tilde{p}_i + C_i \tag{3.5}$$

where $\tilde{p}_i$ is the probability computed from the bias-corrected estimators $\tilde{\theta}$ and C_i is a correction term, which is defined as

$$C_i = \left(0.5 - \tilde{p}_i\right)\tilde{p}_i\left(1 - \tilde{p}_i\right)\mathbf{x}_i \mathrm{Var}(\tilde{\theta})\mathbf{x}_i^T \tag{3.6}$$

In Eq.(3.6) above, the estimation uncertainty in $\tilde{\theta}$ is captured by the variance of $\tilde{\theta}$. We can therefore view King & Zeng's [2001b] correction for estimation uncertainty as a methods that adds more variance to the probability model $\tilde{p}_i(\mathbf{x}_i; \tilde{\theta})$. This correction is also consistent with what it is shown in Figure 3.3. According to King & Zeng [2001b], as an estimator of the true p, $\tilde{p}_i + C_i$ is superior to both $\tilde{p}_i$ and $\hat{p}_i$ in the sense that $\tilde{p}_i + C_i$ has the smallest *mean square error* (MSE), which is defined as [Wackerly et al., 2008]

$$\mathrm{MSE}(\hat{\theta}) = \mathbb{E}\left[(\hat{\theta} - \theta)^2\right] = \mathrm{Var}(\hat{\theta}) + bias(\hat{\theta})^2. \tag{3.7}$$

However, as Eq.(3.7) suggests, an estimator that minimises the MSE is not necessarily unbiased. This also applies to $\tilde{p}_i + C_i$. According to King & Zeng [2001b], if we choose to interpret Eq.(3.4) as the expected value of $\tilde{p} = \Pr(Y_i = 1|\tilde{\theta})$, as mention above, the result will be $\mathbb{E}[\tilde{p}_i] = p_i + C_i$, meaning that the correction term C_i now become the bias between $\tilde{p}$ and the true p. In this scenario, the bias-corrected estimator would be $\tilde{p}_i - C_i$. We now have two superior estimators with different characteristics: $\tilde{p}_i + C_i$

[23] Using integration to alleviate estimation uncertainty is common in Bayesian statistics, see for instance Wackerly et al. [2008, Sect. 16]

[24] Recall that for a continuous random variable Y, the expected value of a function of Y is defined as an integral [Wackerly et al., 2008]. Since $\Pr(Y_i = 1|\tilde{\theta})$ can be treated as a function of $\tilde{\theta}$, we have

$$\mathbb{E}[g(\tilde{\theta})] = \int_{-\infty}^{\infty} g(\tilde{\theta})\pi_{\tilde{\theta}}(\tilde{\theta})d\tilde{\theta}$$

where $\pi(\tilde{\theta})$ is the probability density function of $\tilde{\theta}$.

that is superior in terms of MSE, and $\tilde{p}_i - C_i$ that is unbiased. In choosing between these two estimators, King & Zeng [2001b] suggested that we should use $\tilde{p}_i + C_i$ in the majority of cases.

3.2.2 Skewed link function

In Section 2.2.4, we mentioned that in LR, the probability p_i approaches 1 at the same rate as it approaches 0. This is because the symmetric form of the logistic density, as shown in Figure (2.4).

Given the special structure of the rare event data, i.e data that contains very few ones and many zeros, it is reasonable to model rare events with a GLM using link functions that are non-symmetrical and skewed towards 1 [Van der Paal, 2014]. One such candidate is the *complementary log-log* link, a non-symmetrical link with the characteristic of approaching 1 at a mush faster rate than the logit link do, as shown in the Figure 3.4 below. The complementary log-log link (henceforth referred as 'the Cloglog link') has the following formulation [Agresti, 2015; McCullagh & Nelder, 1989]

$$g(\mu_i) = \eta_i = \log\left(-\log\left(1 - p_i\right)\right) \tag{3.8}$$

which yields the following probability model

$$p(X) = 1 - \exp\left(-\exp\left(\sum_{j=1}^{h} w_j X_j + b\right)\right). \tag{3.9}$$

In Section 2.2.4, we discussed the latent variable formulation of the LR, showing that the logit link is tied to the logistic distribution. As for the GLM with Cloglog link, it can be shown that the Cloglog link is closely related to the *Type I extreme value distribution* (also known as *Gumbel* distribution) [Agresti, 2015]. Note that the cdf of the standard extreme value distribution is formulated as

$$\Pr(y_i = 1) = p_i = F_\epsilon\left(\eta_i\right) = \exp(-\exp(-\eta_i)). \tag{3.10}$$

By inverting the cdf above, we obtain

$$F_\epsilon^{-1}(p_i) = \eta_i = -\log\left(-\log\left(p_i\right)\right). \tag{3.11}$$

The expression at the right-hand side of Eq.(3.11) is known as the *log-log* link. We can see that this link function is reversely related to the Cloglog link defined in Eq.(3.8) above. Therefore, the GLM with Cloglog link can be viewed as the case when the error term of the latent variable y^* has a reverse extreme value distribution [Agresti, 2015; McCullagh & Nelder, 1989].

Concerning the interpretation of the parameter coefficient of the Cloglog model, Eq.(3.9) implies that, with $w_j > 0$, the complement probability $p(X_j + 1)$ is equal to the complement probability $p(X_j)$ raised to the $\exp(w_j)$ power [Agresti, 2015]. With some arrangements, we can express $\exp(w_j)$ as

$$\exp(w_j) = \frac{\log p(X_j + 1)}{\log p(X_j)}. \tag{3.12}$$

That is, the term $\exp(w_j)$ can be interpret as the relative increase in the log probability when X_j is increased by 1, given other predictors are fixed. Fitting a GLM with Cloglog link requires the use of Fisher scoring algorithm, a method slightly different from the Newton-Raphson algorithm used in fitting logistic models [Agresti, 2015].

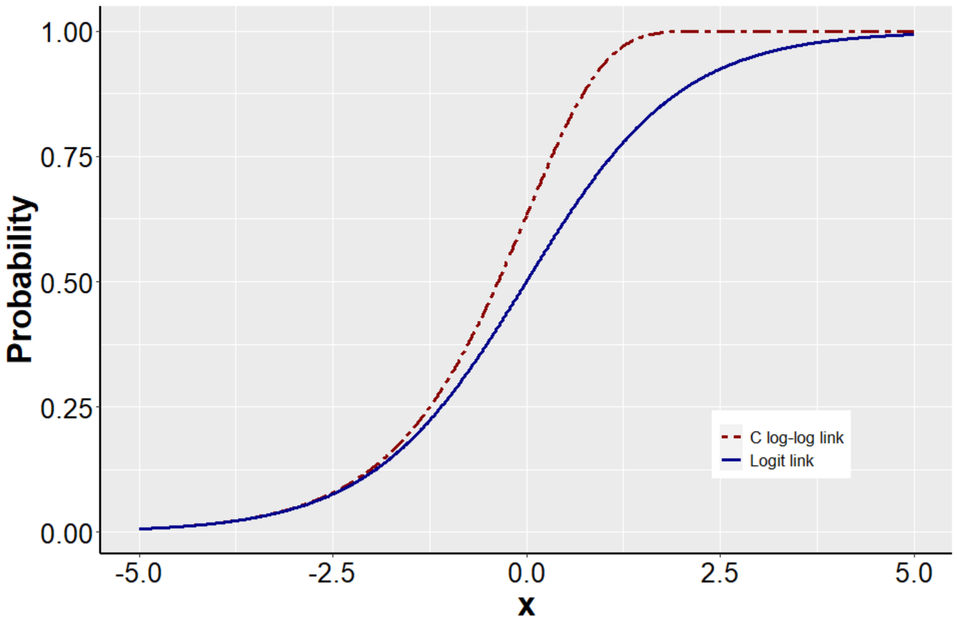


Figure 3.4: Comparison of the logit link and complementary log-log link

3.2.3 Cost-sensitive SVM

In contrast to other classifiers, SVM is expected to have an inherent advantage in handling imbalanced data. This advantage is originated from the Lagrange multipliers α_i. Recall that according to Eq.(2.63), the optimal decision boundary is determined by the function $\hat{f}(\mathbf{x}; \hat{W}, \hat{b})$, in which the weight parameter $\hat{W}$ is replaced by α_i. Because of the constraint $\sum_{i=1}^{n} \alpha_i y_i = 0$ in the optimisation problem in Eq.(2.50), the magnitude of those α_i associated with training observations coming from the minority class must be larger than that of those associated with observations from the majority class. As a result, support vectors from the minority class are given more weights than those from the majority class in our computation of the optimal decision boundary [Akbani et al., 2004; Batuwita & Palade, 2013].

However, several studies have reported that running SVM on imbalanced or rare event data would still yield suboptimal decision boundaries, despite the method's inherent weighting characteristics described above [Akbani et al., 2004; Batuwita & Palade, 2010; Veropoulos et al., 1999]. Such outcome occurs because of the tuning parameter C, which also can be interpreted as the penalty for misclassification, assigns the same cost to the classification errors in both the minority and the majority class. Recall that the optimisation problem in Eq.(2.54) has two conflicting goals: maximise the margin and minimise the training error. In an imbalanced training set, observations from the majority class are more dense than those from the minority class around the class boundary region. Therefore, in the setting that misclassifications in both classes are penalised equally, prioritising the minimisation of classification errors in the majority class and ignoring the minority class would yield the best result [Batuwita & Palade, 2013; Fernández et al., 2018]. In addition, as shown in Figure 3.2, it is common that in rare event data, the minority event class overlaps with the majority non-event class. This could amplify the overrepresentation of the non-event class around the decision boundary, leading to a decision boundary that favours the non-events even more [Fernández et al., 2018]. For these reasons, if rare events are presented in the training set, it is possible that the standard SVM results in a model that classifies all observations as non-events [Akbani et al., 2004].

One way to avoid such suboptimal decision boundaries is to construct a *cost-sensitive SVM* with *different error costs* (DEC) [Batuwita & Palade, 2013; Fernández et al., 2018]. This means that we use two separate misclassification costs for each of the classes. Denoting the misclassification cost for the positive class as C^+ and that for the negative class as C^-, we can modify Eq.(2.53) to

$$\begin{aligned} \min_{W,b,\xi_1,\ldots,\xi_n} \quad & \frac{1}{2}\|W\|^2 + C^+ \sum_{i|y_i=+1}^{n} \xi_i + C^- \sum_{i|y_i=-1}^{n} \xi_i \\ \text{subject to} \quad & y_i \left(\sum_{j=1}^{h} w_j x_{ij} + b \right) \geq 1 - \xi_i,\ i = 1, \ldots, n \\ & \xi_i \geq 0,\ i = 1, \ldots, n \end{aligned} \tag{3.13}$$

which corresponds to the following dual optimisation problem

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_i \alpha_k y_i y_k \langle \mathbf{x}_i, \mathbf{x}_k \rangle \\ \text{subject to} \quad & 0 \leq \alpha_i^+ \leq C^+ \ ,\ 0 \leq \alpha_i^- \leq C^- \text{ and } b = \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & i = 1, \ldots, n. \end{aligned} \tag{3.14}$$

where α_i^+ and α_i^- are the Lagrange multipliers or weights for the positive and negative class, respectively [Batuwita & Palade, 2013]. According to Akbani et al. [2004], a good role of thumb is to set misclassification costs according to the minority-to-majority ratio. That is, assuming the positive class is the minority class, the ratio C^+/C^- should be equal to the fraction of positive class to the negative class [Akbani et al., 2004; Batuwita & Palade, 2013].

3.3 Sampling methods for alleviating rare events bias

Although this thesis concerns the capability of different statistical methods (and their modifications) in handling rare event data, the reader should bear in mind that it is also possible to address the problems caused by the rare events using *data-level preprocessing*, or more specifically, *sampling methods*. In contrast to the modifications described in Section 3.2, which also known as the *internal methods*, the sampling methods are often termed as *external methods* [Batuwita & Palade, 2013]. The aim of such methods is to use various sampling techniques to generate a more balanced data set. These techniques includes: 1) *undersampling*, which means that we balance the data by creating a subset of the original data through randomly dropping observations that belong to the majority class; 2) *oversampling*, which means that we increase the observations in the minority class using replication or interpolation; and 3) *hybrids*, which combines the two sampling techniques above [Fernández et al., 2018]. Since sampling methods alter the structure of the original data, fitting a model to such altered data is often accompanied by adjustments in some of the parameter estimates [King & Zeng, 2001a, 2001b].

Methods for oversampling and undersampling are common in econometrics. The *choice-based sampling*, for instance, is a method of drawing independent random samples from the classes $Y = 1$ and $Y = 0$ separately. Another example is the *case-cohort study*, a method of collecting all ones in the data set and randomly selecting zeros [King & Zeng, 2001a]. In the case of rare events, the case-cohort format is equivalent to randomly dropping the non-events while keeping all the events [King & Zeng, 2001a], i.e. a version of undersampling methods. According to King & Zeng [2001a, 2001b], if we fit a logistic regression model to such undersampled data, we have to adjust for the intercept estimate[25], $\hat{b}$. They therefore proposed the following adjustment

$$\hat{b} - \log\left(\left(\frac{1-\kappa}{\kappa}\right)\left(\frac{\bar{y}}{1-\bar{y}}\right)\right) \tag{3.15}$$

where κ is fraction of events in the population, and $\bar{y}$ is the fraction of events (or *rarity*) in the sample. As a rule of thumb, King & Zeng [2001a] suggested that the number of non-events should not exceed 2/3 or 5/6 of the data set.

Synthetic minority oversampling technique (SMOTE) is another popular sampling method [Chawla et al., 2002; Fernández et al., 2018]. In contrast to the conventional oversampling technique, i.e. oversampling with replacement, SMOTE oversamples by generating synthetic data that assigned to the minority class. The procedure of generating new data involves the use of *k-nearest neighbours* (KNN) algorithm and can be informally described as followed. Before running SMOTE, we prespecify the number of observation that have to be generated for a balanced data. The SMOTE-algorithm then randomly select an observation $\mathbf{x}_i$ from the minority class, and compute the KNN of that observation in the training set[26]. After that, the algorithm proceeds by calculating the distances between $\mathbf{x}_i$ and its KNN. New data points are then generated within these distances. This part of generation procedure is done by multiplying each of the k distances by a random number that ranges from 0 to 1. In sum, for a single observation $\mathbf{x}_i$, k synthetic data points are generated. The algorithm then repeats itself and stops when the number of the synthetic observations equals the number we prespecified [Chawla et al., 2002; Fernández et al., 2018]. To achieve the best possible classification performance, Chawla et al. [2002], the developers of SMOTE, suggest a combination of SMOTE and undersampling of the majority class.

[25] The fact that we only need to adjust $\hat{b}$ is because that, for a logistic model, the parameter estimates associated with the predictors, $\hat{w}_1, \ldots, \hat{w}_h$, are statistically consistent estimates of $w_1, \ldots, w_h$. The adjustment is therefore not needed. Note that this does not apply to other models such as probit and linear regression [King & Zeng, 2001a]

[26] The SMOTE sets $k = 5$ by default, meaning that the five nearest neighbour of $\mathbf{x}_i$ is computed.

Chapter 4

Methodology

In this chapter, I present a full list of statistical methods evaluated in this study and explain how the evaluation was performed.

Generally speaking, the methodology of this thesis combines a simulation study with an empirical analysis. The purpose of the simulation study is to investigate how changes in rarity, number of predictors, and sample size would influence the predictive performance of the selected classifiers, while the aim of the empirical analysis is not only to investigate how different classifiers perform in real data, which seldom has the ideal distribution required by these methods (e.g. normality assumption), but also to add a practical dimension to the study. As Hand [2006] had pointed out, there is no single classifier that is universally preferable to the others. This is especially true in those cases where real data is used for comparison: depending on the underlying nature of the outcome variable, the cost of misclassification may differ from case to case. Therefore, the empirical analysis and the simulation study are of equal importance. The reader must not treat the former as merely a complement to the latter.

4.1 Statistical models and methods

The statistical methods evaluated in this study can be grouped into two families: the GLM family and the SVM family, as shown in Table 4.1 below. In general, the fitting procedure of the methods from the GLM family was performed in `R`, while models from the SVM family were fitted in `Python` using the `scikit-learn` package (Version 0.24.0)[27].

More specifically, among the methods from the GLM family, the LR and Cloglog were both fitted using the `glm()` function in `Base R`, while the ReLogit model was fitted using the `zelig()` function provided by the `Zelig` or `zeligverse` package.

As for the methods from the SVM family, the training and calibration procedures were performed using the functions `SVC`, `GridSearchCV`, and `CalibratedClassifierCV` combined. The function `SVC` was used to define the SVM model. By specifying the `kernel` parameter in the `SVC` function differently, I constructed four SVMs with different kernels. The cost-sensitive versions of SVM were obtained by activating the option `class_weight = "balanced"`, which automatically adjust weights inversely proportional to the class frequency [Pedregosa et al., 2020b]. Both the standard and the cost-sensitive SVMs were trained using the function `GridSearchCV`, which also performed a 5-fold cross-validation simultaneously

[27] The reasons behind my choice of dividing the fitting procedures of the GLMs and SVMs into two programming languages – instead of fitting both GLMs and SVMs using one of these languages alone – are twofold. Firstly, the function that implements the *ReLogit* model exists only in `R` and `STATA`. And secondly, fitting the SVM requires a huge amount of time. It therefore is more convenient to train a SVM in `Python`, since the `Python` IDE allows me to run multiple session in parallel (this feature is not available for the free version of RStudio), so that I can use the waiting time more effectively.

to find the optimal value of parameter C among the following five candidates[28]: $\{0.001, 0.1, 1, 5, 10, 50\}$. The scores of the four SVMs can be obtained by using the `decision_function` method (included in both `SVC` and `GridSearchCV`) after the model training was completed. Finally, I used the the function `CalibratedClassifierCV` to implement Platt scaling and obtain the calibrated probabilistic outputs for the SVMs. Unlike Platt [2000], who used a 3-fold cross-validation for obtaining parameters (A, B), I used a 5-fold cross-evaluation to obtain more accurate estimates for the parameters. As for the other parameters in `CalibratedClassifierCV`, the default setting was used. Running the function on its default setting assures that the Platt scaling, especially that part of the procedure concerning the estimation of parameters (A, B), is implemented in accordance with the description provided in Section 2.3.4.

Table 4.1: List of methods included in this study

	Name	Description
GLM	*LR*	Standard logistic regression (or GLM with logit link)
	ReLogit	Logistic regression for rare event, with correction developed by King & Zeng [2001a, 2001b]
	Cloglog	GLM with complementary log-log link
SVM	*Linear SVM*	Standard SVM with linear kernel
	Gaussian SVM	Standard SVM with Gaussian kernel
	Sigmoid SVM	Standard SVM with Sigmoid kernel
	Polynomial SVM	Standard SVM with third-degree polynomial kernel
	Linear SVM-DEC	The cost-sensitive version of *Linear SVM*
	Gaussian SVM-DEC	The cost-sensitive version of *Gaussian SVM*
	Sigmoid SVM-DEC	The cost-sensitive version of *Sigmoid SVM*
	Polynomial SVM-DEC	The cost-sensitive version of *Polynomial SVM*

4.2 Simulation study

4.2.1 Experiment setup

Data sets used in the simulation study were generated in `R`[29]. These synthetic data sets vary in three aspects: sample size, rarity (i.e., the fraction of 1 in the data set) and dimensions (i.e., the number of predictors). Since both sample size and the fraction of events can incur rare event bias, as noted in Section 3.1, it is natural to include both these two aspects into the study. As for the dimensional aspect, the point here is to investigate whether the rare events bias can be amplified by increased model complexity. Additionally, all predictors in the simulated data sets are continuous, Gaussian distributed random variables with different means and variances. To generate the predictors, I first specified a covariance matrix by randomly generating numbers from a uniform distribution and then use these numbers as the covariances between different predictors. The means of different predictors are randomly selected integers. I then used the vector of means and the covariance matrix to construct a design matrix

[28] The number of candidates was limited to 5 because training a SVM classifier on large data sets is extremely time-consuming, not to mention performing a 5-fold cross-validation at the same time.

[29] Codes and data files for replication can be accessed in this link: https://github.com/LukasHMa/bsc_thesis_in_statistics

using the `mvrnorm()` function from the `MASS` package. By using `mvrnorm()`, I ensured that all predictors are Gaussian distributed random variables.

After generating the predictors, I proceed to the simulation of rare events data. For this task, I followed King & Zeng's [2001a, 2001b] approach, by using the logistic model as the true model and testing different values of the intercept b to control the fraction of 1 in the data set. The levels of rarity presented in the simulated data sets can be informally described as: *moderate* ($\bar{y} \approx 16\%$), *severe* ($\bar{y} \approx 3.4\%$), and *extreme* ($\bar{y} = 0.8\%$). While the values of the intercept are prespecified numbers, the vector of parameters associated with different predictors, θ, was generated randomly from a Gaussian distribution. The linear combination $\theta^T X$, once being specified, was transformed to probabilities using the sigmoid function. Finally, to generate the outcome variable y, I first generated a uniformly distributed random number ranging from 0 to 1, and then compared it with the probabilities generated from the logistic model. In this setting, if the random number is smaller than or equal to the probability generated from the true model, then it will be coded as $y = 0$, and $y = 1$ otherwise. A rare events data is generated after completing all these steps.

A total of seven distinctive synthetic data sets were generated. These data sets can be grouped into three different scenarios based on their size and the number of predictors used to create them. The characteristics of the seven simulated data sets are shown in Table 4.1 below.

Table 4.2: Charateristics of simulated data

	Name	Intercept	Rarity (*Full*)	Rarity (*Train*)	Rarity (*Test*)	Obs	Predictors
Scenario I	`small_a`	-3.6	16%	16.19%	15.56%	150	5
	`small_b`	-6.08	3.33%	3.77%	2.27%	150	5
Scenario II	`med_a`	-3	16.2%	16.29%	16.00%	500	10
	`med_b`	-5	3.4%	3.43%	3.33%	500	10
	`med_c`	-7	0.8%	0.857%	0.666%	500	10
Scenario III	`large_a`	-3.32	16.2%	16.14%	16.33%	1000	20
	`large_b`	-5	3.5%	3.43%	3.67%	1000	20

In the subsequent analysis, the data was divided into a training set and a test set. Dividing data in this way would enable us to evaluate both the *in-sample* and *out-of-sample* classification performance of different methods. I decided to reserve 70 percent of the observations for the training set, while the remaining 30 percent were used as testing. Considering the special characteristic of rare events data, the 70:30 ratio is the most appropriate choice in order to ensure that some observations from the event class were included in the test set. Additionally, to ensure that the training set has the structure similar to the original data set, I applied *stratified sampling* when selecting observations for the training set. Generally speaking, stratified sampling is a method that selects samples by first dividing the original data into several mutually exclusive, exhaustive strata – usually performed on the basis of certain categorical variables in the original data – and then drawing random samples within each of these strata [Levy & Lemeshow, 2008]. Since I only have continuous predictors in my simulated data sets, the stratification in my case was performed on the basis of the outcome variable. This means that the events and non-events in the training set were sampled simultaneously but independently from each other, as in the case of case-control studies. After the sampling for the training set is completed, the remaining observations form the test set.

4.3 Empirical analysis

4.3.1 Data

Table 4.3: Variables in the MID data

Variables	Descriptions
Militarised dispute	$y = 1$ if the dyad (pair of countries) was engaged in a militarised interstate dispute
Contiguous	$y = 1$ if the pair of countries are neighbours
Allies	$y = 1$ if the pair of countries are allies
Foreign policy	Score measuring similarity between two countries in their foreign policy portfolio
Balance of Power	Score measuring the balance of the military power between two countries
Max. democracy	The maximum degree of democracy in the pair of countries, indicating which of the two countries has the higher degree of democracy
Min. democracy	The minimum degree of democracy in the pair of countries, indicating which of the two countries has the lower degree of democracy
Max. trade	The maximum degree of trade dependency in the pair of countries, indicating which of the two countries has the higher degree of trade dependency
Min. trade	The minimum degree of trade dependency in the pair of countries, indicating which of the two countries has the lower degree of trade dependency
Year since dispute	The number of years since the last dispute
Major Power	$y = 1$ if the pair of countries includes a major power

The data I used for the empirical analysis is a version of *Militarized Interstate Dispute Data*(MID) compiled by King & Zeng [2001a, 2001b]. In general, the MID data is associated with the Correlation of War project and has many versions. The version of MID data used in King & Zeng [2001a, 2001b] was based upon MID Version 3.0, which is presented in the non-directed dyad-year[30] format [Ghosn et al., 2004]. King & Zeng's version of MID consists 303,814 dyads (i.e., pair of countries) covering the period 1946–1992. The outcome variable is the event "militarised interstate dispute", which is coded as $y = 1$ if two particular states had engage in military conflict and $y = 0$ otherwise. A total of 1,042 military conflicts were recorded in this period, implying a rarity of 0.3423 percent. Other than variables that already existed in the original MID data, King & Zeng had also included variables that are typical for the research in peace and conflict studies, such as the balance of military power, degree of democracy and trade dependency etc. A full list of variables included in King & Zeng's [2001a] version of MID data is provided in Table 4.3[31].

[30] The use of dyad-year data is very popular in the quantitative analysis of international conflicts. The main difference between the non-directed dyad-year and the directed one is the latter codes the same military interstate dispute twice. In other words, if Country X and Country Y engaged in a military conflict in a certain year, then this conflict is recorded once in Country X, and once in Country Y. The non-directed dyad-year data does not make such distinction. For this reason, the directed dyad-year data could be very useful in studying the deliberate initiation of military conflict [Bennett & Stam, 2000].

[31] Variables in the original data set are unlabeled. The labels of the variables in Table 4.3 were recreated based on King & Zeng [2001a, 2001b]. There are more variables included in the data set than what Table 4.3 shows, but because King & Zeng [2001a, 2001b] did not used all variables in the original data (they had even created several new variables from the original data) in their analysis, there is no way for me to determine what these variables are. For this reason, the variables not included in King & Zeng [2001a, 2001b] were omitted from Table 4.3. The original data set can be obtained from Gary King's dataverse, through the following link: `https://doi.org/10.7910/DVN/RNSU7V`.

4.3.2 Initial preparation before analysis

As mentioned in Section 2.3 above, the fitting procedure of SVM is equivalent to solving a dual optimisation problem, and it is well documented in the statistical literature that such optimisation procedure is exceedingly time consuming (see for instance Platt [1998]; Tsang et al. [2005] and Pedregosa et al. [2020b]). With a data set consisting over 300,000 observations, the amount of time required for training a SVM would be substantial, not to mention the additional time needed for the subsequent tuning of the parameter C, which is performed using cross-validation. A reasonable choice is therefore to drop 90 percent of the observations in the MID data and only use the remaining 10 percent for the empirical analysis.

As in the case of the simulation study, I implemented stratified sampling to ensure that the 10 percent subset data contains the same statistical information, most importantly the event rarity, $\bar{y}$, as the full data. As showed in Table 4.4, the $\bar{y}$ in the 10 percent subset data is nearly identical to that in the full data. By comparing the Table 4.4 and Table A.1 in Appendix A, it could also be shown that the 10 percent subset data generally maintained the structure of the full data, indicating that the stratified sampling is a reliable strategy in creating subsets of the original data. For this reason, I implemented the stratified sampling once again in the train-test-split of the empirical data, using the same 70:30 ratio as in the case of the simulation study.

Table 4.4: Descriptive statistics for 10% of the MID data

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
Militarised dispute	30,381	0.0034	0.0584	0	0	0	1
Contiguous	30,381	0.0402	0.1963	0	0	0	1
Allies	30,381	0.0885	0.2840	0	0	0	1
Foreign policy	30,381	0.7728	0.1726	−0.0450	0.6667	0.9098	1.0000
Balance of Power	30,381	0.3633	0.2985	0.0002	0.0935	0.6011	1.0000
Max. democracy	30,381	3.2838	7.2816	−10	−6	10	10
Min. democracy	30,381	−5.0190	5.6587	−10	−9	−5	10
Max. trade	30,381	−4.8227	2.5771	−8.6840	−8.6840	−2.8145	0.2954
Min. trade	30,381	−5.7636	2.5249	−9.5361	−9.5361	−3.7720	−0.8995
Year since dispute	30,381	17.6006	11.5144	0	8	26	46
Major Power	30,381	0.0837	0.2769	0	0	0	1

4.4 Evaluation metrics

4.4.1 Conventional evaluation metrics for classification performance

When evaluating the classification performance of a classifier, a common approach is to summarise the classification result in a *confusion matrix*. Table 4.5 below shows a typical confusion matrix, where the rows shows the predicted class labels and the columns the true labels [Agresti, 2015; James et al., 2013].

Using the information provided by the confusion matrix, we can compute a number of different evaluation metrics. Nearly all evaluation metrics used in this study are derived from the confusion matrix. We can start with measures that are frequently used in the statistical literature. These include *sensitivity*, *specificity*, *receiver operating characteristic curve* (ROC) and *area under the ROC curve* (AUC).

Table 4.5: A confusion matrix

	Predicted label	
True label	True Positive	False Negative
	False Positive	True Negative

Sensitivity and specificity

Sensitivity, also known as *recall* and *true positive rate*, shows the prediction accuracy of the positive class. It is defined as the fraction of correctly classified ones to true ones (i.e., true positives plus false negatives), that is [Agresti, 2015]

$$\text{Sensitivity} = \frac{TP}{TP + FN} = \Pr(\hat{Y} = 1 | Y = 1) \tag{4.1}$$

On the contrary, specificity shows the prediction accuracy of the negative class and defined as the fraction of correctly classified zeros to true zeros (i.e., true negatives plus false positives), that is [Agresti, 2015]

$$\text{Specificity} = \frac{TN}{TN + FP} = \Pr(\hat{Y} = 0 | Y = 0) \tag{4.2}$$

ROC/AUC

In Section 2.2.4, we mentioned that we could obtain different classification results by altering the value of the threshold τ (or $\hat{p}_0$ if expressed as predicted probability). Hence, the sensitivity and specificity for a certain classifier might vary depending on the threshold which we selected. This variation can be visualised using a ROC curve, which displays the *true positive rate* (i.e., sensitivity) and *the false positive rate* (i.e., $1 -$ specificity, also known as *Type I error*) for all possible thresholds [James et al., 2013]. With the true positive rate on the y-axis and the the false positive rate on the x-axis, the perfect classification performance is illustrated by the top-left corner of the ROC space. A good classifier should therefore have its ROC curve bent towards this point as close as possible [Fernández et al., 2018]. In addition, when evaluating a certain classifier, we only consider those classifiers whose ROC curve lies above the upward diagonal line in the ROC space, since this line represents the expected performance of a ”no information” classifier, that is, a classifier that makes random predictions [James et al., 2013; Kelleher et al., 2020].

Other than evaluating visually, we can also quantify the results shown in the ROC curve by calculating the integral of the curve. This yields the area under the ROC curve (AUC), a numeric measure that ranges from 0 to 1. Note that since AUC $= 0.5$ corresponds to the upward diagonal line in the ROC space, a good classifier should therefore have a AUC score above this value [James et al., 2013] or, following from the recommendation of Kelleher et al. [2020], a AUC score above 0.7.

4.4.2 Evaluation metrics for imbalanced classification

According to Fernández et al. [2018], the conventional evaluation metrics for classification performance, such as the overall accuracy and the ROC/AUC, could be misleading under the presence of class imbalance and rare events. As mentioned in Section 3.1, the abundance of zeros in the data set makes the estimation of $\pi_{X|Y}(X|Y = 0)$ very easy. This would cause the overall classification accuracy and the ROC curve to be dominated by the success in classifying non-events, while the contribution of successful classifications of the event class to these two metrics is merely marginal. Since we are more interested in finding a classifier that can correctly identify the event class in rare event modelling, the use of overall accuracy and ROC curve as evaluation metrics would be inappropriate [Fernández et al., 2018].

When choosing the evaluation metrics for classification performance, it is useful to distinguish between metrics for nominal class prediction and for probabilistic prediction. The former evaluates a classifier by comparing the number of predicted class labels to the true labels, while the latter compares the probabilities generated from the classification model to the true outcomes [Fernández et al., 2018]. My choice of evaluation metrics in this thesis was based upon the recommendations of Fernández et al. [2018]. Among nominal metrics, Fernández et al. [2018] suggested the use of *precision* and *recall*,*F1-score* and *balanced accuracy* (BA); Among probabilistic metrics, they suggested the use of *Brier score* and *precision-recall curve*. Detailed descriptions about these metrics are provided below.

Precision and recall

Precision and recall can be calculated using the information provided by the confusion matrix. As mentioned above, recall is merely an alternative name for sensitivity and hence is defined as in Eq.(4.1). The precision is defined as the fraction of correctly classified ones to the total number of observations that predicted as one (i.e., true positives plus false positives), that is

$$\text{Precision} = \frac{TP}{TP + FP} = \Pr(Y = 1|\hat{Y} = 1). \tag{4.3}$$

As mentioned above, when modelling rare events data, we pay more attention to the classification rate of the positive event class than the negative non-event class. Metrics such as precision and recall contain only information about the positives, making them appropriate for our task [Fernández et al., 2018; James et al., 2013; Murphy, 2012].

F1 score

Other than precision and recall, we may also be interested in the trade-off between these two metrics, or more specifically, the trade-off between correctness (i.e., $\Pr(Y = 1|\hat{Y} = 1)$) and coverage (i.e. $\Pr(\hat{Y} = 1|Y = 1)$) in the prediction of ones [Fernández et al., 2018]. One way of evaluating this trade-off is to calculate the F_1 score, which is defined as

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

The F_1 score is also known as the harmonic mean of precision and recall. It has the range $[0, 1]$, with $F_1 = 1$ as the perfect trade-off between the two metrics. Therefore, a good classifier should have a F_1 score close to 1 [Fernández et al., 2018; Murphy, 2012].

Precision-recall curve

Trade-off between precision and recall can also be visualised graphically using the precision-recall curve. Similar to the ROC curve, the precision-recall curve illustrates the variations in precision and recall caused by our choice of thresholds by plotting the two measurements against each other, with the precision on the y-axis and the recall on the x-axis. The top-right corner of the plot symbolises the perfect trade-off between the two metrics. Therefore, a good classifier should have its precision-recall curve bent towards that corner [Fernández et al., 2018; Murphy, 2012]. Since the choice of threshold could only be meaningful when put into a context, the precision-recall curve is only presented in the empirical analysis.

Balanced accuracy

The balanced accuracy (BA) or *average class accuracy* is defined as the arithmetic mean of sensitivity and specificity, as shown in the following equation [Brodersen et al., 2010; Fernández et al., 2018]

$$BA = \frac{\text{Sensitivity} + \text{Specificity}}{2} = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{4.5}$$

In other words, the BA shows the average of prediction accuracy obtained on either class [Brodersen et al., 2010]. The BA is equivalent to the overall accuracy when the classes are balanced. Under class imbalance or rare events, the BA tends to be smaller than the overall accuracy. Because of its sensitivity to class imbalance, BA is considered a more appropriate measure than the overall accuracy in evaluating imbalanced classification [Brodersen et al., 2010; Fernández et al., 2018; Kelleher et al., 2020]. Similar to other measures for prediction accuracy, the BA has the range $[0, 1]$, with 1 as the perfect score. Note that only classifiers that achieved $BA > 0.5$ are worth to consider, since $BA = 0.5$ implies that the classifier evaluated always assigns the observations to one class [Kelleher et al., 2020].

Brier score

Brier score is a commonly used metric in evaluating the probabilistic output of a certain classifier. It is usually formulated as the MSE between the true outcome and the predicted probability [Fernández et al., 2018]

$$BS = \frac{1}{n}\sum_{i=1}^{n}(\hat{p}_i - y_i)^2 \tag{4.6}$$

where $\hat{p}_i$ is the predicted probability of the i-th observation. The MSE format also implies that the Brier score is a loss function, meaning that a good classifier should have a low Brier score [Fernández et al., 2018].

Chapter 5

Results from the simulation study

5.1 Scenario I

The classification performance of the GLMs and SVMs in this scenario are summarised in Table 5.1 and 5.2, respectively. Data sets in this scenario are characterised by a small sample size and a few number of predictors. The class imbalance of the data sets included in this scenario can be described as *moderate* ($\bar{y} = 16\%$) and *severe* ($\bar{y} = 3.33\%$).

As shown in Table 5.1 below, the GLMs using the alternative threshold $\hat{p}_0 = \bar{y}$ achieved a better in-sample performance in recall, F1 score, and BA than did those using the conventional threshold $\hat{p}_0 = 0.5$. As for the out-of-sample performance, the results are mixed. Additionally, at least in the `small_a` data set, the GLMs using the alternative threshold had a lower precision in general than did the GLMs using the conventional threshold.

Among the GLMs evaluated in the `small_a` data set, the ReLogit model using the conventional threshold had a better in-sample classification performance than did the other two GLMs that used the same threshold. However, in terms of out-of-sample prediction, the GLM with Cloglog link had the best result in all evaluation metrics except for the Brier score. In addition, the Cloglog model also performed better than the ReLogit when using the alternative threshold. One might be tempted to draw the conclusion that the Cloglog model is the best GLM in this data set, but such a conclusion is premature, considering the fact that all GLMs have the same performance in recall regardless of the threshold they used and that the out-of-sample Brier score of the Cloglog model is the highest among the three GLMs evaluated. Therefore, the results from Table 5.1 provide little evidence for the claim that the Cloglog model is better than other GLMs in a data set characterised by a small sample size and a moderate class imbalance.

As for the SVMs, Table 5.2 shows that all standard SVMs, except for the one with third-degree polynomial kernel, did not detect a single event in both the training and the test set. This is an unexpected result and seems to contradict the theoretical result in Akbani et al. [2004] about the inherent advantage of SVM over other classifiers[32] in handling imbalanced data. Even if the SVM with polynomial kernel – the only standard SVM that managed to detect some events in this case – achieved better precision (both in-sample and out-of-sample) than the GLMs, the classifier still had a much lower recall (or sensitivity) than its GLM counterparts.

The cost-sensitive SVMs performed much better than the standard versions. They all managed to

[32] The original sentence in Akbani et al. [2004, p.42] is "This [the inherent weighting through α] shows why SVM does not perform too badly compared to other machine learning algorithms for moderately skewed datasets". Since machine learning classifiers are more sophisticated and flexible than the traditional classification methods, I expected this also holds for the GLMs.

Table 5.1: Classification performance the GLMs (Scenario I)

	In-sample					*Out-of-sample*				
	Precision	Recall	F1	BA	Brier	Precision	Recall	F1	BA	Brier
Data: `small_a`										
LR	0.5833	0.4118	0.4828	0.6775	0.094748	0.3333	0.2857	0.3077	0.5902	0.167916
	(0.4286)	(0.8824)	(0.5769)	(0.8275)		**(0.2000)**	(0.4286)	**(0.2727)**	**(0.5564)**	
ReLogit	**0.6364**	0.4118	**0.5000**	**0.6832**	0.094953	0.3333	0.2857	0.3077	0.5902	**0.166617**
	(0.4167)	(0.8824)	(0.5660)	(0.8219)		(0.1765)	(0.4286)	(0.2500)	(0.5301)	
Cloglog	0.5833	0.4118	0.4828	0.6775	**0.094000**	**0.4000**	0.2857	**0.3333**	**0.6034**	0.172697
	(0.4412)	(0.8824)	**(0.5882)**	**(0.8332)**		**(0.2000)**	(0.4286)	**(0.2727)**	**(0.5564)**	
Data: `small_b`										
LR	0	0	0	0.5000	0.034375	0	0	0	0.4767	0.040913
	(0.1000)	(0.7500)	**(0.1765)**	**(0.7426)**		**(0.0476)**	(1)	**(0.0909)**	**(0.7674)**	
ReLogit	0	0	0	0.5000	0.035614	0	0	0	0.4767	**0.040754**
	(0.0678)	**(1)**	(0.1270)	(0.7304)		(0.0294)	(1)	(0.0571)	(0.6163)	
Cloglog	0	0	0	0.5000	**0.034299**	0	0	0	0.4767	0.046425
	(0.1000)	(0.7500)	**(0.1765)**	**(0.7426)**		**(0.0476)**	(1)	**(0.0909)**	**(0.7674)**	

Note: Values *without* brackets are the classifications results obtained using the conventional threshold $\hat{p}_0 = 0.5$. Values *in* brackets are classifications results obtained using the threshold $\hat{p}_0 = \bar{y}$. The best values in each column given to the specified thresholds are in **boldface**.

correctly classify some events in this data set, especially the one with Gaussian kernel, which managed to classify 94 percent of the true events in-sample. However, in terms of out-of-sample prediction, the performance of the Gaussian SVM-DEC dropped sharply, indicating a possible overfitting in the model training procedure. Since the Gaussian SVM-DEC also has the highest value of C among the cost-sensitive SVMs, meaning that the method was fitted less closely to the data than did the others, the issue of overfitting might be originated from the Gaussian kernel. Meanwhile, the out-of-sample performance of the sigmoid SVM-DEC is even better than its in-sample performance. Nevertheless, this could be attributed partly to chance and partly to the small sample size – with only 7 true events in the test set, one additional correctly classified event would impose a large impact on the sensitivity.

The data set `small_b` is characterised by small sample size and severe class imbalance. In the presence of an event rarity equal to 3.33 percent, all GLMs that used the conventional threshold failed to identify observations from the event class in the training set and hence also those in the test set. This has also occurred in the standard SVM. Meanwhile, comparing with these two versions of classifiers, the GLMs using the alternative threshold and the cost-sensitive SVMs had much better in-sample classification performance. Among these classifiers, all cost-sensitive SVMs and the Relogit model have even achieved the perfect classification performance. However, both GLMs with alternative threshold and cost-sensitive SVMs showed poor performance in terms of precision. There are also some signs of severe overfitting for the Gaussian SVM-DEC, whose F1 score dropped from 0.73 to 0. Compared to the Gaussian version, the polynomial SVM-DEC had a rather moderate drop in the F1 score, from 0.73 to 0.5. However, as in the previous case, this drop in out-of-sample F1 score might also be caused by the small sample size, since the test set in this setting only contains one event.

Table 5.2: Classification performance the SVMs (Scenario I)

		In-sample					*Out-of-sample*				
		Precision	Recall	F1	BA	Brier	Precision	Recall	F1	BA	Brier
Data: small_a											
Linear SVM	$C = 0.001$	0	0	0	0.5000	0.102113	0	0	0	0.5000	0.148378
Gaussian SVM	$C = 0.001$	0	0	0	0.5000	**0.096858**	0	0	0	0.5000	0.133019
Sigmoid SVM	$C = 0.001$	0	0	0	0.5000	0.133561	0	0	0	0.5000	**0.130851**
Polynomial SVM	$C = 0.1$	**1**	**0.29**	**0.45**	**0.6471**	0.098911	**0.33**	**0.14**	**0.2**	**0.5451**	0.139052
Linear SVM-DEC	$C = 0.1$	0.44	0.88	0.59	0.8332	**0.101795**	0.17	0.43	0.24	0.5169	0.141892
Gaussian SVM-DEC	$C = 10$	**0.59**	**0.94**	**0.73**	**0.9081**	0.103642	0.14	0.14	0.14	0.4925	0.135437
Sigmoid SVM-DEC	$C = 5$	0.15	0.47	0.23	0.4853	0.134791	0.25	**0.71**	**0.37**	**0.6598**	0.132932
Polynomial SVM-DEC	$C = 0.001$	0.56	0.29	0.38	0.6243	0.104309	**0.33**	0.14	0.20	0.5451	**0.126985**
Data: small_b											
Linear SVM	$C = 0.001$	0	0	0	0.5000	0.036713	0	0	0	0.5000	0.023466
Gaussian SVM	$C = 0.001$	0	0	0	0.5000	**0.032298**	0	0	0	0.5000	0.024147
Sigmoid SVM	$C = 0.001$	0	0	0	0.5000	0.036110	0	0	0	0.5000	0.022768
Polynomial SVM	$C = 0.001$	0	0	0	0.5000	0.037755	0	0	0	0.5000	**0.020313**
Linear SVM-DEC	$C = 50$	0.11	1	0.21	0.8480	0.036468	0.06	**1**	0.11	0.8023	0.022785
Gaussian SVM-DEC	$C = 5$	**0.57**	1	**0.73**	**0.9853**	0.037441	0	0	0	0.4767	0.023346
Sigmoid SVM-DEC	$C = 0.001$	0.04	1	0.07	0.5000	**0.037380**	0.02	**1**	0.04	0.5000	0.023031
Polynomial SVM-DEC	$C = 5$	**0.57**	1	**0.73**	**0.9853**	0.039142	**0.33**	**1**	**0.50**	**0.9767**	**0.019630**

Note: The best values in each column are in **boldface**.

5.2 Scenario II

The classification performance of the GLMs and SVMs in this scenario are summarised in Table 5.3 and 5.4, respectively. Compare to Scenario I, data sets in this scenario have more predictors – hence increased dimensions – and larger sample size. The class imbalance presented in the three data sets in this scenario can be described as *moderate* ($\bar{y} = 16.2\%$), *severe* ($\bar{y} = 3.4\%$), and *extreme* ($\bar{y} = 0.8\%$).

Concerning the classification performance of the GLMs, Table 5.3 shows that the GLMs only managed to identify some events in the data set med_a, which is characterised by moderate class imbalance, while failed to do so in others. In data set med_a, all three GLMs have achieved a better in-sample and out-of-sample precision than what they did in the data set small_a. Their performances in all other metrics, however, have decreased in both the training and the test set. It is likely that this drop of classification performance was caused by the increased model complexity in the data set med_a, since the rarities of the data sets med_a and small_a only differ in 0.2 percentage points.

As in Scenario I, it is hard to tell which of these three GLMs is superior, since they all produced identical performance in out-of-sample prediction using the conventional threshold. Using the alternative threshold, the Cloglog model managed to achieve a slightly better out-of-sample performance in precision, F1 score and BA. From Table 5.3, we can also observe that the LR and ReLogit models have identical classification performance. These two methods only differed in the Brier score, with King & Zeng's [2001a] ReLogit model performing slightly better than the LR. In addition, the results from data sets med_b and med_c seem to suggest that the GLMs using the conventional threshold are not suitable for modelling events that occur less than 5 percent of the time. There are some improvements in their classification performance if the alternative threshold is used, most noticeably in the data set med_b. As for data set med_c, only the ReLogit model managed to achieve a out-of-sample BA above 0.5.

As for the SVMs, their classification results in Scenario II are to some extent different from those

Table 5.3: Classification performance the GLMs (Scenario II)

	In-sample					*Out-of-sample*				
	Precision	Recall	F1	BA	Brier	Precision	Recall	F1	BA	Brier
Data: `med_a`										
LR	0.6818	**0.2632**	**0.3798**	**0.6196**	**0.107824**	0.6667	0.0833	0.1482	0.5377	0.110802
	(0.3419)	(0.7018)	**(0.4598)**	**(0.7195)**		(0.3333)	(0.7083)	(0.4533)	(0.7192)	
ReLogit	0.6818	**0.2632**	**0.3798**	**0.6196**	0.108114	0.6667	0.0833	0.1482	0.5377	0.110790
	(0.3228)	**(0.7193)**	(0.4457)	(0.7129)		(0.3091)	(0.7083)	(0.4304)	(0.7034)	
Cloglog	**0.6842**	0.2281	0.3421	0.6038	0.108631	0.6667	0.0833	0.1482	0.5377	**0.110705**
	(0.3276)	(0.6667)	(0.4393)	(0.7002)		**(0.3617)**	(0.7083)	**(0.4789)**	**(0.7351)**	
Data: `med_b`										
LR	0	0	0	0.5000	**0.031599**	0	0	0	0.5000	0.031944
	(0.0826)	(0.7500)	(0.1488)	(0.7271)		**(0.0571)**	(0.4000)	**(0.1000)**	(0.5862)	
ReLogit	0	0	0	0.5000	0.031931	0	0	0	0.5000	0.032661
	(0.0692)	**(0.9167)**	(0.1287)	**(0.7394)**		(0.0500)	**(0.6000)**	(0.0923)	**(0.6034)**	
Cloglog	0	0	0	0.5000	0.031634	0	0	0	0.5000	**0.031884**
	(0.0833)	(0.7500)	**(0.1500)**	(0.7286)		(0.0556)	(0.4000)	(0.0976)	(0.5828)	
Data: `med_c`										
LR	0	0	0	**0.5000**	0.008453	0	0	0	**0.5000**	**0.008799**
	(0.0566)	(1)	**(0.1071)**	**(0.9280)**		(0)	(0)	(0)	(0.3725)	
ReLogit	0	0	0	0.4986	0.012813	0	0	0	0.4966	0.017448
	(0.0124)	(1)	(0.0245)	(0.6556)		**(0.0090)**	**(1)**	**(0.0179)**	**(0.6300)**	
Cloglog	0	0	0	**0.5000**	**0.008396**	0	0	0	**0.5000**	0.009002
	(0.0566)	(1)	**(0.1071)**	**(0.9280)**		(0)	(0)	(0)	(0.3691)	

Note: Values *without* brackets are the classifications results obtained using the conventional threshold $\hat{p}_0 = 0.5$. Values *in* brackets are classifications results obtained using the threshold $\hat{p}_0 = \bar{y}$. The best values in each column given to the specified thresholds are in **boldface**.

in Scenario I. In the data set `med_a`, there were two standard SVMs that managed to identify some events: the SVM with Gaussian kernel and then one with polynomial kernel. The in-sample precision and recall of these two standard SVMs have also increased, indicating that the kernelised SVMs might be better in handling complex data than did the GLMs. However, the performance of these two SVMs in out-of-sample recall remained on a relatively low level, with the best of these two – the Gaussian SVM – slightly surpassing the 20 percent level.

The cost-sensitive SVMs produced a better performance than the standard SVM in this data set, especially in terms of recall. The cost-sensitive SVM with sigmoid kernel has even reached the maximum score of recall in both the training and the test set. However, as indicated by its result in BA, the sigmoid SVM-DEC achieved the maximum performance in recall simply because it always assign the observations to the positive class. This renders the sigmoid SVM-DEC as an unreliable method in rare event modelling. Additionally, the results in Table 5.4 also show that the strong performance of the cost-sensitive SVMs comes at the price of increased false positives and hence low precision. As a consequence, in this data set, none of the cost-sensitive SVMs have an out-of-sample F1 score that equaled or surpassed the 0.5 level. In other words, the cost-sensitive SVMs did not deliver a prediction performance superior to the best-performing GLMs (in this case, GLMs that used the alternative threshold) in data set `med_a`.

In the data sets `med_b` and `med_c`, the standard SVMs had all failed to identify the events, while its

Table 5.4: Classification performance the SVMs (Scenario II)

		In-sample					*Out-of-sample*				
		Precision	Recall	F1	BA	Brier	Precision	Recall	F1	BA	Brier
Data: `med_a`											
Linear SVM	$C = 0.001$	0	0	0	0.5000	0.114565	0	0	0	0.5000	**0.116475**
Gaussian SVM	$C = 10$	**0.89**	**0.42**	**0.57**	**0.7054**	**0.088602**	0.62	**0.21**	**0.31**	**0.5923**	0.116688
Sigmoid SVM	$C = 0.001$	0	0	0	0.5000	0.136337	0	0	0	0.5000	0.134420
Polynomial SVM	$C = 1$	0.86	0.33	0.48	0.6615	0.102705	**0.75**	0.12	0.21	0.5585	0.119579
Linear SVM-DEC	$C = 0.001$	0.32	0.67	0.44	0.6985	0.111129	0.35	0.71	**0.47**	**0.7272**	0.113676
Gaussian SVM-DEC	$C = 1$	**0.50**	0.68	**0.58**	0.7756	**0.094952**	**0.38**	0.54	0.45	0.6875	**0.110362**
Sigmoid SVM-DEC	$C = 5$	0.16	**1**	0.28	0.5000	0.131892	0.16	**1**	0.28	0.5000	0.131209
Polynomial SVM-DEC	$C = 1$	0.44	0.74	0.55	**0.7763**	0.107354	0.31	0.62	0.41	0.6776	0.114890
Data: `med_b`											
Linear SVM	$C = 0.001$	0	0	0	0.5000	0.033220	0	0	0	0.5000	0.032595
Gaussian SVM	$C = 0.001$	0	0	0	0.5000	**0.031901**	0	0	0	0.5000	0.032023
Sigmoid SVM	$C = 0.001$	0	0	0	0.5000	0.033148	0	0	0	0.5000	0.032273
Polynomial SVM	$C = 0.001$	0	0	0	0.5000	0.032928	0	0	0	0.5000	0.032607
Linear SVM-DEC	$C = 1$	0.10	0.83	0.18	0.7835	**0.032331**	**0.05**	0.4	**0.09**	**0.5724**	**0.031994**
Gaussian SVM-DEC	$C = 50$	0.80	**1**	0.89	0.9956	0.032854	0	0	0	0.4897	0.032144
Sigmoid SVM-DEC	$C = 5$	0.03	**1**	0.07	0.5000	0.033148	0.03	**1**	0.07	0.5000	0.032273
Polynomial SVM-DEC	$C = 50$	**0.86**	**1**	**0.92**	**0.9970**	0.032890	0	0	0	0.4759	0.032301
Data: `med_c`											
Linear SVM	$C = 0.001$	0	0	0	0.5000	0.008518	0	0	0	0.5000	**0.006661**
Gaussian SVM	$C = 0.001$	0	0	0	0.5000	0.008559	0	0	0	0.5000	0.006767
Sigmoid SVM	$C = 0.001$	0	0	0	0.5000	0.008529	0	0	0	0.5000	0.006678
Polynomial SVM	$C = 0.001$	0	0	0	0.5000	**0.008475**	0	0	0	0.5000	0.006716
Linear SVM-DEC	$C = 1$	0.15	1	0.26	0.9755	0.008421	0	0	0	0.4631	**0.006679**
Gaussian SVM-DEC	$C = 50$	**1**	1	**1**	**1**	**0.007916**	0	0	0	0.4765	0.006723
Sigmoid SVM-DEC	$C = 1$	0.01	1	0.02	0.5000	0.008502	**0.01**	**1**	**0.01**	**0.5000**	0.006700
Polynomial SVM-DEC	$C = 5$	0.60	1	0.75	0.9971	0.008424	0	0	0	0.4765	0.006689

Note: The best values in each column are in **boldface**.

cost-sensitive versions varied in classification performance. The problem of overfitting in the Gaussian and polynomial SVM-DEC might be augmented as the event class becomes more rare. Classification results for the data sets `med_b` and `med_c` shows that both classifiers have achieved high F1 scores in the training set, indicating great harmony in the precision-recall trade-off. In the test set, however, all nominal evaluation metrics suddenly drop to zero. This would imply that the model resulted from the Gaussian and polynomial SVM-DEC have no practical value in these data sets, since they were unable to predict any future outcomes. The classification performances of linear SVM-DEC and of sigmoid SVM-DEC appeared to be relatively stable between the training and the test set. However, they are still poor results, especially considering the large number of out-of-sample false positives resulted from these two cost-sensitive SVMs.

5.3 Scenario III

The classification performance of the GLMs and SVMs in this scenario are summarised in Table 5.5 and 5.6, respectively. In Scenario III, the number of observations included in the data sets have doubled from that in Scenario II. And the number of predictors have increased to 20. The event rarity in the data sets included in this scenario can be described as *moderate* ($\bar{y} = 16.2\%$) and *severe* ($\bar{y} = 3.5\%$).

Similar to the previous scenarios, the GLMs using the conventional threshold failed to identify events in the data set that characterised by severe rarity. In the data set `large_b`, all GLMs that used the threshold $\hat{p}_0 = 0.5$ obtained zeros in in-sample classification. For this reason, the unusual out-of-sample performance of the Cloglog model seems to occur merely by chance. The classification performance of these GLMs was improved when the threshold $\hat{p}_0 = \bar{y}$ was used, with the ReLogit model outperforming the other two models in terms of recall. However, the models' low scores in precision showed that this strong performance comes at the cost of increased false positives.

In the `large_a` data set, the in-sample classification performances of the GLMs have decreased compared to their performances in similar data sets in previous scenarios, suggesting that the learning capacity of GLM are negatively correlated to the dimensions of data. However, the out-of-sample recall, F1 score, and BA for all GLMs are slightly higher in the data set `large_a` than that in `med_a`, a phenomenon which we have not observed between `med_a` and `small_a`. It is highly unlikely that the difference in rarity caused this phenomenon, since the difference in the event rarity presented in the test sets of `small_a`, `med_a`, and `large_a` are less than 1 percentage point. Meanwhile, as shown in Table 5.5, all three GLMs delivered similar classification performance in the `large_a` data set. The LR and the ReLogit model had even delivered identical classification performance when using the convectional threshold.

Table 5.5: Classification performance the GLMs (Scenario III)

	In-sample					*Out-of-sample*				
	Precision	Recall	F1	BA	Brier	Precision	Recall	F1	BA	Brier
Data: `large_a`										
LR	0.5000	0.1150	0.1871	0.5464	0.115578	**0.4444**	**0.1633**	**0.2388**	**0.5617**	0.129612
	(0.3089)	(0.7080)	(0.4301)	(0.7015)		**(0.2946)**	(0.6735)	(0.4099)	(0.6794)	
ReLogit	0.5000	0.1150	0.1871	0.5464	0.115565	**0.4444**	**0.1633**	**0.2388**	**0.5617**	**0.129394**
	(0.3004)	**(0.7257)**	(0.4249)	(0.7001)		(0.2917)	**(0.7143)**	**(0.4142)**	**(0.6878)**	
Cloglog	**0.5600**	**0.1239**	**0.2029**	**0.5526**	**0.115502**	0.4118	0.1429	0.2121	0.5515	0.131194
	(0.3160)	(0.6991)	**(0.4353)**	**(0.7039)**		(0.2897)	(0.6327)	(0.3974)	(0.6649)	
Data: `large_b`										
LR	0	0	0	0.5000	0.030129	0	0	0	0.5000	**0.032901**
	(0.0784)	(0.6667)	(0.1404)	(0.6943)		(0.0633)	(0.4545)	(0.1111)	(0.5992)	
ReLogit	0	0	0	0.5000	0.030469	0	0	0	0.4983	0.033487
	(0.0669)	**(0.7917)**	(0.1234)	**(0.6998)**		(0.0696)	**(0.7273)**	(0.1270)	**(0.6785)**	
Cloglog	0	0	0	0.5000	**0.030013**	**0.5000**	**0.0909**	**0.1539**	**0.5437**	0.032978
	(0.0804)	(0.6667)	**(0.1435)**	(0.6980)		**(0.0750)**	(0.5455)	**(0.1319)**	(0.6447)	

Note: Values *without* brackets are the classifications results obtained using the conventional threshold $\hat{p}_0 = 0.5$. Values *in* brackets are classifications results obtained using the threshold $\hat{p}_0 = \bar{y}$. The best values in each column given to the specified thresholds are in **boldface**.

As for the standard SVMs, the results are similar to that in Scenario II. In the `large_b` data set, all standard SVMs failed to identify the events. Meanwhile, in the `large_a` data set, the Gaussian and polynomial SVMs were once again the only two standard SVMs that managed to detect some observations from the event class. As in the `med_a` data set, in this data set, the Gaussian and polynomial SVMs demonstrated the flexibility of kernelised SVMs in handling increased model complexity, outperforming

GLMs that used the conventional threshold. However, this did not lead to more correctly predicted events in the test set. For both the Gaussian and polynomial SVMs, only 6 percent of the events in the test set were correctly predicted, fewer than the GLMs with $\hat{p}_0 = 0.5$ threshold, not to mention those that uses the threshold $\hat{p}_0 = \bar{y}$.

Table 5.6: Classification performance the SVMs (Scenario III)

		In-sample					*Out-of-sample*				
		Precision	Recall	F1	BA	Brier	Precision	Recall	F1	BA	Brier
Data: `large_a`											
Linear SVM	$C = 0.001$	0	0	0	0.5000	0.126496	0	0	0	0.5000	0.132109
Gaussian SVM	$C = 10$	**0.97**	**0.32**	**0.48**	**0.6584**	0.110489	0.38	**0.06**	**0.11**	0.5207	0.127116
Sigmoid SVM	$C = 0.001$	0	0	0	0.5000	0.135370	0	0	0	0.5000	0.136658
Polynomial SVM	$C = 1$	0.91	0.28	0.43	0.6390	**0.100920**	**0.50**	**0.06**	**0.11**	**0.5246**	**0.126924**
Linear SVM-DEC	$C = 50$	0.31	0.75	0.44	0.7177	0.118064	**0.30**	0.69	**0.42**	**0.6916**	0.126674
Gaussian SVM-DEC	$C = 50$	0.70	**1**	0.82	0.9591	**0.099250**	**0.30**	0.47	0.37	0.6291	0.129519
Sigmoid SVM-DEC	$C = 5$	0.16	**1**	0.28	0.5000	0.123883	0.16	**1**	0.28	0.5000	**0.126332**
Polynomial SVM-DEC	$C = 50$	**0.76**	**1**	**0.86**	**0.9693**	0.107780	0.26	0.39	0.31	0.5883	0.131405
Data: `large_b`											
Linear SVM	$C = 0.001$	0	0	0	0.5000	0.032870	0	0	0	0.5000	0.035180
Gaussian SVM	$C = 0.001$	0	0	0	0.5000	0.031444	0	0	0	0.5000	**0.034210**
Sigmoid SVM	$C = 0.001$	0	0	0	0.5000	0.033114	0	0	0	0.5000	0.035322
Polynomial SVM	$C = 0.001$	0	0	0	0.5000	**0.031198**	0	0	0	0.5000	0.035038
Linear SVM-DEC	$C = 0.001$	0.09	0.75	0.17	0.7456	0.032171	**0.09**	**0.73**	**0.16**	**0.7235**	0.034741
Gaussian SVM-DEC	$C = 0.001$	0	0	0	0.5000	0.031822	0	0	0	0.5000	0.034916
Sigmoid SVM-DEC	$C = 0.001$	0	0	0	0.5000	**0.031779**	0	0	0	0.5000	**0.034595**
Polynomial SVM-DEC	$C = 10$	**0.86**	**1**	**0.92**	**0.9970**	0.032152	0.05	0.09	0.06	0.5109	0.035161

Note: The best values in each column are in **boldface**.

Regarding the cost-sensitive SVMs, the results we observed in Table 5.6 are also similar to that in the previous scenarios. They achieved a better classification performance than the GLMs and the standard SVMs in the data sets `large_a`, although the improvements compared to the result of GLMs using the alternative threshold are merely marginal. We can also observe some degrees of overfitting in Gaussian and polynomial SVM-DEC, as well as the phenomenon that the sigmoid SVM-DEC assigns all observations to the positive class. Similar results were also obtained in the the data sets `large_b`. Generally, in the presence of moderate class imbalance, the kernelised cost-senstive SVMs did not perform much better than the non-kernelised (linear) version in terms of out-of-sample metrics. Under the presence of severe class imbalance, as in the case of `large_b`, the linear SVM-DEC even performed better than those kernelised SVM-DECs in the test set, despite the large number of false positives.

Chapter 6

Results from the empirical analysis

6.1 Classification performance

Through the simulation study, we have obtained some knowledge about the classification performance of GLMs and SVMs under the presence of various levels of class imbalance. In the empirical analysis, I investigated whether the classification patterns of different methods observed in the simulation study still hold. The MID data set has a event rarity around 0.34 percent and contains 10 predictors. For this reason, I expected that the classification results obtained from the MID data would be similar to those obtained from the `med_b7` data set in Scenario II.

Table 6.1: Classification performance the GLMs (MID)

	In-sample					*Out-of-sample*				
	Precision	Recall	F1	BA	Brier	Precision	Recall	F1	BA	Brier
LR	**0.5000** (**0.0295**)	0.0137 (0.8767)	**0.0267** (0.0570)	0.5068 (0.8886)	**0.003198**	**0.5000** (0.0283)	0.0323 (0.8065)	**0.0606** (0.0547)	**0.5161** (**0.8589**)	0.003343
ReLogit	0.3333 (0.0271)	0.0137 (0.8767)	0.0263 (0.0526)	0.5068 (0.8841)	0.003201	**0.5000** (0.0255)	0.0323 (0.8065)	**0.0606** (0.0494)	**0.5161** (0.8506)	0.003355
Cloglog	0.2000 (**0.0295**)	0.0137 (0.8767)	0.0256 (**0.0572**)	0.5068 (**0.8888**)	0.003200	0.3333 (**0.0284**)	0.0323 (0.8065)	0.0588 (**0.0548**)	0.5160 (0.8561)	**0.003336**

Note: Values *without* brackets are the classifications results obtained using the conventional threshold $\hat{p}_0 = 0.5$. Values *in* brackets are classifications results obtained using the threshold $\hat{p}_0 = \bar{y}$. The best values in each column given to the specified thresholds are in **boldface**.

The classification performance of the GLMs and the SVMs are shown in Tables 6.1 and 6.2, respectively. The results show that the GLMs using the conventional threshold managed to identify some of the events in both the training and the test set, although the low recall score indicates that all these GLMs had low sensitivity. The in-sample and out-of-sample BA scores of these GLMs show that they are only marginally better than a classifier that always assigns observations to one class. As expected, the GLMs using the alternative threshold achieved a better result in recall at the cost of decreased precision. The BA for such GLMs had also improved. But the low F1 score still indicates a poor trade-off between the ability of detecting the actual events and the ability of making the correct decision. The LR and the ReLogit models yielded nearly identical results as expected. Using the conventional threshold, the GLM with Cloglog link seems to be an inferior model relative to LR and ReLogit in this data set because of its poor performance in both in-sample and out-of-sample precision. However, when the alternative threshold was adopted, the classification performances of the three GLMs listed in 6.1 hardly differed.

Regarding the SVMs, the standard SVMs had, as expected, all failed to identify the events. The cost-sensitivity versions achieved better performance than both the standard SVMs and the GLMs using the conventional threshold. As shown in Table 6.2, the cost-sensitive SVMs in general managed to identify more than 75 percent of the actual military disputes in the training set and more than 60 percent of the same outcome in the test set. However, such high accuracies in identifying actual conflicts come at the price of the large number of false positives and hence a low F1 score both in-sample and out-of-sample. Table 6.2 also shows that the sigmoid SVM-DEC achieved a better classification performance in the test set than in the training set. Since all other cost-sensitive SVMs showed the reversed case, it is unlikely that such anomaly was caused by the data structure of MID. Therefore, this phenomenon would imply that the sigmoid SVM-DEC tends to classify the observation randomly when rare events are presented in the data. Meanwhile, the issue of overfitting in the Gaussian and the polynomial SVM-DEC is also presented in the empirical analysis, so does the relative stable performance of the linear SVM-DEC across the training and the test set. Since I did not split the MID data by time periods as in Beck et al. [2000], but randomly select observations from the event and the non-event class separately, we can also eliminate the possibility that the reductions in out-of-sample recall for SVMs with Gaussian and polynomial kernels were caused by exogenous changes in the real world after a certain point of time. In addition, I had also attempted to prevent overfittning by tunning the parameter C when fitting the SVMs to the MID data using `GridSearchCV`. As shown in Table 6.2, the optimal value of C chosen for the Gaussian and the polynomial SVM-DEC is the largest of the five candidates, hence reaching the maximum level of violations to the training set I am prepared to tolerate. Still, the issue of overfitting remained. Thus, the results in Table 6.2 provide some evidence for the claim that Gaussian and polynomial SVM-DEC tend to overfit the data.

Table 6.2: Classification performance the SVMs (MID)

		In-sample					*Out-of-sample*				
		Precision	Recall	F1	BA	Brier	Precision	Recall	F1	BA	Brier
Linear SVM	$C = 0.001$	0	0	0	0.5000	**0.003349**	0	0	0	0.5000	**0.003336**
Gaussian SVM	$C = 0.001$	0	0	0	0.5000	0.003418	0	0	0	0.5000	0.003390
Sigmoid SVM	$C = 0.001$	0	0	0	0.5000	0.003418	0	0	0	0.5000	0.003388
Polynomial SVM	$C = 0.001$	0	0	0	0.5000	0.003413	0	0	0	0.5000	0.003382
Linear SVM-DEC	$C = 0.001$	0.03	0.90	0.05	0.8925	0.003287	0.03	**0.87**	0.05	**0.8813**	0.003409
Gaussian SVM-DEC	$C = 50$	**0.07**	**0.99**	**0.13**	**0.9721**	**0.003203**	**0.04**	0.61	**0.08**	0.7819	**0.003312**
Sigmoid SVM-DEC	$C = 0.001$	0.01	0.78	0.01	0.6962	0.003382	0.01	0.84	0.01	0.7283	0.003355
Polynomial SVM-DEC	$C = 50$	0.04	**0.99**	0.07	0.9482	0.003390	0.03	0.74	0.05	0.8245	0.003395

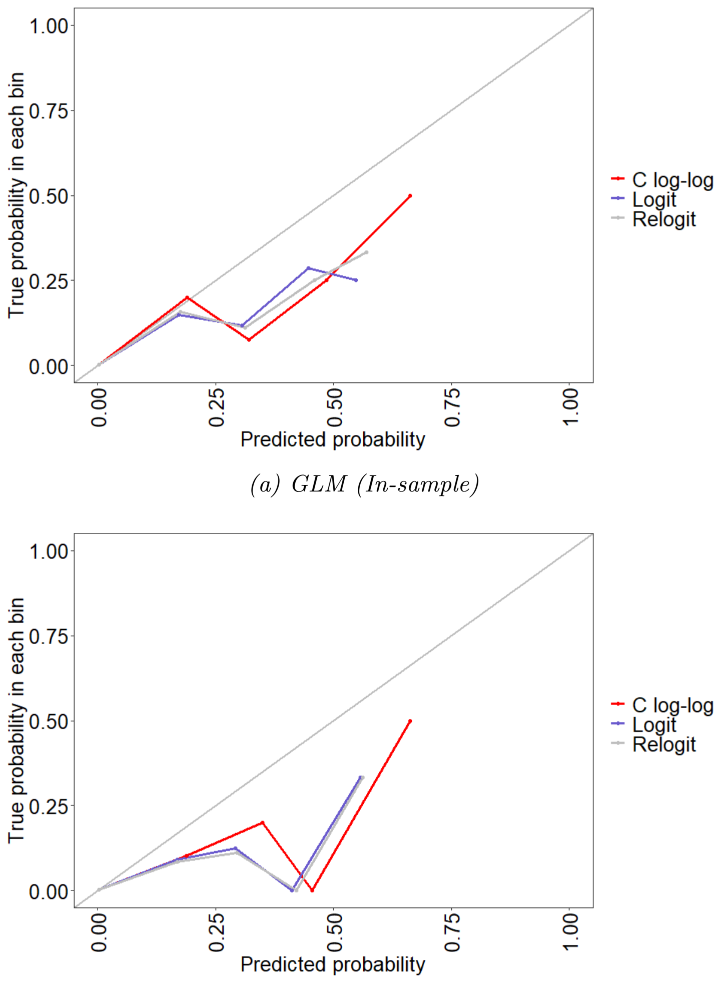
Note: The best values in each column are in **boldface**.

6.2 Probabilistic estimates

As mentioned in the Section 2.3.4, the SVMs do not produce any probabilistic outputs, such as the regression output of GLM shown in Table A.2. The probabilistic evaluation metrics, such as the Brier scores presented in the tables above, were computed using Platt scaling. To compare the probabilistic output of the SVMs with that of the GLMs in a fair manner, I used the the reliability diagram (Figures 6.1–6.3), the ROC curve (Figures 6.4 and 6.5), and the precision-recall curve (Figures 6.6–6.8).

6.2.1 Reliability diagram

Comparing Figure 6.1 with Figures 6.2 and 6.3, it is clear that the probabilistic outputs produced by the GLMs are superior than those by the SVMs. The three curves in subfigures 6.1a and 6.1b all demonstrated the tendency of converging towards the diagonal line, which, as mentioned in Section 2.3.4 above, represents the perfect match between the actual probabilities and the predicted probabilities. We can also observe that the probability estimates of LR and that of ReLogit are nearly identical to each other, while the predicted probabilities from the Cloglog model extend far more beyond the 0.5 threshold than do the predicted probabilities from LR and ReLogit. This is coherent with the theoretical claim that the Cloglog link approaches to 1 at a faster rate than it approaches to 0.[33]



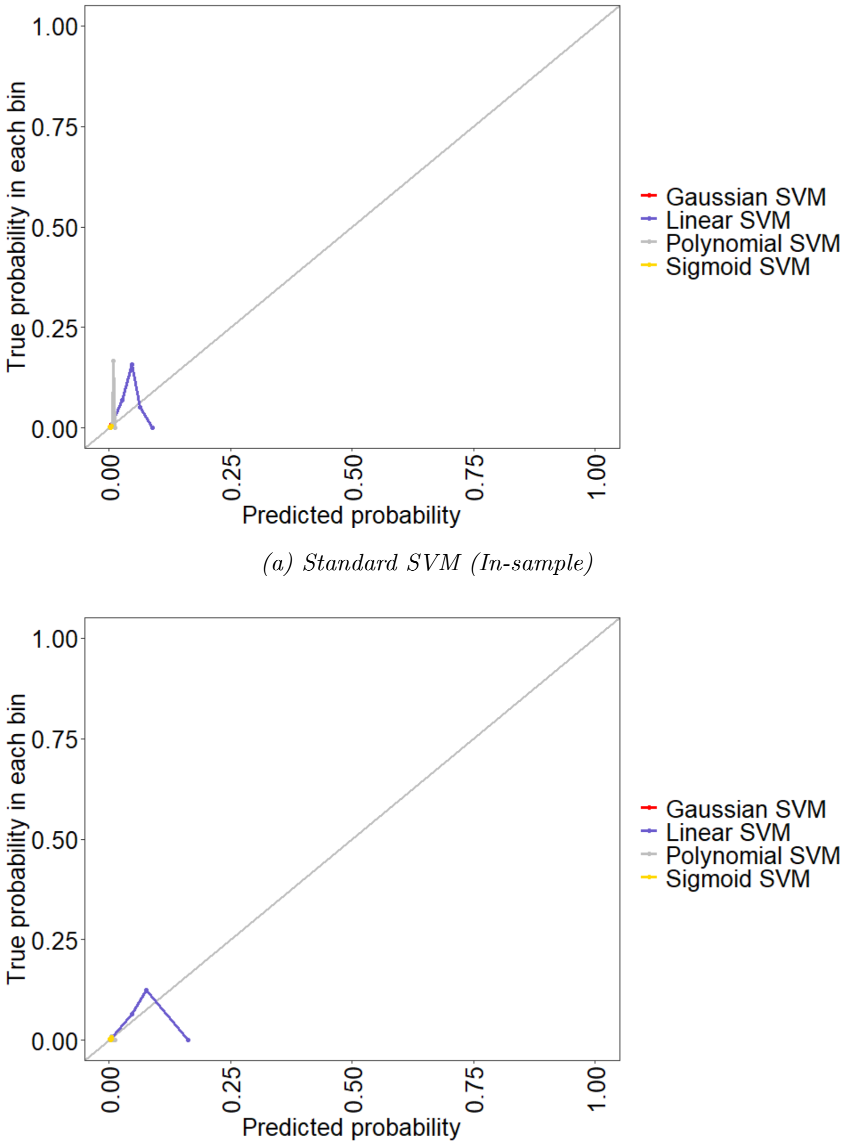
(a) GLM (In-sample)

(b) GLM (Out-of-sample)

Figure 6.1: Reliability diagram of the GLMs applied to the MID data.

As for the standard SVMs, Figure 6.2 shows that only the linear SVM managed to produce a calibration curve similar to that of the GLMs, while the kernelised versions all ended in some points near the origin, meaning that the classifier predicted nearly zero probability for all observations in the data set. In contrast to the standard SVMs, the cost-sensitive SVMs did produce predicted probabilities

[33] As a side note, Figure 6.1 also shows that the curves for the LR, ReLogit, and Cloglog model lie under the diagonal line, indicating that the predicted probabilities from these models are too high. It is worth to mention that this has nothing to do with the overestimation of the (conditional) probability of event occurrence, $\Pr(Y = 1|X)$, which we discussed in Section 3.1. The reason is simple: as mentioned in Section 2.3.4, the y-axis of the reliability diagram is the observed frequency of positive cases conditional to the *predictions*, $\Pr(y = 1|s(\mathbf{x}) = s)$, which is not the same as $\Pr(Y = 1|X)$, which is conditional to the *predictors*, X. Therefore, the calibration curves of the LR, ReLogit, and Cloglog model shown in Figure 6.1 do not contradict the theoretical results presented in Section 3.1.



(a) Standard SVM (In-sample)

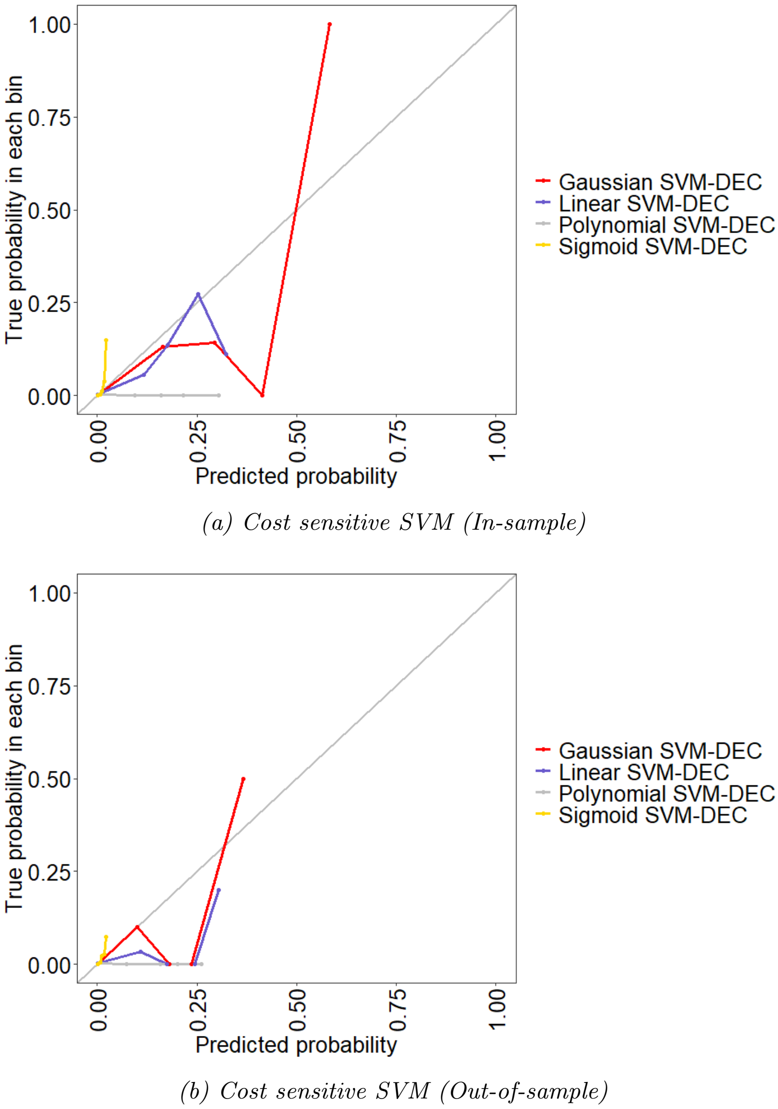
(b) Standard SVM (Out-of-sample)

Figure 6.2: Reliability diagram of the standard SVMs applied to the MID data.

considerably greater than zero, most noticeably, the probabilities predicted by the Gaussian SVM-DEC, which, as the reliability diagram for in-sample prediction in subfigure 6.3a shows, surpassed the 0.5 threshold. However, in terms of out-of-sample predictions, the probability estimates produced by cost-sensitive SVMs are still inferior to those produced by GLMs. In addition, subfigures 6.3a and 6.3b also shows that the calibration curves of all four classifiers diverge from the diagonal line in various extents.

Comparing the results shown in Table 6.2 with those in Figure 6.2, we might see some indications of inconsistency between the classification performance of SVM measured in nominal terms and the method's probabilistic outputs provided through Platt scaling. From Table 6.2, we know that all standard SVMs had failed to detect events in the MID data and classified all observations as non-events. The probabilistic outputs of the kernelised standard SVMs can be regarded as coherent with this result. However, this is not the case for the linear SVM. The calibration curves presented in Figure 6.2 imply that, using the predicted probabilities produced by Platt scaling, the linear SVM is capable of identifying some events in both the training and the test set – if we set the threshold to $\hat{p}_0 = \bar{y}$. However, as we have seen from Table 6.2, this implication is not consistent with the nominal metrics.

To explain this inconsistency, recall that the predicted probabilities produced by Platt scaling are converted from the SVM scores, which in turn are related to the distance of each of the observations to the separating hyperplane. In the case of the linear SVM, it seems that the Platt scaling had given those observations that lie in close distance to the separating hyperplane – probably support vectors – a probability of belonging to the event class higher than zero. As mentioned in Section 2.3.4, this relies on the assumption that the distance of an observation to the separating hyperplane – the ”confidence” we have for our classification – reflects the probability of that observation belonging to the positive class. However, this assumption might not always hold, for the simple reason that ”confidence” is not equivalent to probability. The fact that an observation lies in close distance to the separating hyperplane could only imply that we are uncertain about our classification of that observation (i.e., about whether the decision boundary is correctly placed), it does not necessarily imply that this observation has a higher probability of belonging to the other class. It is probably here the nominal classification results of the base classifier and the predicted probabilities produced by Platt scaling begin to diverge.



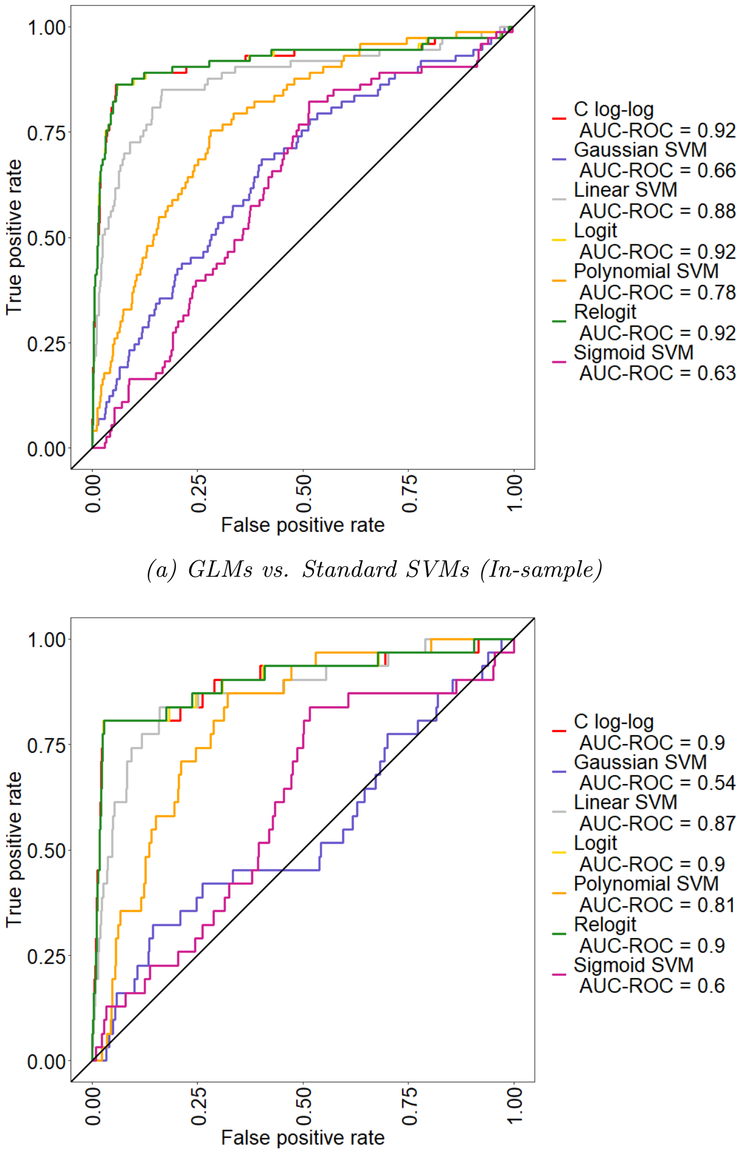
(a) Cost sensitive SVM (In-sample)

(b) Cost sensitive SVM (Out-of-sample)

Figure 6.3: Reliability diagram of the cost sensitive SVMs applied to the MID data.

6.2.2 ROC curve and AUC

As mentioned in Section 4.4.2, the use of ROC/AUC has many shortcomings in evaluating different classifiers' classification performance under class imbalance. However, the ROC curve is still a convenient tool for assessing the reliability of the probability estimates generated from Platt scaling. In Figures 6.4 and 6.5, I plotted the ROC curves of the standard and of the cost-sensitive SVMs, respectively, and compared these curves alongside the ROC curves of the GLMs. The AUC was also computed at the same time.

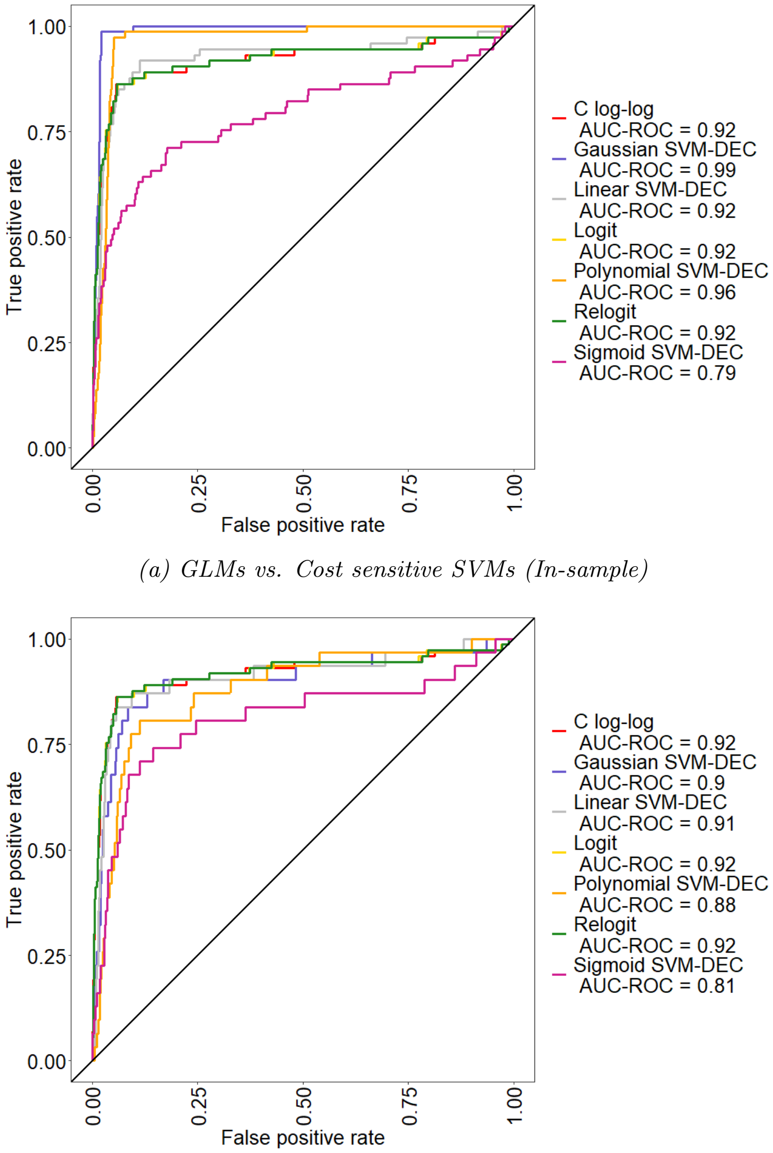


(a) GLMs vs. Standard SVMs (In-sample)

(b) GLMs vs. Standard SVMs (out-of-sample)

Figure 6.4: The ROC curve of the GLMs and the standard SVMs applied to the MID data.

Figure 6.4 shows that the standard SVMs – after being converted to probabilistic models through Platt scaling – are generally inferior than the GLMs. In both the training and the test set, the ROC curves of all four standard SVMs are located under those of the GLMs. The curves of the kernelised SVMs are also not distant from the diagonal line, which represents the performance of a ”no information” classifier that classifies observations randomly. For this reason, the AUCs of these standard SVMs are also lower than those of the GLMs. The Gaussian SVM, in particular, delivered a AUC score slightly above 0.5 in the test set, implying a performance nearly the same as a ”no information” classifier. The only standard SVM that achieved ROC/AUC similar to those of the GLMs is the linear SVM. Since the MID data has 10 predictors, I had expected that the kernelised SVMs are better than the linear SVM in handling data with this level of model complexity. Therefore, the result that the linear SVM performs better than its kernelised counterparts is truly striking. One possible explanation of this could be the inconsistency between the nominal classification results and the probability predictions provided through Platt scaling mentioned in the previous section. As shown in the reliability diagrams in Figure 6.2, the linear SVM is the only standard SVM that managed to obtain – through Platt scaling – probability predictions that are not zeros. For this reason, the linear SVM also became the only standard SVM that is capable of classifying some observations to the event class in the MID data set, and hence capable of achieving a better ROC/AUC than its kernelised counterparts.



(a) GLMs vs. Cost sensitive SVMs (In-sample)

(b) GLMs vs. Cost sensitive (out-of-sample)

Figure 6.5: The ROC curve of the GLMs and the cost-sensitive SVMs applied to the MID data.

In Table 6.2 above, we observe that the cost-sensitive SVMs generally showed a better classification performance than the standard SVMs. As shown in Figure 6.5, this is still the case after these cost-sensitive SVMs were converted to probabilistic models through Platt scaling. Although the ROC curve for sigmoid SVM-DEC, as the standard SVM using the same kernel, is still located under the ROC curves of GLMs, the curves of all the other cost-sensitive SVMs are either located above or at the same level as the curves for GLMs. The Gaussian and the polynomial SVM-DEC, in particular, have achieved a AUC score equal to 0.99 and 0.96, respectively, in the training set. The AUC scores of these two cost-sensitive SVMs dropped below 0.9 eventually, probably because of the overfitting issue linked to the base classifiers using the same kernel. Still, the AUC scores of these two cost-sensitive SVMs are considered as high and hence indicate strong prediction performance. However, the reader must not forget that the ROC/AUC results could be misleading when rare events are presented in the data. In this case, the high AUC score of the Gaussian and the polynomial SVM-DEC might be resulted from the abundance of non-events, which masked the high number of false positives resulted from these two cost-sensitive SVMs and hence led to a low false positive rate that, in turn, produced a seemingly good sensitivity-specificity trade-off.

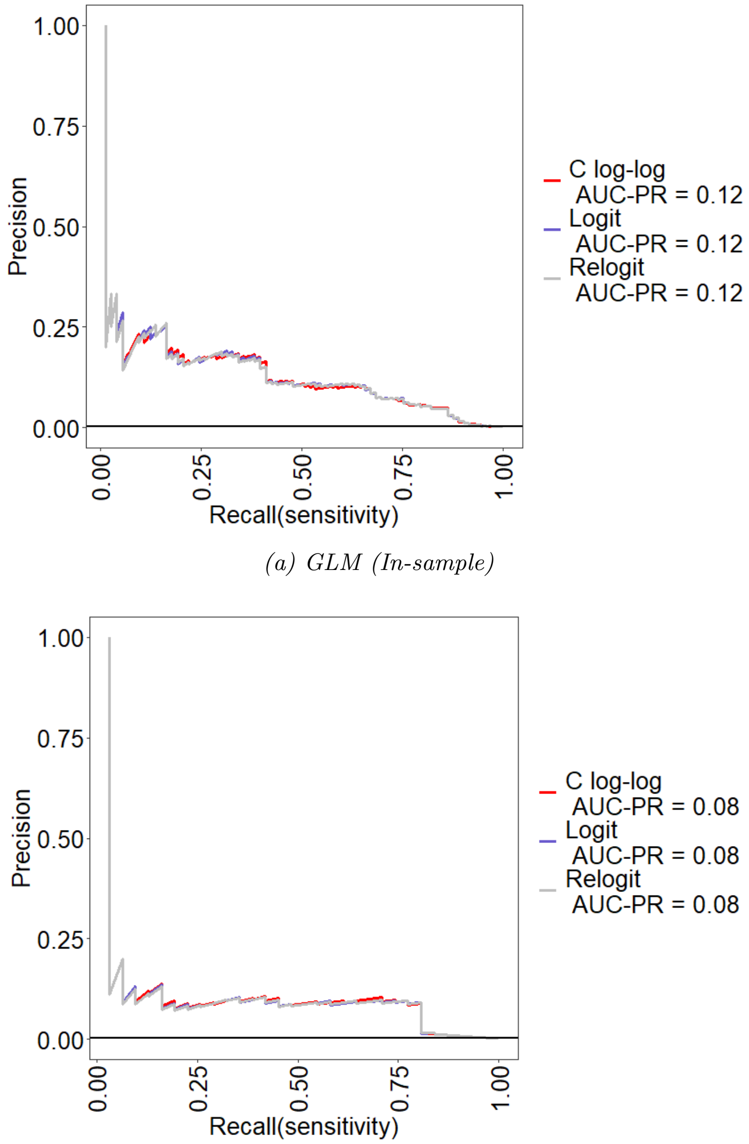
6.2.3 Precision-Recall curve

As mentioned in Section 4.4 above, the precision-recall curve illustrates the trade-off between precision and recall for different classification methods, with good classifiers having their curves bent towards the top-right corner. Like the ROC curve, the precision-recall curve also requires probabilistic outputs. As in the previous cases, both the standard SVMs and the cost-sensitive SVMs were converted to probabilistic models using Platt scaling.

Figures 6.6–6.8 show the precision-recall curves for all classifiers evaluated in this study. The results shown in these figures are unequivocal: all classifiers have their precision-recall curve bent to the left bottom corner – the precise opposite of what a ”good” classifier ought to be. For the GLMs, Figure 6.6 shows that there are almost no differences between the three models in terms of the precision-recall trade-off, as their curves overlap each other. Among the standard SVMs, the linear SVM once again outperformed its kernelised counterpart in this category – an expected result given to what we have observed in the reliability diagrams and the ROC curves. However, compared with that of the GLMs, the precision-recall curve of the linear SVM shows a rather poor trade-off between the two metrics, not to mention the curves of the kernelised versions, which are nearly horizontal lines that lie at the bottom of the diagram. Additionally, Figure 6.8 shows that for SVMs, the trade-off between precision and recall slightly improved when the cost-sensitive versions were used. However, despite the improvements, the curves of all four cost-sensitive SVMs show no better performance than the curves of the GLMs.

The issue of inconsistency between the nominal metrics and the probabilistic outputs is also presented in the precision-recall curves. Comparing the curves of the GLMs with the classification results presented in Table 6.1, it is evident that the nominal metrics and the probability estimates are coherent with each other. This, however, seems not to be the case for the SVMs. For instance, in Table 6.2, the linear SVM showed no better classification performance in terms of precision and recall than its kernelised counterparts, but in Figure 6.7, the linear SVM appears to outperform other SVMs in terms of precision-recall trade-off.

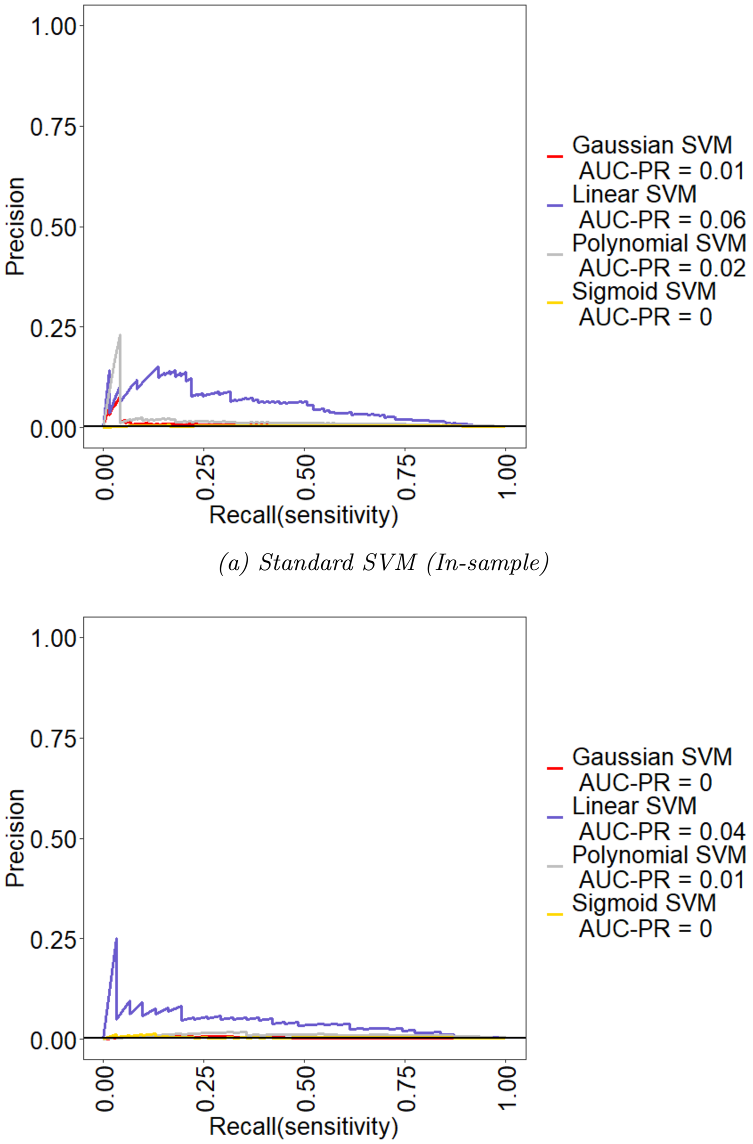
In sum, the inconsistent results shown in this section and the above sections cast doubt on whether the predicted probabilities produced through Platt scaling truly reflect the classification performance of the SVMs. Nevertheless, even if such probabilities are consistent with the classification performance of the base classifier, the precision-recall curves in Figures 6.7 and 6.8 still show that both the standard SVMs and the cost-sensitive version had a poor precision-recall trade-off, indicating not only that these probability estimates are not reliable for rare event prediction, but also that the SVMs might not be an appropriate method to model rare events.



(a) GLM (In-sample)

(b) GLM (Out-of-sample)

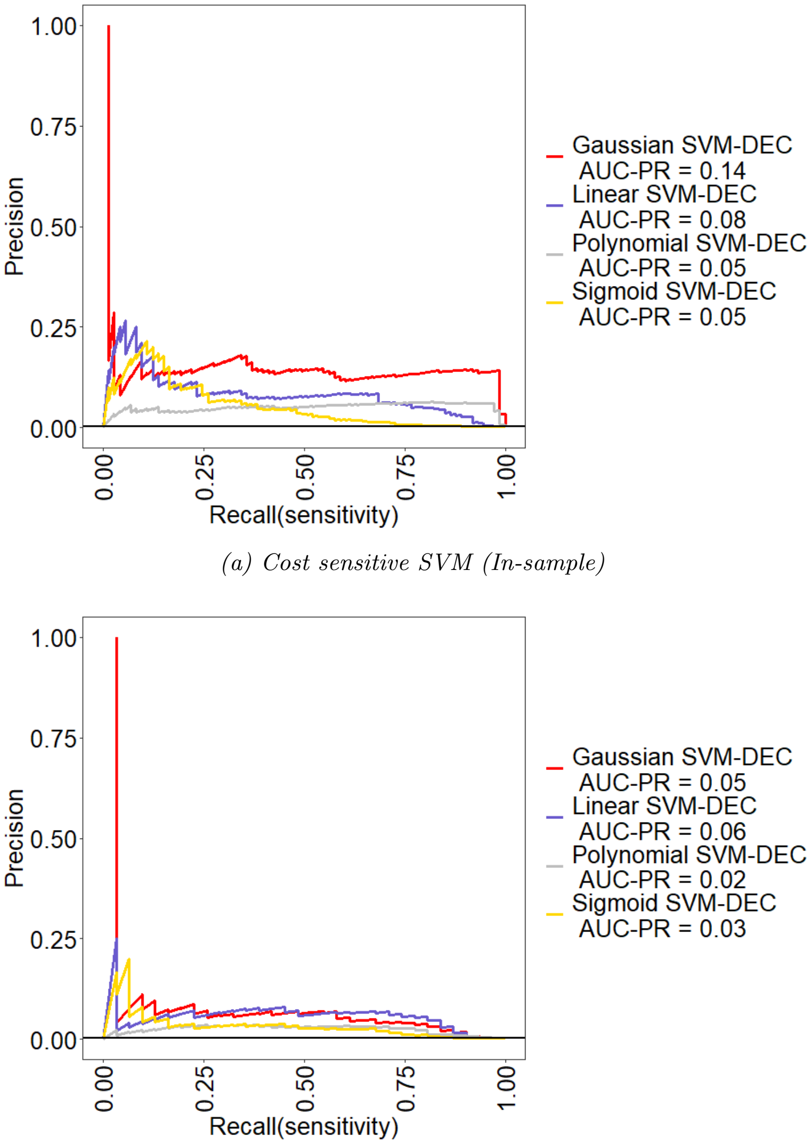
Figure 6.6: Precision-recall curve of the GLMs applied to the MID data.



(a) Standard SVM (In-sample)

(b) Standard SVM (Out-of-sample)

Figure 6.7: Precision-recall curve of the standard SVMs applied to the MID data.



(a) Cost sensitive SVM (In-sample)

(b) Cost sensitive SVM (Out-of-sample)

Figure 6.8: Precision-recall curve of the cost-sensitive SVMs applied to the MID data.

Chapter 7

Discussions

7.1 Empirical implications

7.1.1 Modelling rare events using SVM

The results from the simulation study and the empirical analysis show that the standard SVMs, in the majority of cases, did not achieve a better classification performance than did the GLMs. In Scenario III of the simulation study and the empirical analysis, the standard SVMs had even delivered a weaker performance in out-of-sample predictions than did the GLMs using the conventional threshold, not to mention those using the alternative one. This implies that the standard SVMs might be more sensitive to imbalanced or rare event data than Akbani et al. [2004] had suggested.

In addition, under the presence of moderate class imbalance, the standard SVMs tend to have low out-of-sample sensitivity – an undesirable characteristic in the modelling of rare events. As mentioned in the introduction, rare events (or black swan events) often have a major impact in the real world, implying that the cost of false negatives (i.e., misclassifying the events as non-events) could be much greater than the cost of false positives. This is especially true in forecasting financial crises, or in the context of the MID data – forecasting military conflicts. In the latter case, using a classifier that has low sensitivity would most probably lead to signals indicating the break of war be ignored. Such outcome is not only undesirable from a military point of view, but also from a social one because of the high costs associated with warfare [Beck et al., 2000].

As for the cost-sensitive SVMs, the results from the simulation study and empirical analysis show that they achieved a better classification performance than the GLMs and the standard SVMs under the presence of moderate class imbalance. However, as the events become rarer, the out-of-sample classification performance of the cost-sensitive SVMs dropped sharply due to the rising number of false positives. Although the cost-sensitive SVMs have much higher sensitivity than the standard versions – a characteristic that could be beneficial in forecasting military conflicts – the method also created a large number of false positives when applied to the the MID data. This raises some concerns about using the cost-sensitive SVMs in conflict research. Meanwhile, the tendency of overfitting occurs in some kernelised cost-sensitive SVMs, especially those with Gaussian and polynomial kernels. This issue might also cast some doubts on using such methods in other real-world applications.

7.1.2 The reliability issue of probabilities produced by Platt scaling

The results from the empirical analysis in this study indicate that the probabilistic outputs produced by Platt scaling might have a reliability issue, as these probabilities are not consistent with the nominal classification performance of the SVMs. Additionally, the results from ROC/AUC and from the precision-recall curves also show that these probability estimates are inferior to that produced by the GLMs.

One can argue that it was the Platt scaling and the inconsistency it induced that caused the weak performance of the SVMs in ROC/AUC and prediction-recall curves. Although we can not entirely rule out the possibility of this scenario, it might still be imprudent to attribute the weak performance of SVMs to Platt scaling alone, since the inferior probability estimates shown in the ROC and the precision-recall curves are most likely caused by the rare events presented in the data set, which affected both the base classifier (i.e., the SVMs) and the calibrator (i.e, the Platt scaling).

As mentioned in Section 2.3.4, Platt scaling is essentially a calibrator that takes the SVM scores, $\hat{f}_i$, as inputs, and then transformed $\hat{f}_i$ to probabilities by fitting a sigmoid to the training set $(\hat{f}_i, t_i)$. Note that the scores $\hat{f}_i$ in this training set are selected through cross-validation. As described in Section 3.2.3, the SVM has a tendency of favouring the majority class when applied to a data set in which the imbalanced classes or rare events are presented. In the case of MID data, this would lead to only those $\hat{f}_i$ with a negative sign (which represent observations that were classified as non-events) are passed on to the training set $(\hat{f}_i, t_i)$. Fitting a sigmoid to this training set would most certainly yield a set of inferior probability predictions that are not capable of predicting a single event – "garbage in, garbage out". Therefore, the inferior probability estimates shown in the ROC and the precision-recall curves have more to do with the base classifiers' inability of handling imbalanced data, as implied by the results of the simulation study.

However, we must also acknowledge that the Platt scaling might also suffer from imbalanced classes. In Section 2.3.4, we mentioned that the method's use of the sigmoid function to calibrate probabilities (or scores produced by the base classifier) is only justified if the scores within each class are Gaussian distributed with equal variance. For the MID data, which has an event rarity around 0.34 percent, the equal variance requirement is unlikely to be satisfied. We had also mentioned that the Platt scaling is most effective when the score distortions have a sigmoidal shape. However, as shown in Figure B.1 in the appendix, this is not the case for the MID data. Thus, it is not unexpected that the probability estimates produced by Platt scaling become inferior to those produced by the GLMs.

Therefore, regarding the question whether we can use the probabilistic outputs produced by Platt scaling in rare event modelling, the results of this study suggest that such outputs might be unreliable if the base classifier is sensitive to imbalanced dataset. Considering the fact that the SVM outputs are generally hard to interpret and that statistical inference is an important part of quantitative social science research, it would be appropriate to avoid using the SVM and the Platt scaling in building a prediction model for rare events.

7.2 Methodological limitations

There are several methodological limitations in this study. Firstly, the data sets in the simulation study were all generated using the LR as the true model. In other words, I did not consider other options, such as the probit model and linear discriminant analysis, in simulating rare event data. Additionally, rare event data could also be simulated using stochastic methods such as the Markov process [Bucklew, 2004]. For this reason, the classification performance of the GLMs and the SVMs on data simulated using the above-mentioned techniques might differ from the results presented in this thesis.

Secondly, the simulated data sets in this study assumed a linear relationship between the outcome variable and the predictors. The case of non-linear relationship is not included in this study. As mentioned in Section 2.3.3, the SVMs, especially the kernelised versions, might achieve a better classification performance on data characterised by such non-linear relationship.

Thirdly, even though the thesis aims to provide evidence for a wider application of the use of SVMs in social science research, the reader should still interpret the results of the empirical analysis with caution. Such caution is also advised for future extrapolations of the results of the empirical analysis to other social science disciplines, as the empirical analysis of this study is based upon a data set typical for peace and conflict research. And the relationships between the outcome variable and the predictors presented in the MID data set might not be necessarily representative for data in other fields such as economics and finance. In addition, the signal-to-noise ratio, which is a measure of the amount of information contained in the data, might also vary in data typical to different social science disciplines.

Chapter 8

Conclusions

Using both simulated and real-world data, this thesis is intended to provide more insight to the research concerning the use of machine learning techniques in social science. More specifically, I evaluated the classification performance of the SVMs under the presence of class imbalance and rare events, and compared it with the classification performance of the traditional GLMs. The main conclusion of this study is that the use of SVMs – both the standard version and the cost-sensitive version – in rare event modelling could not be motivated in any case. There are several reasons for this conclusion. Firstly, the standard SVMs showed no better performance than the GLMs in classifying rare events. The results from both the simulation study and the empirical analysis also showed that the standard SVMs have a relatively low sensitivity, a characteristic that brings more harm than benefit to the prediction of rare events in real applications.

Secondly, as in the case of the standard SVMs, the cost-sensitive SVMs could not be a satisfactory classifier in rare event modelling either – for the reason that they generally tend to overfit the training data, leading to a poor out-of-sample precision and hence a poor F1 score. Other than overfitting, cost-sensitive SVMs also result in a large number of false positives, which occur in connection with severe and extreme class imbalance.

Thirdly, as a machine learning classifier, the SVM has already an inherent drawback in model interpretability in comparison to the GLMs. In addition, the fact that the SVMs do not produce a probabilistic output makes issue with interpretability even more problematic, as the probability estimates are important for quantitative analysis in social science. The Platt scaling is designed for mitigating this lack of probabilistic output. However, as the empirical analysis of this thesis suggests, the predicted probabilities produced by Platt scaling are not consistent with the nominal evaluation metrics and might suffer from class imbalance as well. Hence, such probabilities are not reliable. The reliability issue of probabilities produced by Platt scaling could be a further argument against the use of SVMs in rare event modelling.

In sum, the results from both the simulation study and the empirical analysis provided little support for the use of the SVMs in rare event modelling in general, not to mention in social science research. Perhaps a better way of addressing the rare event bias is to create a more balanced data set, by applying the sampling method described in the Section 3.3. However, the results of this thesis do not suggest that we should completely abandon the internal methods. They are rather a call for more research focused on the development of novel methods that addresses the problems associated with rare events, as well as a call for more studies concerning the application of machine learning methods in the field of social science.

References

Agresti, A. (2012). *Categorical data analysis* (3rd ed.). Hoboken, N.J.: Wiley.

Agresti, A. (2015). *Foundations of linear and generalized linear models.* Hoboken, New Jersey: John Wiley & Sons.

Akbani, R., Kwek, S., & Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. In *European conference on machine learning* (pp. 39–50).

Batuwita, R., & Palade, V. (2010). Efficient resampling methods for training support vector machines with imbalanced datasets. In *The 2010 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).

Batuwita, R., & Palade, V. (2013). Class imbalance learning methods for support vector machines. In H. He & Y. Ma (Eds.), *Imbalanced learning: Foundations, algorithms, and applications.* Wiley-IEEE Press.

Beck, N., King, G., & Zeng, L. (2000). Improving quantitative studies of international conflict: A conjecture. *American Political science review*, 21–35.

Bennett, D. S., & Stam, A. C. (2000). Research design and estimator choices in the analysis of interstate dyads: When decisions matter. *Journal of conflict resolution*, *44*(5), 653–685.

Bishop, C. M. (2006). *Pattern recognition and machine learning.* New York, NY: Springer.

Brodersen, K. H., Ong, C. S., Stephan, K. E., & Buhmann, J. M. (2010). The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition* (p. 3121-3124).

Bucklew, J. A. (2004). *Introduction to rare event simulation.* New York: Springer.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Cristianini, N., & Shawe-Taylor, J. (2000). Linear learning machines. In *An introduction to support vector machines and other kernel-based learning methods* (p. 9–25). Cambridge University Press.

Fernández, A., Krawczyk, B., García, S., Galar, M., Herrera, F., & Prati, R. C. (2018). *Learning from imbalanced data sets.* Springer.

Friedman, J., Hastie, T., & Tibshirani, R. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). New York: Springer.

Ghosn, F., Palmer, G., & Bremer, S. A. (2004). The mid3 data set, 1993—2001: Procedures, coding rules, and description. *Conflict management and peace science*, *21*(2), 133–154.

Greene, W. H. (2018). *Econometric analysis* (Global edition; 8th ed.). Harlow, Essex: Pearson.

Hain, D., & Jurowetzki, R. (2020). *Introduction to rare-event predictive modeling for inferential statisticians – a hands-on application in the prediction of breakthrough patents.* (https://arxiv.org/abs/2003.13441[Accessed: 2020-12-27])

Hand, D. J. (2006). Classifier technology and the illusion of progress. *Statistical science*, 1–14.

Herbrich, R. (2002). *Learning kernel classifiers : theory and algorithms.* Cambridge, Mass.: MIT Press.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With Applications in R* (1st ed.). New York: Springer.

Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2020). *Fundamentals of machine learning for predictive data analytics : algorithms, worked examples, and case studies* (2. ed. ed.). Cambridge, Massachusetts: The MIT Press.

King, G., & Zeng, L. (2001a). Explaining rare events in international relations. *International Organization*, *55*(3), 693–715.

King, G., & Zeng, L. (2001b). Logistic regression in rare events data. *Political analysis*, *9*(2), 137–163.

Kull, M., Silva Filho, T. M., Flach, P., et al. (2017). Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, *11*(2), 5052–5080.

Levy, P. S., & Lemeshow, S. (2008). *Sampling of populations : methods and applications* (4th ed.). Hoboken, N.J.: Wiley.

McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models* (2nd ed.). London: Chapman & Hall.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective.* Cambridge, MA: MIT press.

Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning* (pp. 625–632).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Édouard Duchesnay (2020a). *1.16. Probability calibration.* (https://scikit-learn.org/stable/modules/calibration.html[Accessed: 2021-02-21])

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Édouard Duchesnay (2020b). *1.4. Support Vector Machines.* (https://scikit-learn.org/stable/modules/svm.html[Accessed: 2020-12-27])

Platt, J. (1998). *Sequential minimal optimization: A fast algorithm for training support vector machines* (Tech. Rep. No. MSR-TR-98-14). Redmond, Washington, United States: Microsoft Research.

Platt, J. (2000). Probabilities for sv machines. In A. J. Smola, P. Bartlett, B. Schölkopf, & D. Schuurmans (Eds.), *Advances in large margin classifiers.* Cambridge, MA: The MIT Press.

Prati, R. C., Batista, G. E., & Monard, M. C. (2011). A survey on graphical methods for classification predictive performance evaluation. *IEEE Transactions on Knowledge and Data Engineering*, *23*(11), 1601–1618.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Taleb, N. N. (2007). *The black swan: The impact of the highly improbable* (Vol. 2). Random house.

Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, *1*(Jun), 211–244.

Tsang, I. W., Kwok, J. T., & Cheung, P.-M. (2005). Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research*, *6*(Apr), 363–392.

Van der Paal, B. (2014). *A comparison of different methods for modelling rare events data* (Unpublished master's thesis). Ghent University.

Veropoulos, K., Campbell, C., & Cristianini, N. (1999). Controlling the sensitivity of support vector machines. In *Proceedings of the international joint conference on AI* (Vol. 55, p. 60).

Wackerly, D. D., Mendenhall, W. I., & Scheaffer, R. L. (2008). *Mathematical statistics with applications* (7th ed.). Southbank: Thomson Learning.

Wasserman, L. (2004). *All of statistics : a concise course in statistical inference.* New York: Springer.

Xu, X., Mao, Y., Xiong, J., & Zhou, F. (2007). Classification performance comparison between RVM and SVM. In *2007 International Workshop on Anti-Counterfeiting, Security and Identification (ASID)* (pp. 208–211).

Zadrozny, B., & Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 694–699).

Zeng, L. (1999). Prediction and classification with neural network models. *Sociological methods & research*, *27*(4), 499–524.

Appendix A

Additional tables

Table A.1: Descriptive statistics for the full MID data

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
Militarised dispute	303,814	0.0034	0.0585	0	0	0	1
Contiguous	303,814	0.0414	0.1991	0	0	0	1
Allies	303,814	0.0903	0.2866	0	0	0	1
Foreign policy	303,814	0.7726	0.1736	−0	0.7	0.9	1
Balance of Power	303,814	0.3644	0.2989	0.0002	0.0949	0.6018	1.0000
Max. democracy	303,814	3.2815	7.2791	−10	−6	10	10
Min. democracy	303,814	−5.0347	5.6620	−10	−9	−5	10
Max. trade	303,814	−4.8147	2.5843	−8.6840	−8.6840	−2.7884	0.2954
Min. trade	303,814	−5.7591	2.5303	−9.5361	−9.5361	−3.7558	−0.7376
Year since dispute	303,814	17.6841	11.5093	0	8	26	46
Major Power	303,814	0.0831	0.2760	0	0	0	1

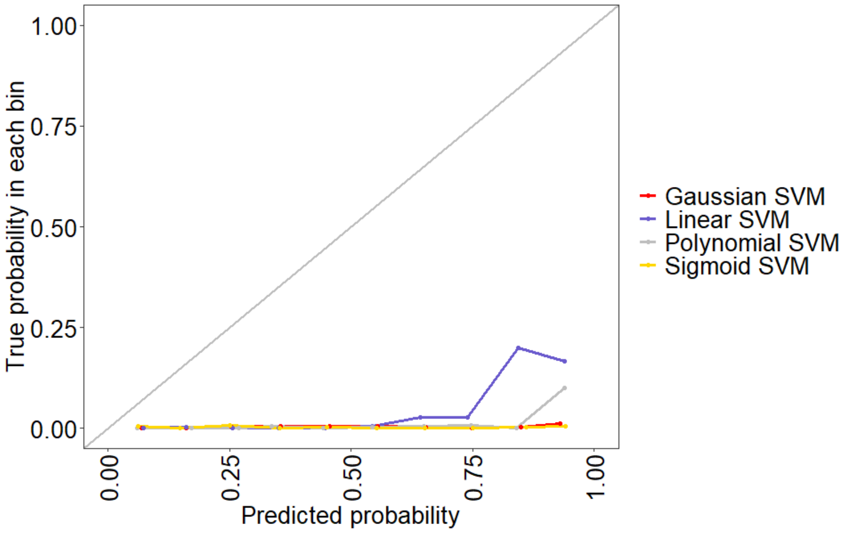
Table A.2: Model for Militarised Interstate Dispute

	Dependent variable:		
	Militarised Interstate Dispute		
	LR	*ReLogit*	*Cloglog*
	(1)	(2)	(3)
Contiguous	3.686^{***}	3.665^{***}	3.557^{***}
	(0.302)	(0.303)	(0.304)
Allies	-0.642^{*}	-0.614^{*}	-0.591
	(0.385)	(0.385)	(0.371)
Foreign policy	0.387	0.358	0.388
	(0.860)	(0.860)	(0.825)
Balance of Power	0.733	0.747	0.732
	(0.517)	(0.517)	(0.489)
Max. democracy	0.083^{***}	0.082^{***}	0.076^{***}
	(0.021)	(0.021)	(0.020)
Min. democracy	-0.095^{***}	-0.091^{***}	-0.087^{***}
	(0.026)	(0.026)	(0.025)
Max. trade	0.090	0.104	0.118
	(0.224)	(0.224)	(0.218)
Min trade	-0.077	-0.093	-0.108
	(0.232)	(0.232)	(0.226)
Years since dispute	-0.078^{***}	-0.076^{***}	-0.076^{***}
	(0.024)	(0.024)	(0.023)
Major power	1.950^{***}	1.935^{***}	1.794^{***}
	(0.329)	(0.329)	(0.312)
Constant	-7.468^{***}	-7.375^{***}	-7.387^{***}
	(0.836)	(0.837)	(0.813)
Observations	21,267	21,267	21,267
Log Likelihood	-328.019	-328.019	-329.285
Akaike Inf. Crit.	678.037	678.037	680.570

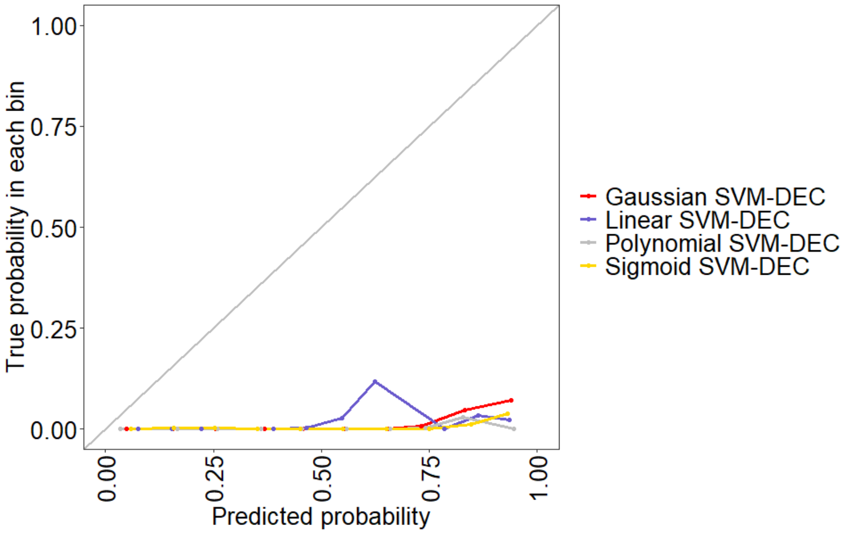
Note: Parameters are estimated based upon training data (70% of the observations). Numbers in brackets are robust standard errors.
*p<0.1; **p<0.05; ***p<0.01

Appendix B

Additional figures



(a) Scores produced by the standard SVMs, normalised



(b) Scores produced by the cost-sensitive SVMs, normalised

Figure B.1: Reliability diagram of the uncalibrated probabilities (i.e., normalised classification scores) produced by the standard SVMs and the cost-sensitive SVMs.