RISK-AVERSE MULTI-CLASS SUPPORT VECTOR MACHINES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

By
Ayşenur Karagöz
December 2018

Risk-Averse Multi-Class Support Vector Machines
By Ayşenur Karagöz
December 2018

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Özlem Çavuş İyigün(Advisor)

A. Ercüment Çiçek

Sinan Gürel

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

RISK-AVERSE MULTI-CLASS SUPPORT VECTOR MACHINES

Ayşenur Karagöz
M.S. in Industrial Engineering
Advisor: Özlem Çavuş İyigün
December 2018

A classification problem aims to identify the class of new observations based on the previous observations whose classes are known. It has many applications in a variety of disciplines such as medicine, finance and artificial intelligence. However, presence of outliers and noise in previous observations may have significant impact on the classification performance. Support vector machine (SVM) is a classifier introduced to solve binary classification problems under the presence of noise and outliers. In the literature, risk-averse SVM is shown to be more stable to noise and outliers compared to the original SVM formulations. However, we often observe more than two classes in real-life datasets. In this study, we aim to develop risk-averse multi-class SVMs following the idea of risk-averse binary SVM. We use risk measures, VaR and CVaR, to implement risk-aversion to multi-class SVMs. Since VaR constraints are nonconvex in general, SVMs with VaR constraints are more complex than SVMs with CVaR. Therefore, we propose a strong big-M formulation to solve multi-class SVM problems with VaR constraints efficiently. We also provide a computational study on the classification performance of the original multi-class SVM formulations and the proposed risk-averse formulations using artificial and real-life datasets. The results show that multi-class SVMs with VaR are more stable to outliers and noise compared to multi-class SVMs with CVaR, and both of them are more stable than the original formulations.

Keywords: Support vector machines, multi-class classification problem, risk-aversion, Conditional Value-at-Risk, Value-at-Risk.

ÖZET

RİSKTEN KAÇINAN ÇOK SINIFLI DESTEK VEKTÖR MAKİNELERİ

Ayşenur Karagöz
Endüstri Mühendisliği, Yüksek Lisans
Tez Danışmanı: Özlem Çavuş İyigün
Aralık 2018

Sınıflandırma problemi, sınıfı bilinen daha önceden gözlemlenmiş örneklere dayanarak, yeni örneğin sınıfının tespit edilmesini amaçlamaktadır. Bu problemin tıp, finans ve yapay zeka gibi farklı disiplinlerde pek çok uygulaması bulunmaktadır. Ancak, önceden gözlemlenmiş örneklerde uç değerler ve gürültü olması, başarım performansını önemli ölçüde etkileyebilmektedir. Destek vektör makinesi (DVM), uç değerler ve gürültü barındıran iki sınıflı sınıflandırma problemlerini çözmek için geliştirilmiştir. Literatürde, riskten kaçınan DVM'nin uç değerlere ve gürültüye asıl DVM formülasyonlarına göre daha kararlı olduğu gösterilmiştir. Fakat, gerçek veri setlerinde, ikiden çok sınıflı sınıflandırma problemleriyle daha çok karşılaşılmaktadır. Bu çalışmada, riskten kaçınan iki sınıflı DVM takip edilerek, riskten kaçınan çok sınıflı DVM'ler geliştirilmesi amaçlanmıştır. Riskten kaçınma, çok sınıflı DVM'lere, riske maruz değer ve koşullu riske maruz değer risk ölçütleri kullanılarak dahil edilmiştir. Riske maruz değer kısıtları genel olarak dış bükey olmadıkları için, riske maruz değer kısıtı eklenmiş DVM'ler, koşullu riske maruz değer eklenmiş DVM'lere göre daha karmaşıktır. Bu nedenle, riske maruz değer kısıtı eklemiş çok sınıflı DVM'leri etkin bir şekilde çözmek için, güçlü büyük M formülasyonu önerilmiştir. Bununla birlikte, asıl çok sınıflı DVM'lerin ve riskten kaçınan DVM'lerin gerçek ve yapay veri setleri üzerindeki başarım performansını gösteren bir çalışma sunulmuştur. Sonuçlar, riske maruz değer kısıtı eklenmiş çok sınıflı DVM'lerin veri setlerindeki uç değerlere ve gürültüye koşullu riske maruz değer eklemiş DVM'lere göre daha kararlı olduğunu, ve riskten kaçınan DVM'lerin asıl formülasyonlara göre daha kararlı olduğunu göstermektedir.

Anahtar sözcükler: Destek vektör makineleri, çok sınıflı sınıflandırma problemi, riskten kaçınma, koşullu riske maruz değer, riske maruz değer.

Acknowledgement

I would like to express my gratitude to my advisor Asst. Prof. Özlem Çavuş İyigün for her continuous support in my research. Her guidance and passion motivates me throughout my studies. Her immense knowledge and expert advice made this work possible. It has been a privilege for me to be her student.
I would like to thank to Prof. Sinan Gürel and Asst. Prof. Abdullah Ercüment Çiçek for their precious time to read and review my thesis.
I also would like to extend my gratitude to Assoc. Prof. Cem İyigün for his continuous support, guidance and significant contributions.
Finally, I would like to thank my family for their ceaseless support and patience.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Classification problem has applications in a variety of disciplines such as medicine, finance, marketing, computer vision and artificial intelligence [2]. It aims to identify the class of new observation based on a training set containing samples whose classes are known [3]. For binary classification where there are two classes, diagnosing a patient as cancer and non-cancer under the guidance of existing data can be given as an example [4]. The applications can be extended to other fields such as fraud detection, spam filtering and etc, see [5], [6], [7]. When datasets have more than two categories, it is called multi-class classification problem.

In most classification datasets, classes contain points that are far away from the majority of observations [8]. These points are referred as outliers which can be result of experimental error or high variability in class distribution [9]. Outliers can have a misleading influence on predictive models since they are extreme observations [8]. Noise in datasets is another issue affecting the classification performance. Class noise denotes the samples with wrong labels while attribute noise denotes the errors in attribute values [10]. Therefore, noise only gives misleading information and should be eliminated from the dataset, on the other hand, outliers may give information about the class distribution [10]. Another important issue in datasets can be stated as imbalanced distribution of classes. When sample size of a class is considerably small compared to the other classes, the designed model does not learn the small class adequately and its predictive

performance is dominated by the other classes.

Support vector machine (SVM) is a classifier introduced to solve binary classification problems by minimizing the training error of the single worst sample [11], [12]. Therefore, it is sensitive to outliers as this single worst error can be a result of measurement error [13]. In the literature, several approaches are proposed for binary SVM to handle outliers. Robust SVM and center SVM use the class centers together with samples to build a classifier that is less sensitive to outliers [14], [15]. The fuzzy SVM is a reformulation of the original model which introduces fuzzy membership to each data point in order to allow different contributions of data points to the construction of the classifier so that the effect of outliers can be reduced [16]. Yet another approach involves using financial risk measures Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR) in SVM. In the literature, it is shown that risk-averse binary SVM provides stability to outliers unlike the original formulation [12], [13].

Considering the stability of risk-averse binary SVM to outliers, we aim to extend this approach to multi-class SVM (MSVM). In this study, only Weston and Watkins multi-class SVM (WW-MSVM) and Crammer and Singer multi-class SVM (CS-MSVM) which follow *all-together* scheme are taken into consideration. For this purpose, we first implement CVaR to multi-class SVM models following the interpretation of ν-SVM as CVaR minimization [12]. CVaR is shown to be a convex function [17]. Proceeding from this result and convex quadratic nature of SVM, CVaR MSVM models are required to solve convex quadratic programming. Unlike CVaR MSVM, VaR MSVM results non-convex programming. In the literature, for finite sample space, it is shown that VaR constraints can be reformulated as big-M constraints [18] due to chance constraint interpretation of VaR constraints [19]. Therefore, implementation of VaR to MSVM results mixed integer quadratic programming which is computationally intractable. To solve VaR MSVM, we propose a strong big-M formulation using the valid inequalities given in [20]. Then, we compare the performance of the proposed strong big-M formulation with branch and cut decomposition algorithm presented by Luedtke [20] and regular big-M formulation in terms of optimality gap and objective value. The comparative study shows that strong big-M formulation outperforms other methods in each criterion. Also, we analyze the behavior of CVaR MSVM and

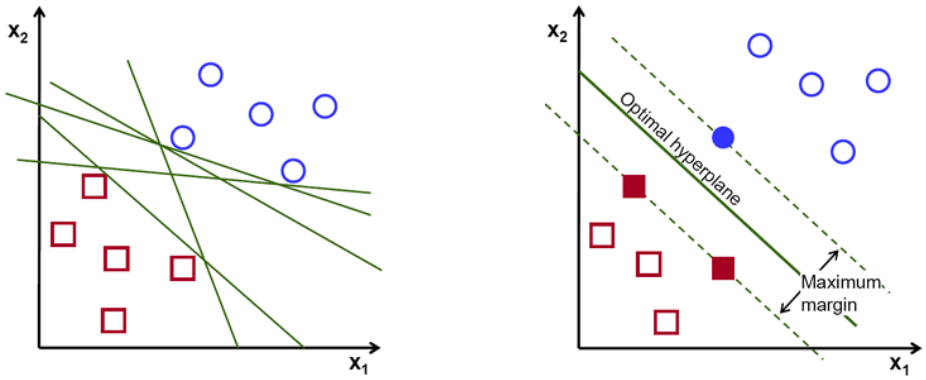
VaR MSVM in presence of noise, outliers and imbalanced class distribution under different class probabilities and different levels of risk-aversion. First observation is that class probability affects the performance of all models. However, risk-averse MSVM is less sensitive to class probability compared to the original model. Moreover, when risk-averse MSVM models are examined, it is seen that CVaR MSVM is more stable to class probability. Also, we observe that VaR MSVM is more responsive to the risk-aversion level as it determines the upper quantile to be ignored. The analysis indicates that risk-averse MSVM models are more robust to noise and outliers while imbalanced class distributions does not have a significant impact on the performance of risk-averse MSVM models. Finally, we test the models on real-life datasets to see whether the results of the geometrical analysis coincide with the real-life examples.

The structure of the thesis is as follows: In Chapter 2, we provide background for SVM, financial risk measures VaR and CVaR together with binary and multi-class SVM models. In Chapter 3, we present the relation between VaR, CVaR and SVM and extend this relation to multi-class case. In Chapter 4, we give solution methodology for solving VaR multi-class SVMs. In Chapter 5, we describe the artificial datasets and compare performance of CVaR and VaR multi-class SVMs on different problem settings. Finally, in Chapter 6, we give our concluding remarks.

Chapter 2

Literature Review

Motivated by the theory of generalization of learning algorithms [21], [22], [23], Guyon et al. [24] show that the maximum margin hyperplane minimizes the generalization error bound, in other words, error in prediction of previously unobserved data. Following these results, Support Vector Machine (SVM) is introduced as a maximum margin classifier that was originally designed to solve binary classification problems [25]. SVM constructs a separating hyperplane by maximizing the minimum distance of the samples to this hyperplane [3].



(a) Separating hyperplanes for binary classification problem

(b) Maximum margin hyperplane constructed by SVM

Figure 2.1: Separation problem and the optimal hyperplane constructed by SVM (reproduced from [1])

As demonstrated by Figure 2.1a, it is possible to construct infinitely many hyperplanes that separate two classes. SVM solves optimization problem of finding optimal hyperplane parameters that maximize margin width to discriminate classes. The term margin corresponds to minimum distance of the samples to the separating hyperplane. Geometric meaning of the term can be better understood in Figure 2.1b. Here, the samples on dotted lines are denoted as support vectors. In other words, the samples whose distance to the optimal separating hyperplane is the minimum among given dataset are referred as support vectors. The distance between hyperplanes passing through support vectors is denoted as maximum margin. Hence, margin width equals to half of maximum margin.
For given training dataset $\{(x_1, y_1), \ldots, (x_l, y_l)\}$, observation set is defined as $I = \{1, \ldots, l\}$ where sample (instance) and corresponding class information are represented as $x_i \in \mathbb{R}^n, y_i \in \{-1, +1\}$ for $i \in I$, respectively. In this setting, assuming blue circles belong to class $+1$ and red squares to class -1, SVM solves the problem below:

$$\max_{\substack{w \in \mathbb{R}^n \\ b \in \mathbb{R}}} \min_{i \in I} \quad \frac{y_i(w^T x_i + b)}{\|w\|_2}. \tag{2.1}$$

Resulting $(w, b) \in \mathbb{R}^n \times \mathbb{R}$ are optimal hyperplane parameters such that w and b denote normal vector and intercept of the separating hyperplane, respectively. Then, for blue circles $(w^T x_i + b) > 0$, similarly, for red squares $(w^T x_i + b) < 0$. Objective function denotes the distance of i-th data point to the hyperplane where the expression $\|w\|_2$ stands for $\sqrt{w^T w}$. By solving this optimization problem, hyperplane parameters which maximize the minimum margin are obtained. To linearize the objective function, an auxiliary variable s is introduced to replace $\min_{i \in I} y_i(w^T x_i + b)$, as in [26]. Then optimization problem takes the form:

$$\begin{aligned} \max \quad & \frac{s}{\sqrt{w^T w}} \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq s, \ i \in I \\ & w \in \mathbb{R}^n, \ b \in \mathbb{R}. \end{aligned} \tag{2.2}$$

Without loss of generality, s can be scaled to 1 and maximizing $\frac{1}{\sqrt{w^T w}}$ is equivalent

to minimizing $w^T w$. Then Problem (2.2) can be rearranged as follows:

$$\begin{aligned} \min \ & \frac{1}{2} w^T w \\ \text{s.t.} \ & y_i(w^T x_i + b) \geq 1, \ i \in I \\ & w \in \mathbb{R}^n, \ b \in \mathbb{R}. \end{aligned} \tag{2.3}$$

The Problem (2.3) is called hard-margin SVM [11]. With this optimization problem, the margin width is forced to be $\frac{1}{\sqrt{w^T w}}$ and instance i is correctly classified if $y_i(w^T x_i + b) > 0$. The factor $\frac{1}{2}$ is included for convenience in Karush-Kuhn-Tucker conditions. As the objective function is quadratic and it is subject to linear constraints, (2.3) is in the form of constrained convex optimization problem, therefore, can be solved by Lagrange multiplier method [27]. To solve this problem, Lagrange multipliers $\alpha_i \in \mathbb{R}_+, \ i \in I$ are introduced to the linear inequality constraint where $\mathbb{R}_+ = [0, +\infty)$. Then the Langrangian function is:

$$\begin{aligned} L(w, b, \alpha) &= \frac{1}{2} w^T w - \sum_{i \in I} \alpha_i [y_i(w^T x_i + b) - 1] \\ &= \frac{1}{2} w^T w - \sum_{i \in I} \alpha_i y_i (w^T x_i + b) + \sum_{i \in I} \alpha_i. \end{aligned} \tag{2.4}$$

When partial derivatives with respect to w and b are set to 0, we obtain:

$$\begin{aligned} \frac{\partial L}{\partial w} &= w - \sum_{i \in I} \alpha_i y_i x_i = 0, \\ \frac{\partial L}{\partial b} &= \sum_{i \in I} \alpha_i y_i = 0. \end{aligned} \tag{2.5}$$

From the equations above, we get the expression $w = \sum_{i \in I} \alpha_i y_i x_i$. Substituting this in the Lagrangian, we obtain dual problem as:

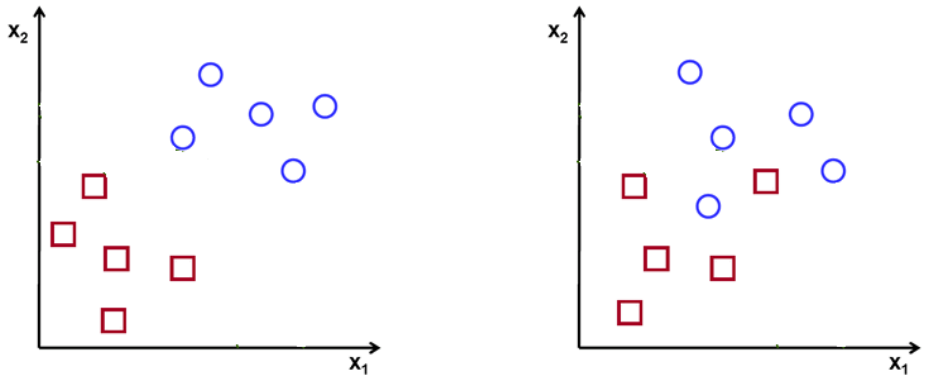
$$\begin{aligned} \max \ & \sum_{i \in I} \alpha_i - \frac{1}{2} \sum_{i \in I} \sum_{j \in I} \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \text{s.t.} \ & \sum_{i \in I} \alpha_i y_i = 0 \\ & \alpha \in \mathbb{R}_+^l. \end{aligned} \tag{2.6}$$

By the complementary slackness condition, when $\alpha_i > 0$ in dual problem, the corresponding constraint of primal problem strictly holds which means $y_i(w^T x_i + b) =$

1. Support vectors are the observations that satisfy the constraint as equality, hence, Lagrange multipliers corresponding to support vectors are strictly positive, i.e, $\alpha_i > 0$. As $w = \sum_{i \in I} \alpha_i y_i x_i$ from (2.5), only support vectors contribute to determination of the optimal hyperplane parameters.

2.1 Binary SVM

Boser et al. [11] develop hard-margin SVM (2.3) where each sample is required to be classified correctly without margin violation. In other words, hard-margin SVM constructs the separating hyperplane in a way that all samples are on the right side of the hyperplane and their distances to this hyperplane should be greater than or equal to the margin width. However, this can only be accomplished if given dataset is linearly separable. Consider two sets: one consists of red points and the other consists of blue points. These sets are linearly separable if it is possible to find at least one hyperplane such that all red points is on one side of the hyperplane and all blue points is on the other side. Linear separability can be better understood in Figure 2.2 .



(a) Linearly separable dataset (b) Linearly inseparable dataset

Figure 2.2: Linear separability of given dataset (reproduced from [1]).

Therefore, hard-margin SVM is feasible only when training dataset is linearly

separable. To overcome this limitation, different SVM formulations are introduced in the literature. In this section, variants of SVM are presented.

2.1.1 Soft-Margin SVM

Overfitting and underfitting terms are used to describe the modeling error. Overfitting occurs when built model memorizes or fits too closely to the given training dataset. Resulting from this, it may give poor predictive performance on previously unobserved data while overperforms in training dataset. On the contrary, underfitting occurs when the built model does not learn the training dataset. Consequently, it will produce poor predictive performance on both training dataset and newly acquired data. Therefore, when hard-margin SVM problem in (2.3) is feasible, overfitting can be an issue. In order to control the sensitivity of SVM to outliers and deal with overfitting, Cortes and Vapnik [25] extended the hard-margin SVM to non-separable case by introducing slack variables $\xi \in \mathbb{R}^l_+$, i.e, $\xi_i \geq 0,\ i \in I$ and error penalization parameter $C > 0$:

$$\begin{aligned} \min\ & \frac{1}{2} w^T w + C \sum_{i \in I} \xi_i \\ \text{s.t.}\ & y_i(w^T x_i + b) \geq 1 - \xi_i,\ i \in I \\ & w \in \mathbb{R}^n,\ b \in \mathbb{R},\ \xi \in \mathbb{R}^l_+. \end{aligned} \tag{2.7}$$

Here, for small values of C, larger margin hyperplane is constructed. However, if C is too small, soft-margin SVM formulation underfits the training dataset which shows that the classifier is not trained adequately. With increasing value of C, the model fits too closely to the training dataset which may lead to overfitting. Consequently, penalization parameter C has a significant impact on the predictive performance by means of preventing overfitting and underfitting with controlling the trade off between training error and margin width.

When $0 < \xi_i \leq 1$, the data point i is on the correct side of the separating hyperplane but its distance to the optimal hyperplane is less than the margin width implying a margin violation. When $\xi_i > 1$, the data point i is misclassified. Therefore, soft-margin SVM allows misclassifications, however, by introducing

parameter C, both margin violation and misclassification are penalized.

2.1.2 ν-SVM

ν-SVM is a variant of soft-margin SVM that uses parameter $\nu \in (0, 1]$ instead of C. Parameter ν is introduced to eliminate the effect of parameter C in soft-margin formulation and it enables to control the number of support vectors effectively [28]. Recall that only support vectors contribute to determination of the optimal hyperplane parameters by (2.5). Resulting from that, control on support vectors provides control on construction of the separating hyperplane. ν-SVM model is given as follows:

$$\begin{aligned} \min \ & \frac{1}{2} w^T w - \nu\rho + \frac{1}{l} \sum_{i \in I} \xi_i \\ \text{s.t.} \ & y_i(w^T x_i + b) \geq \rho - \xi_i, \ i \in I \\ & w \in \mathbb{R}^n, \ b \in \mathbb{R}, \ \xi \in \mathbb{R}^l_+, \rho \in \mathbb{R}_+. \end{aligned} \tag{2.8}$$

Here ρ plays a role in determination of margin width such that when slack variables $\xi_i = 0, \ i \in I$, margin width equals to $\frac{\rho}{\sqrt{w^T w}}$. Crisp and Burges [29] report that the constraint $\rho \in \mathbb{R}_+$ is redundant. When the optimal solution of ν-SVM results $\rho > 0$, Schölkopf et al. [28] show soft-margin SVM with $C = \frac{1}{\rho l}$ constructs the same separating hyperplane as ν-SVM. Furthermore, parameter ν is an upper bound on the fraction of training error, i.e, $\frac{\#misclassifications}{\#samples} \leq \nu$ and a lower bound on the fraction of support vectors, i.e, $\frac{\#support\ vectors}{\#samples} \geq \nu$. Therefore, in real applications, ν-SVM is potentially more effective compared to soft-margin SVM as it allows more control in the training phase [28].

2.2 Multi-Class SVM

Up to this point, proposed SVM approaches consider only binary classification problems. However, we mostly observe more than two classes in real-life problems. Different from binary classification problems, multi-class problems are given

in the following form. Let $\{(x_1, y_1), \ldots, (x_l, y_l)\}$ be training dataset with size l, observation set $I = \{1, \ldots, l\}$ and class set $M = \{1, \ldots, k\}$ where sample and corresponding class information are provided as $x_i \in \mathbb{R}^n, y_i \in M$ for $i \in I$, respectively.

In the literature, it is observed that there are two main approaches to extend binary SVM classifier to multi-class case. First approach divides the multi-class dataset into partitions such that several binary classification problems can be constructed then it solves these problems separately. The second approach follows the *all-together* scheme to solve the problem. In *all-together* scheme, given a dataset of $k > 2$ classes, the designed model considers all dataset in one optimization problem and constructs k separating hyperplanes. The objective of this scheme is to maximize margin width with respect to each separating hyperplane simultaneously. Two different methods are proposed by following the first approach: *one against one* [30] and *one against all* [31], while the second one proposes two different methods: formulation of Weston and Watkins [32] and formulation of Crammer and Singer [33].

2.2.1 One Against All Method

Based on the soft-margin SVM problem, the *one against all* method creates k binary classification problems [31]. To discriminate the class $m \in M$ from the remaining $k-1$ classes, similar to binary SVM, the proposed method reconstructs the dataset by relabeling the samples belonging to class m as $+1$ and remaining ones as -1. Then the soft-margin classifier m, which produces the separating hyperplane between class m and the rest, is trained on this reconstructed dataset. The resulting optimal hyperplane parameters are denoted as $w_m \in \mathbb{R}^n$ and $b_m \in \mathbb{R}$ which are normal vector and intercept of the hyperplane, respectively. Note that this process is repeated for each class $m \in M$. Thus, the *one against all* SVM solves the following soft-margin problem with error penalization parameter $C > 0$ to discriminate class m from the rest as follows:

$$
\begin{aligned}
\min \ & \frac{1}{2} w_m^T w_m + C \sum_{i \in I} \xi_i^m \\
\text{s.t.} \ & w_m^T x_i + b_m \geq 1 - \xi_i^m, \ i \in I : y_i = m \\
& w_m^T x_i + b_m \leq -1 + \xi_i^m, \ i \in I : y_i \neq m \\
& w_m \in \mathbb{R}^n, \ b_m \in \mathbb{R}, \ \xi^m \in \mathbb{R}_+^l.
\end{aligned} \tag{2.9}
$$

Here, slack variables $\xi^m \in \mathbb{R}_+^l$ denote margin violation with respect to the hyperplane constructed by classifier m. Notice that this problem only separates class m from the other classes, therefore, to solve multi-class problem, this problem should be evaluated for remaining $k-1$ classes. For sample i, the binary classifier that gives the maximum output value, i.e., $\arg\max_{m \in M} w_m^T x_i + b_m$ is assigned as its class label.

2.2.2 One Against One Method

One against one method creates $\frac{k(k-1)}{2}$ binary classification problems. Different from *one against all* method, this approach designs soft-margin classifier that discriminates class $m \in M$ from class $j \in M \setminus \{m\}$[30]. For this purpose, the proposed method reconstructs the dataset by relabeling the samples of class m as $+1$ and class j as -1. Then the soft-margin classifier mj, which produces the separating hyperplane between class m and class j, is trained on this reconstructed dataset. The optimal hyperplane parameters resulting from this classifier are denoted as $w_{mj} \in \mathbb{R}^n$ and $b_{mj} \in \mathbb{R}$ which are normal vector and intercept of the corresponding hyperplane, respectively. Thus, the *one against one* SVM solves the following soft-margin problem with error penalization parameter $C > 0$ to separate class m from class j:

$$
\begin{aligned}
\min\ & \frac{1}{2} w_{mj}^T w_{mj} + C \sum_{i \in I : y_i \in \{m,j\}} \xi_i^{mj} \\
\text{s.t.}\ & w_{mj}^T x_i + b_{mj} \geq 1 - \xi_i^{mj},\ i \in I : y_i = m \\
& w_{mj}^T x_i + b_{mj} \leq -1 + \xi_i^{mj},\ i \in I : y_i = j \\
& \xi_i^{mj} \geq 0,\ i \in I : y_i \in \{m, j\} \\
& w_{mj} \in \mathbb{R}^n,\ b_{mj} \in \mathbb{R}.
\end{aligned}
\tag{2.10}
$$

Here, slack variables $\xi_i^{mj} \geq 0,\ i \in I : y_i \in \{m, j\}$ denote the margin violation with respect to the hyperplane that separates class m and class j. Note that this problem separates only the classes m and j which implies that it requires less time to train a classifier compared to *one against all* method for the reason that only a smaller portion (subset) of dataset is considered. However, to solve multi-class problem, (2.10) should be solved for all possible pairs, i.e, $\frac{k(k-1)}{2}$ classifiers should be trained. Simple voting is used to determine the class label of data points. If $sgn(w_{mj}^T x_i + b_{mj})$ is $+1$, then x_i belongs to class m and the vote for class m increases by one, otherwise the vote is added to class j. Here $sgn(a) \in \{-1, 0, +1\}$. $sgn(a) = -1$ if $a < 0$, $sgn(a) = +1$ if $a > 0$ and $sgn(a) = 0$ if $a = 0$. After votes from all possible classifiers are achieved, x_i is labeled as class collecting the highest vote [34].

2.2.3 Weston and Watkins Multi-Class SVM

Weston and Watkins [32] present an *all-together* scheme which considers given multi-class dataset in one optimization problem. This model can be seen as a variant of soft-margin SVM formulation for multi-class case and the margin violation committed by each sample consists of $k-1$ components. For sample i, this model considers the violation obtained from all possible pairwise comparisons, i.e, $\xi_i^m \geq 0,\ m \in M \setminus \{y_i\}$ and calculates the total margin violation committed by this sample as $\sum_{m \in M \setminus \{y_i\}} \xi_i^m$ [34]. Given the training set, the Weston and Watkins multi-class SVM (WW-MSVM) formulation with error penalization parameter $C > 0$ is given as follows:

$$\min \ \frac{1}{2}\sum_{m\in M} w_m^T w_m + C\sum_{i\in I}\sum_{m\in M\setminus\{y_i\}} \xi_i^m \tag{2.11a}$$

$$\text{s.t.} \ \ w_{y_i}^T x_i + b_{y_i} \geq w_m^T x_i + b_m + 2 - \xi_i^m, \ i \in I, \ m \in M \setminus \{y_i\} \tag{2.11b}$$

$$\xi_i^m \geq 0, \ i \in I, \ m \in M \setminus \{y_i\} \tag{2.11c}$$

$$w_m \in \mathbb{R}^n, \ b_m \in \mathbb{R}, \ m \in M. \tag{2.11d}$$

In the objective (2.11a), the quadratic term maximizes the margin width with respect to each separating hyperplane while the second term minimizes the margin violation and training error. The constraint (2.11b) constructs the separating hyperplanes by allowing margin violation. Consider samples $x_1,\ x_2$ with classes $m_1,\ m_2 \in M$ and $m_1 \neq m_2$, respectively. Then the constraint (2.11b) for these samples is written as $(w_{m_1} - w_{m_2})^T x_1 + b_{m_1} - b_{m_2} \geq 2 - \xi_1^{m_2}$ and $(w_{m_2} - w_{m_1})^T x_2 + b_{m_2} - b_{m_1} \geq 2 - \xi_2^{m_1}$. If the second inequality is multiplied by -1, we obtain $(w_{m_1} - w_{m_2})^T x_2 + b_{m_1} - b_{m_2} \leq -2 + \xi_2^{m_1}$. Notice that the hyperplane $(w_{m_1} - w_{m_2})^T x_2 + b_{m_1} - b_{m_2} = 0$ separates the samples $x_1,\ x_2$ if $\xi_2^{m_1} < 2$ and $\xi_1^{m_2} < 2$.Therefore, any $\xi_i^m > 2$ indicates misclassification. The last constraint (2.11c) bounds violation term below by 0. For sample i, the binary classifier that gives the maximum output value, i.e., $\arg\max_{m\in M} w_m^T x_i + b_m$ is assigned as its class label [32].

2.2.4 Crammer and Singer Multi-Class SVM

Crammer and Singer [33] present another *all-together* approach similar to WW-SVM, therefore, can be seen as another variant of soft-margin SVM formulation for multi-class case. Similar to WW-SVM, the margin violation committed by each sample consists of different components. However, in this approach, slack variable for sample i, $\xi_i \geq 0$, is interested in only the maximum margin violation committed by this sample [34]. Crammer and Singer Multi-Class SVM (CS-MSVM) solves the following optimization problem with error penalization

parameter $C > 0$:

$$\min \ \frac{1}{2} \sum_{m \in M} w_m^T w_m + C \sum_{i \in I} \xi_i \tag{2.12a}$$

$$\text{s.t.} \ \ w_{y_i}^T x_i + b_{y_i} - w_m^T x_i - b_m \geq 1 - \xi_i, \ i \in I, \ m \in M \backslash \{y_i\} \tag{2.12b}$$

$$w_m \in \mathbb{R}^n, \ b_m \in \mathbb{R}, \ m \in M \tag{2.12c}$$

$$\xi \in \mathbb{R}_+^l. \tag{2.12d}$$

As in WW-MSVM, the objective (2.12a) maximizes the margin width with respect to each separating hyperplane while the second term minimizes the margin violation and training error. By the same discussion in Section 2.2.3, the constraint (2.12b) constructs the separating hyperplanes and the constraint (2.12d) bounds the violation term ξ below by zero. For sample i, the binary classifier that gives the maximum output value, i.e., $\arg\max_{m \in M} w_m^T x_i + b_m$ is assigned as its class label [33].

2.3 Financial Risk Measures: Value-at-Risk and Conditional Value-at-Risk

VaR is a widely used financial risk measure in market risk. For a given confidence level $\alpha \in (0, 1)$, VaR_α denotes the α quantile of the loss distribution [35]. Let L be a random variable denoting loss with the cumulative distribution function $F_L(t) = P\{L \leq t\}$. Then the mathematical formulation for $VaR_\alpha(L)$ with confidence level α is

$$\text{VaR}_\alpha(L) = \min\{t | F_L(t) \geq \alpha\}. \tag{2.13}$$

Main drawback of VaR is that it ignores the deviation of the losses that are exceeding the quantile. Consequently, VaR is indifferent to the situations with overwhelming losses which can be seen as an optimistic behavior rather than conservative. Another drawback reported in the literature is related to the undesirable mathematical characteristics of this measure, see [36], [37], [38]. VaR is shown to be lack of sub-additivity in addition to being computationally intractable unless the loss distribution is normal [37], [38].

CVaR is another popular financial risk measure which was first introduced by Rockafellar and Uryasev [17]. For continuous random variables, CVaR_α stands for the conditional expectation of the losses that are greater than the threshold indicated by VaR_α. Particularly, unlike VaR_α, CVaR_α takes the distribution of the losses exceeding α quantile into consideration. When random variable L is continuous, mathematical representation of $\text{CVaR}_\alpha(L)$ is given as:

$$\text{CVaR}_\alpha(L) = E[L|L \geq VaR_\alpha(L)]. \tag{2.14}$$

Rockafellar and Uryasev [39] report mathematical representation of CVaR for general distributions as:

$$\text{CVaR}_\alpha(L) = \min_{\eta \in \mathbb{R}} \left\{ \eta + \frac{1}{1-\alpha} E[L-\eta]_+ \right\}, \tag{2.15}$$

where $[t]_+ := \max\{0, t\}$. As an alternative risk measure, it is shown that CVaR has more desirable mathematical properties than VaR such that it can be linearized for discrete distributions [38], [40]. Additionally, CVaR is proven to be convex, positive homogeneous, translation invariant and monotonic by Pflug [41] resulting that CVaR is computationally superior to VaR in applications [17], [39]. Definition of VaR and CVaR can be better understood in Figure 2.3.

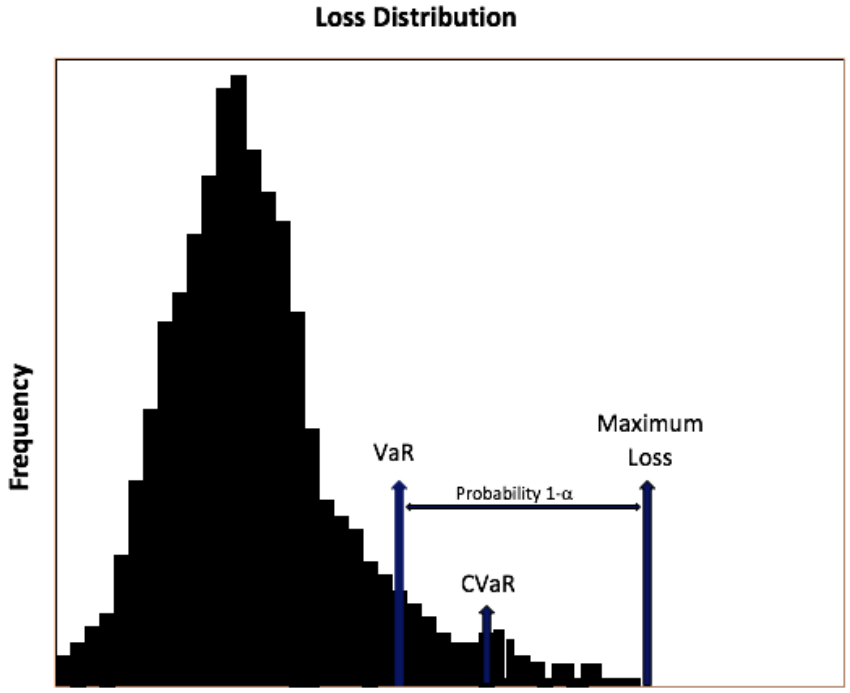


Figure 2.3: VaR and CVaR representation

In Figure 2.3, VaR denotes the α quantile of the distribution, in other words, probability of observing a loss greater than VaR is no larger than $1 - \alpha$. CVaR stands for the conditional expected value of the losses exceeding VaR.

Chapter 3

Risk-Averse SVM

In this chapter, relation between financial risk measures (VaR and CVaR) and SVM is introduced. In section 3.1.1, brief explanation of the connection between CVaR minimization and binary SVM formulations is given. Section 3.1.2 explores the reformulation of hard-margin SVM using VaR constraints and a new variant of SVM with relaxation of these constraints. Section 3.2 focuses on extending risk-averse binary SVM approach to multi-class case with WW-MSVM and CS-MSVM formulations. In the remaining parts of this chapter, implementation of risk measures CVaR and VaR to WW-MSVM and CS-MSVM is provided.

3.1 Risk-Averse Binary SVM

The relation between SVM and risk minimization is first presented by Gotoh and Takeda [42]. They propose an SVM model which minimizes misclassification risk measured by CVaR and show that the proposed model is equivalent to ν-SVM formulation (2.8). Later Takeda and Sugiyama [12] reformulate ν-SVM as CVaR minimization, namely $E\nu$-SVM, by fixing euclidean norm of hyperplane parameter $w \in \mathbb{R}^n$, i.e, $w^T w = 1$ and provide theoretical background for good

generalization performance of the proposed method. Lastly, VaR SVM is introduced as robust SVM classifier and shown to perform better than ν-SVM when existence of outliers is an issue [13].

3.1.1 ν-SVM and CVaR Minimization

Let $\Omega = \{\omega_1, \ldots, \omega_l\}$ be a finite sample space where $\mathbb{P}(\omega_i) = \frac{1}{l},\ i \in I$. [1] Also, let $X : \Omega \to \mathbb{R}^n$ and $Y : \Omega \to \{-1, +1\}$ be discrete random variables such that $X(\omega_i) = x_i,\ Y(\omega_i) = y_i$ for $i \in I$. Let's define a loss function as follows:

$$L^B_{\omega_i}(w, b) = -y_i(w^T x_i + b),\ i \in I, \tag{3.1}$$

where $\mathbb{P}(L^B(w, b) = L^B_{\omega_i}(w, b)) = \frac{1}{l}$. Recall the CVaR formulation (2.15) and let:

$$\begin{aligned} \eta =& -\rho, \\ \alpha =& 1 - \nu, \\ L =& L^B(w, b). \end{aligned} \tag{3.2}$$

Then, we get:

$$\mathrm{CVaR}_{1-\nu}(L^B(w, b)) = \min_{\rho \in \mathbb{R}} \{-\rho + \frac{1}{\nu l} \sum_{i \in I} [\rho - y_i(w^T x_i + b)]_+\}. \tag{3.3}$$

Recall the ν-SVM formulation given in (2.8). If it is reformulated as an unconstrained optimization problem, the resulting model becomes:

$$\min_{\substack{w \in \mathbb{R}^n, \\ b \in \mathbb{R},\ \rho \in \mathbb{R}}} \quad \frac{1}{2} w^T w + \nu(-\rho + \frac{1}{\nu l} \sum_{i=1}^{l} [\rho - y_i(w^T x_i + b)]_+). \tag{3.4}$$

Note that the second term is equal to $\nu \mathrm{CVaR}_{1-\nu}(L^B(w, b))$. In this context, CVaR measures the misclassification risk where the loss incurred by sample i is defined as $-y_i(w^T x_i + b)$, in other words, its distance to the separating hyperplane. When data point i is correctly classified, the expression $-y_i(w^T x_i - b)$ takes negative value. If sample i belongs to class $+1$ and is correctly classified, it should be on

[1] The approach can be extended to an arbitrary discrete probability distribution, i.e, $\mathbb{P}(\omega_i) = p_i,\ i \in I$ where $p_i \geq 0,\ i \in I,\ \sum_{i \in I} p_i = 1$ [13].

the positive side of the separating hyperplane, i.e, $(w^T x_i + b) \geq 0$, similarly, if sample i belongs to class -1 and is correctly classified, it should be on the negative side of the separating hyperplane, i.e, $(w^T x_i + b) \leq 0$ indicating $y_i(w^T x_i + b) \geq 0$ for correctly classified samples. Therefore, loss, $-y_i(w^T x_i + b)$, is negative if sample i is on the right side of the separating hyperplane. Higher values of loss are obtained if these samples are close to the hyperplane. Proceeding from this observation, loss is positive for misclassified data points. Considering the loss distribution obtained from the data, misclassified samples incur higher loss values together with samples that are close to the separating hyperplane. Hence, these samples contribute more to the upper tail of the distribution of loss and $\text{CVaR}_{1-\nu}(L^B(w,b))$ aims to minimize the conditional expectation of the losses incurred by these samples.

3.1.2 Hard-Margin SVM and VaR Representation

As hard-margin SVM model (2.3) requires all constraints to hold with no violation, the formulation can be rewritten as chance constrained optimization using the loss function (3.1). Then the hard-margin SVM model becomes:

$$\begin{aligned} \min \ & \frac{1}{2} w^T w && \text{(3.5a)} \\ \text{s.t.} \ & \mathbb{P}(L^B(w,b) \leq -1) = 1 && \text{(3.5b)} \\ & w \in \mathbb{R}^n, \ b \in \mathbb{R}. && \text{(3.5c)} \end{aligned}$$

Constraint (3.5b) of this problem means the random loss $L^B(w,b)$ should be less than or equal to -1 for all scenarios in sample space, i.e, for all $\omega_i \in \Omega$. This chance constraint can be relaxed in a way that the random loss can violate the given threshold of -1 for some scenarios in sample space where violations are restricted with a probability level $\alpha \in (0,1]$. Then the problem becomes:

$$\begin{aligned} \min \ & \frac{1}{2} w^T w \\ \text{s.t.} \ & \mathbb{P}(L^B(w,b) \leq -1) \geq \alpha \\ & w \in \mathbb{R}^n, \ b \in \mathbb{R}. \end{aligned} \tag{3.6}$$

Recall the definition of VaR given in (2.13). The VaR of $L^B(w,b)$ at level α is written as $\text{VaR}_\alpha(L^B(w,b)) = \min\{t|\mathbb{P}(L^B(w,b) \leq t) \geq \alpha\}$. It is clear that $\text{VaR}_\alpha(L^B(w,b)) \leq -1$ is equivalent to $P(L^B(w,b) \leq -1) \geq \alpha$ by definition. Therefore, formulation (3.6) is equivalent to following VaR constrained SVM problem, namely, VaR SVM [13]:

$$\begin{aligned} \min \ & \frac{1}{2}w^T w && \text{(3.7a)} \\ \text{s.t.} \ & \text{VaR}_\alpha(L^B(w,b)) \leq -1 && \text{(3.7b)} \\ & w \in \mathbb{R}^n, \ b \in \mathbb{R}. && \text{(3.7c)} \end{aligned}$$

The constraint (3.7b) is recognized as chance constraint $\mathbb{P}(-L^B(w,b) \geq 1) \geq \alpha$. For given random variable $L^B(w,b)$ with scenario set Ω, $\mathbb{P}(-L^B(w,b) \geq 1) \geq \alpha$ can be linearized as follows [19]:

$$\begin{aligned} & -L_{\omega_i}(w,b) + \delta_i B \geq 1, \ i \in I \\ & \sum_{i=1}^{l} p_i \delta_i \leq 1 - \alpha \\ & w \in \mathbb{R}^n, \ b \in \mathbb{R}, \ \delta_i \in \{0,1\}, \ i \in I, \end{aligned} \tag{3.8}$$

where $i \in I$ corresponds to index of scenario ω_i and $B \in \mathbb{R}_+$ is a sufficiently large number such that when $\delta_i = 1$, the corresponding constraint is not active. Hence final VaR SVM formulation takes the form:

$$\begin{aligned} \min \ & \frac{1}{2}w^T w \\ \text{s.t.} \ & y_i(w^T x_i + b) + \delta_i B \geq 1, \ i \in I \\ & \sum_{i=1}^{l} p_i \delta_i \leq 1 - \alpha \\ & w \in \mathbb{R}^n, \ b \in \mathbb{R}, \ \delta_i \in \{0,1\}, \ i \in I. \end{aligned} \tag{3.9}$$

3.2 Risk-Averse MSVM

As it is stated before, good generalization performance of ν-SVM is theoretically justified [12]. Also, note that ν-parameterization provides more control on construction of the separating hyperplanes for binary case [28]. For this purpose,

we extend ν-parameterization to multi-class case. However, as CVaR considers the extreme losses exceeding threshold indicated by VaR, which can be a result of a rare event or outliers, it may not give stable results when existence of outliers is an issue. Tsyurmasto et al. [13] show that VaR SVM is more stable to outliers as it ignores the extreme losses at a given confidence level. Considering the performance of binary SVM with risk measures in applications, we aim to extend the risk-averse binary SVM approach to multi-class case. In this section, implementation of CVaR and VaR to WW-MSVM and CS-MSVM models is provided.

3.2.1 CVaR WW-MSVM

Recall that WW-MSVM is a multi-class version of soft-margin SVM as the model allows margin violation and training error. Considering the difference between soft-margin SVM and ν-SVM, CVaR WW-MSVM can be introduced by replacing 2 with decision variable $\rho \in \mathbb{R}$ in (2.11b) and adding term $-\nu\rho$ to the objective function:

$$\begin{aligned} \min \ & \frac{1}{2}\sum_{m\in M} w_m^T w_m - \nu\rho + \frac{1}{l(k-1)}\sum_{i\in I}\sum_{m\in M\setminus\{y_i\}} \xi_i^m \\ \text{s.t.} \ & \xi_i^m \geq \rho - ((w_{y_i} - w_m)^T x_i + b_{y_i} - b_m), i \in I,\ m \in M \setminus \{y_i\} \\ & \xi_i^m \geq 0,\ i \in I,\ m \in M \setminus \{y_i\} \\ & w_m \in \mathbb{R}^n,\ b_m \in \mathbb{R},\ m \in M \\ & \rho \in \mathbb{R}. \end{aligned} \tag{3.10}$$

Let $\bar{\Omega} = \{\omega_i^m | i \in I,\ m \in M \setminus \{y_i\}\}$ be a finite sample space where $\mathbb{P}(\omega_i^m) = \frac{1}{l(k-1)},\ i \in I,\ m \in M \setminus \{y_i\}$. [2] Let $X : \bar{\Omega} \to \mathbb{R}^n$, $Y : \bar{\Omega} \to M$ and $\mathcal{M} : \bar{\Omega} \to M$ be discrete random variables such that $X(\omega_i^m) = x_i,\ Y(\omega_i^m) = y_i,\ \mathcal{M}(\omega_i^m) = m$ for $i \in I,\ m \in M \setminus \{y_i\}$. The loss function is defined as follows:

$$L_{\omega_i^m}^{WW}(\boldsymbol{w}, \boldsymbol{b}) = -((w_{y_i} - w_m)^T x_i + b_{y_i} - b_m),\ \omega_i^m \in \bar{\Omega}. \tag{3.11}$$

[2] The approach can be extended to an arbitrary discrete probability distribution, i.e, $\mathbb{P}(\omega_i^m) = p_i^m\ i \in I,\ m \in M \setminus \{y_i\}$ where $p_i^m \geq 0,\ i \in I,\ m \in M \setminus \{y_i\}\ \sum_{i\in I}\sum_{m\in M\setminus\{y_i\}} p_i^m = 1$.

Here $\boldsymbol{w} = (w_1, \ldots, w_k)$ and $\boldsymbol{b} = (b_1, \ldots, b_k)$ with $w_m \in \mathbb{R}^n,\ b_m \in \mathbb{R},\ m \in M$ and $\mathbb{P}(L^{WW}(\boldsymbol{w}, \boldsymbol{b}) = L^{WW}_{\omega_i^m}(\boldsymbol{w}, \boldsymbol{b})) = \frac{1}{l(k-1)}$. Recall the CVaR formulation (2.15) and let:

$$\begin{aligned} \eta =& -\rho, \\ \alpha =& 1-\nu, \\ L =& L^{WW}(\boldsymbol{w}, \boldsymbol{b}). \end{aligned} \tag{3.12}$$

Then, we get:

$$\begin{aligned} \mathrm{CVaR}_{1-\nu}(L^{WW}(\boldsymbol{w}, \boldsymbol{b})) = \min_{\rho \in \mathbb{R}} \{ -\rho + \\ \frac{1}{\nu l(k-1)} \sum_{i \in I} \sum_{m \in M \setminus \{y_i\}} [-((w_{y_i} - w_m)^T x_i + b_{y_i} - b_m) + \rho]_+ \}. \end{aligned} \tag{3.13}$$

If we rewrite (3.10) as an unconstrained optimization problem, we obtain:

$$\begin{aligned} \min_{\substack{w_m \in \mathbb{R}^n, m \in M, \\ b_m \in \mathbb{R}, m \in M \\ \rho \in \mathbb{R}}} \frac{1}{2} \sum_{m \in M} w_m^T w_m + \\ \nu(-\rho + \frac{1}{\nu l(k-1)} \sum_{i \in I} \sum_{m \in M \setminus \{y_i\}} [-((w_{y_i} - w_m)^T x_i + b_{y_i} - b_m) + \rho]_+). \end{aligned} \tag{3.14}$$

Using equation (3.13), the expression after quadratic term equals to $\nu \mathrm{CVaR}_{1-\nu}(L^{WW}(\boldsymbol{w}, \boldsymbol{b}))$. In this context, CVaR measures the misclassification risk where loss incurred by sample $i \in I$ with respect to component $m \in M \setminus \{y_i\}$ is defined as $-((w_{y_i} - w_m)^T x_i + b_{y_i} - b_m)$, in other words, its distance to the hyperplane that separates class y_i and m. The sign of loss function indicates misclassification. Recall that class label of sample i is determined by $\arg\max_{m \in M} \{w_m^T x_i + b_m\}$, particularly, if sample i is correctly classified, y_i should maximize the argument, implying $w_{y_i}^T x_i + b_{y_i} \geq w_m^T x_i + b_m,\ m \in M \setminus \{y_i\}$. Therefore, the inequality $(w_{y_i} - w_m)^T x_i + b_{y_i} - b_m \geq 0$ should hold for $m \in M \setminus \{y_i\}$. Proceeding from here, the expression $-((w_{y_i} - w_m)^T x_i + b_{y_i} - b_m)$ takes negative value if sample i is correctly classified otherwise it is positive. Similar to the discussion given in Section 3.1.1, samples that are misclassified or violate margin contribute more to the upper tail of the loss distribution. Hence, the aim of $\mathrm{CVaR}_{1-\nu}(L^{WW}(\boldsymbol{w}, \boldsymbol{b}))$ is to minimize the conditional expectation of the losses incurred by these data points.

3.2.2 VaR WW-MSVM

Similar to approach proposed in binary VaR SVM, in this section, we aim to introduce VaR to WW-MSVM. For this purpose, we consider the hard-margin version of WW-MSVM which omits slack variables:

$$\begin{aligned} \min\ & \frac{1}{2}\sum_{m\in M} w_m^T w_m \\ \text{s.t.}\ & (w_{y_i}-w_m)^T x + (b_{y_i}-b_m) \geq 2,\ i\in I,\ m\in M\setminus\{y_i\} \\ & w_m\in\mathbb{R}^n,\ b_m\in\mathbb{R},\ m\in M. \end{aligned} \tag{3.15}$$

Recall the random loss defined in (3.11). Then VaR WW-MSVM formulation is presented as:

$$\begin{aligned} \min\ & \frac{1}{2}\sum_{m\in M} w_m^T w_m \\ \text{s.t.}\ & \mathrm{VaR}_\alpha(L^{WW}(\boldsymbol{w},\boldsymbol{b})) \leq -2 \\ & w_m\in\mathbb{R}^n,\ b_m\in\mathbb{R},\ m\in M. \end{aligned} \tag{3.16}$$

As previously shown this formulation is equivalent to chance constrained optimization problem in the form:

$$\begin{aligned} \min\ & \frac{1}{2}\sum_{m\in M} w_m^T w_m \\ \text{s.t.}\ & \mathbb{P}(-L^{WW}(\boldsymbol{w},\boldsymbol{b}) \geq 2) \geq \alpha \\ & w_m\in\mathbb{R}^n,\ b_m\in\mathbb{R},\ m\in M. \end{aligned} \tag{3.17}$$

Similar to binary VaR SVM, (3.17) is subject to a chance constraint. Following the same linearization procedure given in Section 3.1.2, mixed integer quadratic formulation of VaR WW-MSVM is obtained as follows:

$$\begin{aligned} \min\ & \sum_{m\in M} w_m^T w_m \\ \text{s.t.}\ & (w_{y_i}-w_m)^T x + b_{y_i} - b_m + \delta_i^m B \geq 2,\ i\in I,\ m\in M\setminus\{y_i\} \\ & \sum_{i\in I}\sum_{m\in M\setminus\{y_i\}} p_i^m \delta_i^m \leq 1-\alpha \\ & w_m\in\mathbb{R}^n,\ b_m\in\mathbb{R},\ \delta_i^m\in\{0,1\},\ i\in I,\ m\in M\setminus\{y_i\}. \end{aligned} \tag{3.18}$$

3.2.3 CVaR CS-MSVM

Notice that CS-MSVM (2.12) is another variant of multi-class soft-margin SVM. Based on the transition from soft-margin SVM to ν-SVM, CVaR CS-MSVM can be introduced by replacing 1 with decision variable $\rho \in \mathbb{R}$ in (2.12b) and adding term $-\nu\rho$ to the objective function:

$$\begin{aligned} \min \ & \frac{1}{2}\sum_{m\in M} w_m^T w_m - \nu\rho + \frac{1}{l}\sum_{i\in I}\xi_i \\ \text{s.t.} \ & \xi_i \geq \rho - ((w_{y_i} - w_m)^T x_i + b_{y_i} - b_m), i \in I, \ m \in M \setminus \{y_i\} \\ & w_m \in \mathbb{R}^n, \ b_m \in \mathbb{R}, \ m \in M \\ & \xi \in \mathbb{R}^l_+, \ \rho \in \mathbb{R}. \end{aligned} \tag{3.19}$$

Given sample space Ω in Section 3.1.1, let $X : \Omega \to \mathbb{R}^n$ and $Y : \Omega \to M$ be discrete random variables such that $X(\omega_i) = x_i$, $Y(\omega_i) = y_i$ for $i \in I$. Then, the loss function for CVaR CS-MSVM is defined as follows:

$$L^{CS}_{\omega_i}(\boldsymbol{w}, \boldsymbol{b}) = \max_{m\in M\setminus\{y_i\}}\{-((w_{y_i} - w_m)^T x_i + b_{y_i} - b_m)\}, \ \omega_i \in \Omega. \tag{3.20}$$

Here, $\mathbb{P}(L^{CS}(\boldsymbol{w}, \boldsymbol{b}) = L^{CS}_{\omega_i}(\boldsymbol{w}, \boldsymbol{b})) = \frac{1}{l}$. Recall the CVaR formulation (2.15) and let:

$$\begin{aligned} \eta =& -\rho, \\ \alpha =& 1 - \nu, \\ L =& L^{CS}(\boldsymbol{w}, \boldsymbol{b}). \end{aligned} \tag{3.21}$$

Then, we get:

$$\begin{aligned} \mathrm{CVaR}_{1-\nu}(L^{CS}(\boldsymbol{w}, \boldsymbol{b})) = \min_{\rho\in\mathbb{R}}\{-\rho + \\ \frac{1}{\nu l}\sum_{i\in I}[\rho + \max_{m\in M\setminus\{y_i\}}\{-((w_{y_i} - w_m)^T x_i + b_{y_i} - b_m)\}]_+\}. \end{aligned} \tag{3.22}$$

When (3.19) is rearranged as an unconstrained optimization problem, we obtain:

$$\begin{aligned} \min_{\substack{w_m\in\mathbb{R}^n, m\in M \\ b_m\in\mathbb{R}, m\in M \\ \rho\in\mathbb{R}}} \ & \frac{1}{2}\sum_{m\in M} w_m^T w_m + \\ & \nu(-\rho + \frac{1}{\nu l}\sum_{i\in I}[\max_{m\in M\setminus\{y_i\}}\{\rho - ((w_{y_i} - w_m)^T x_i + b_{y_i} - b_m)\}]_+). \end{aligned} \tag{3.23}$$

Note that, as ρ does not depend on m, it can be written outside of the maximum term:

$$\begin{aligned}\min_{\substack{w_m\in\mathbb{R}^n, m\in M\\ b_m\in\mathbb{R}, m\in M\\ \rho\in\mathbb{R}}} \frac{1}{2}\sum_{m\in M} w_m^T w_m + \\ \nu(-\rho + \frac{1}{\nu l}\sum_{i\in I}[\rho + \max_{m\in M\setminus\{y_i\}}\{-((w_{y_i}-w_m)^T x_i + b_{y_i} - b_m)\}]_+).\end{aligned} \tag{3.24}$$

The expression after quadratic term equals to $\nu\text{CVaR}_{1-\nu}(L^{CS}(\boldsymbol{w}, \boldsymbol{b}))$ by equation (3.22). In this context, CVaR measures the misclassification risk where loss incurred by sample $i \in I$ is defined as $\max_{m\in M\setminus\{y_i\}}\{-((w_{y_i}-w_m)^T x_i + b_{y_i} - b_m)\}$ which considers each margin violations committed by sample i to the hyperplane that separates class y_i and $m \in M \setminus \{y_i\}$ and takes maximum margin violation as loss. By the same discussion given in Section 3.2.1, loss is negative for correctly classified data points, while it takes positive values for misclassified observations. Note that the aim of $\text{CVaR}_{1-\nu}(L^{CS}(\boldsymbol{w}, \boldsymbol{b}))$ is to minimize the conditional expectation of the losses falling into the upper tail.

3.2.4 VaR CS-MSVM

In this section, VaR CS-MSVM is introduced based on the methodology used in Section 3.2.2. Recall the loss function (3.22). Then, the proposed VaR CS-MSVM model is given as:

$$\begin{aligned}\min\ & \frac{1}{2}\sum_{m\in M} w_m^T w_m \\ \text{s.t.}\ & VaR_\alpha(L^{CS}(\boldsymbol{w}, \boldsymbol{b})) \le -1 \\ & w_m \in \mathbb{R}^n,\ b_m \in \mathbb{R},\ m \in M.\end{aligned} \tag{3.25}$$

As previously shown this formulation is equivalent to chance constrained optimization problem in the form:

$$\begin{aligned}\min\ & \frac{1}{2}\sum_{m\in M} w_m^T w_m \\ \text{s.t.}\ & \mathbb{P}(-L^{CS}(\boldsymbol{w}, \boldsymbol{b}) \ge 1) \ge \alpha \\ & w_m \in \mathbb{R}^n,\ b_m \in \mathbb{R},\ m \in M.\end{aligned} \tag{3.26}$$

Here, if the expression inside the chance constraint is written explicitly, we get:

$$-\max_{m\in M\setminus\{y_i\}}\{-((w_{y_i}-w_m)^Tx+b_{y_i}-b_m)\}\geq 1,\ i\in I \tag{3.27}$$

which is equivalent to:

$$(w_{y_i}-w_m)^Tx+b_{y_i}-b_m m\geq 1,\ m\in M\setminus\{y_i\},\ i\in I. \tag{3.28}$$

Here I is the index set of scenarios and in the chance constraint, we have more than one inequality which is called joint chance constraint. Therefore, unlike VaR WW-MSVM, (3.26) is subject to joint chance constraint as random variable $L^{CS}_{\omega_i}$ takes the maximum of $k-1$ inequality for each scenario $\omega_i\in\Omega$. For given random variable $L^{CS}(\boldsymbol{w},\boldsymbol{b})$ with index set $i\in I$ for scenario ω_i, corresponding $\mathbb{P}(-L^{CS}(\boldsymbol{w},\boldsymbol{b})\geq 1)\geq\alpha$ is a joint chance constraint and can be linearized as follows [19]:

$$\begin{aligned}
&-L^{CS}_{\omega_i}(\boldsymbol{w},\boldsymbol{b})+\delta_i^m B\geq 1,\ i\in I,\ m\in M\setminus\{y_i\}\\
&\Delta_i\geq\delta_i^m,\ i\in I,\ m\in M\setminus\{y_i\}\\
&\sum_{i\in I}p_i\Delta_i\leq 1-\alpha,\ i\in I\\
&\Delta_i,\delta_i^m\in\{0,1\},\ i\in I,\ m\in M\setminus\{y_i\}\\
&w_m\in\mathbb{R}^n,\ b_m\in\mathbb{R},\ m\in M.
\end{aligned} \tag{3.29}$$

Hence, final mixed integer formulation of VaR CS-MSVM with scenario index set I is in the form:

$$\begin{aligned}
\min\ &\frac{1}{2}\sum_{m\in M}w_m^Tw_m\\
\text{s.t.}\ &(w_{y_i}-w_m)^Tx+(b_{y_i}-b_m)+\delta_i^m B\geq 1,\ i\in I,\ m\in M\setminus\{y_i\}\\
&\Delta_i\geq\delta_i^m,\ i\in I,\ m\in M\setminus\{y_i\}\\
&\sum_{i\in I}p_i\Delta_i\leq 1-\alpha\\
&w_m\in\mathbb{R}^n,\ b_m\in\mathbb{R},\ \Delta_i\in\{0,1\},\ \delta_i^m\in\{0,1\},\ i\in I,\ m\in M\setminus\{y_i\}.
\end{aligned} \tag{3.30}$$

Chapter 4

Solution Methodoly for VaR MSVM

Note that CVaR WW-MSVM (3.10) and CVaR CS-MSVM (3.19) are convex programming problems and therefore computationally tractable. Unlike CVaR, VaR is difficult to optimize unless the loss is normally distributed. Due to non-convexity of VaR, VaR constrained problems are non-convex optimization problems which is computationally intractable. Therefore, Problems (3.16) and (3.25) are difficult to solve. In Chapter 3, it is shown that VaR constraints can be represented as chance constraints. To solve chance constrained optimization problems, several methods are proposed in the literature. For problems with finite number of scenarios, one is to reformulate the chance constrained problems as mixed integer programming problems by introducing big-M, which can be seen in Problems (3.18) and (3.30). However, large values of big-M give weak continuous relaxations leading poor computational performance in branch and bound methods. McCormick linearization is another method proposed to deal with nonlinear constraints. It builds a big-M formulation by computing different big-M values for each scenario which may result in weak continuous relaxation as well [43]. Similar to this approach, Qiu et al. [44] propose big-M strengthening which requires solving an LP relaxation for each scenario iteratively until the coefficients are converged to a valid bound within a given threshold so that the tightest

coefficients are obtained. However, when scenario set is large, this method is computationally inefficient. To speed up this method, Song et al. [45] present a procedure to obtain an upper bound for chance constrained binary packing problems. Yet, applying this procedure to our problem results loose upper bounds. Another method is using augmented Lagrangian decomposition for mixed integer formulation of chance constrained problems [46]. However, this method does not guarantee to find the global optimum. In this study, to solve VaR MSVM models, we propose a strong big-M formulation using the valid inequalities discussed in [20] and provide a comparison of our formulation and branch and cut decomposition algorithm proposed by Luedtke [20] which avoids the use of big-M.

4.1 A Branch and Cut Decomposition Algorithm for Solving Chance-Constrained Mathematical Programs with Finite Support

In this section, we follow the notation in [20]. Luedtke [20] introduces a branch and cut decomposition algorithm to solve general chance constrained mathematical programming problems that have discrete distributions. Consider the chance constrained problem of the form:

$$\begin{aligned} \min \ & f(w) \\ \text{s.t.} \ & \mathbb{P}((w,b) \in P(\zeta)) \geq \alpha \\ & w \in \mathcal{D}. \end{aligned} \tag{4.1}$$

Here, $w \in \mathbb{R}^n$ and $b \in \mathbb{R}^r$ are the decision variables, $f : \mathbb{R}^n \to \mathbb{R}$ is objective function to be minimized, ζ is a random vector consisting of scenarios $\omega_i,\ i \in I = \{1, \ldots, l\}$, $P(\omega_i)$ is region characterized by $\omega_i \in \Omega$ and $\mathcal{D} \subseteq \mathbb{R}^n \times \mathbb{R}^r$ is the set of deterministic constraints that do not depend on the scenarios $\omega_i,\ i \in I$. Let $P_i = P(\omega_i),\ i \in I$ be defined as follows:

$$P_i = \{w \in \mathbb{R}^n,\ b \in \mathbb{R}^r \mid T^i w + W^i b \geq c^i\}, \tag{4.2}$$

where $c^i \in \mathbb{R}^d$, $T^i \in \mathbb{R}^{d\times n}$ and $W^i \in \mathbb{R}^{d\times r}$. Let $z_i \in \{0,1\},\ i \in I$ be introduced to Problem (4.1) such that if $z_i = 0$, then $(w,b) \in P_i$. Assuming each scenario is equally likely, (4.1) can be reformulated using implication constraints:

$$\min \ f(w) \tag{4.3a}$$

$$\text{s.t.} \ \ z_i = 0 \implies (w,b) \in P_i,\ i \in I \tag{4.3b}$$

$$\sum_{i\in I} z_i \leq p \tag{4.3c}$$

$$(w,b) \in \mathcal{D} \tag{4.3d}$$

$$z_i \in \{0,1\},\ i \in I, \tag{4.3e}$$

where $p = \lfloor l(1-\alpha)\rfloor$. Then the feasible region of (4.1) is $\mathcal{F} = \{(w,b)|(4.3b) - (4.3e)\}$. The decomposition algorithm is based on three subproblems: single scenario optimization problem, single scenario separation problem and master problem.

4.1.1 Subproblems of Branch and Cut Decomposition Algorithm

First subproblem is presented as single scenario optimization problem for scenario index $i \in I$:

$$h_i(\theta,\mu) = \min\{\theta^T w + \mu^T b | (w,b) \in P_i \cap \bar{\mathcal{D}}\}, \tag{4.4}$$

where $\theta \in \mathbb{R}^n$, $\mu \in \mathbb{R}^r$ and $\bar{\mathcal{D}} \subseteq \mathbb{R}^n \times \mathbb{R}^r$ is a fixed closed set containing $\mathcal{D}$, i.e $\bar{\mathcal{D}} \supseteq \mathcal{D}$, chosen such that $P_i \cap \bar{\mathcal{D}} \neq \emptyset$ in order to preserve feasibility.
Secondly, single scenario separation problem, $Sep(i,\hat{w},\hat{b})$, is introduced to check if found solution, $(\hat{w},\hat{b})$, violates any of the scenarios and obtain parameters $(viol,\theta,\mu,\beta)$ to generate valid inequalities. If $viol$ returns $TRUE$, then the given solution $(\hat{w},\hat{b})$ is infeasible and the parameters (θ,μ,β) are used to cut off this solution.
To solve $Sep(i,\hat{w},\hat{b})$ for given scenario ω_i and solution $(\hat{w},\hat{b})$ consider the problem:

$$\begin{aligned} \min\ & \lambda \\ \text{s.t.}\ & T^i\hat{w} + W^i\hat{b} + \lambda\mathbb{1} \geq c^i \\ & \lambda \in \mathbb{R}_+. \end{aligned} \tag{4.5}$$

Note that, if an optimal solution of (4.5) yields $\lambda^* > 0$, then there exists a $t \in \{1, \ldots, d\}$ such that the constraint $T^i_t\hat{w} + W^i_t\hat{b} < c^i_t$, where subscript T^i_t corresponds to the t-th row of T^i. Then, given solution $(\hat{w}, \hat{b})$ violates scenario ω_i, i.e, $(\hat{w}, \hat{b}) \notin P_i$. When dual variable $\tau \in \mathbb{R}^d_+$ is introduced, we obtain dual problem as:

$$\begin{aligned} v(\hat{w}, \hat{b}) = \max\ & \tau^T(c^i - [T^i\hat{w} + W^i\hat{b}]) \\ \text{s.t.}\ & \tau^T\mathbb{1} \leq 1,\ \tau \in \mathbb{R}^d_+. \end{aligned} \tag{4.6}$$

Let $v^*(\hat{w}, \hat{b})$ be optimal objective value of (4.6). Then, by strong duality, we have $v^*(\hat{w}, \hat{b}) = \lambda^*$. If $\lambda^* = 0$, then an optimal solution of (4.6) is $\tau^* = 0$. Otherwise, to obtain an optimal solution of Problem (4.6), it is sufficient to find $t^* = \arg\max_{t\in\{1,\ldots,d\}} c^i_t - [T^i_t\hat{w} + W^i_t\hat{b}]$. Then t^* entry of τ^* will be 1 and all the other entries will be zero. If the optimal value $v^*(\hat{w}, \hat{b}) > 0$, then $viol = TRUE$ and by setting $\theta = (T^i_{t^*})^T$, $\mu = (W^i_{t^*})^T$ and $\beta = c^i_{t^*}$, we obtain a separating inequality in the form $\theta^T w + \mu^T b + \pi z \geq \beta$, where π denotes the coefficient vector for z which will be discussed in next section. Otherwise, $viol = FALSE$ and $(\theta, \mu, \beta) = 0$.

Last problem is presented as master problem $MP(I_0, I_1, R)$:

$$\begin{aligned} \min\ & f(w) \\ \text{s.t}\ & \sum_{i\in I} z_i \leq p, \\ & (w, b, z) \in R,\ (w, b) \in \mathcal{D} \\ & z_i \in [0, 1],\ i \in I \\ & z_i = 0, i \in I_0,\ z_i = 1, i \in I_1, \end{aligned} \tag{4.7}$$

where R is a polyhedron described by the generated valid inequalities and contains the feasible region of (4.1), denoted by $\mathcal{F}$, and $I_0, I_1 \subseteq I$ are such that $I_0 \cap I_1 = \emptyset$.

4.1.2 Generating Valid Inequalities

To obtain valid inequalities of the form $\theta^T w + \mu^T b + \pi z \geq \beta$, first, single scenario separation problem is solved to obtain separation parameters $(viol, \theta, \mu, \beta)$. Then coefficient vector π is obtained by solving single scenario optimization problem (4.4) for $i \in I$ for given θ and μ. Then values $h_i(\theta, \mu)$ are sorted to obtain a permutation σ of I such that [20]:

$$h_{\sigma_1}(\theta, \mu) \geq h_{\sigma_2}(\theta, \mu) \geq \cdots \geq h_{\sigma_l}(\theta, \mu). \tag{4.8}$$

By the argument in [47], the following inequalities are valid for $\mathcal{F}$ [20]:

$$\theta^T w + \mu^T b + (h_{\sigma_i}(\theta, \mu) - h_{\sigma_{p+1}}(\theta, \mu)) z_{\sigma_i} \geq h_{\sigma_i}(\theta, \mu),\ i = 1, \ldots, p. \tag{4.9}$$

As shown in [48], [49], the following inequality is valid for $\mathcal{F}$ [20]:

$$\theta^T w + \mu^T b + \sum_{i=1}^{q} (h_{\sigma_{t_i}}(\theta, \mu) - h_{\sigma_{t_{i+1}}}(\theta, \mu)) z_{\sigma_{t_i}} \geq h_{t_1}(\theta, \mu), \tag{4.10}$$

where $T = \{t_1, t_2, \ldots, t_q\} \subseteq \{\sigma_1, \ldots, \sigma_p\}$ such that $h_{t_i}(\theta, \mu) \geq h_{t_{i+1}}(\theta, \mu)$ for $i = 1, \ldots, q$ and $h_{t_{q+1}}(\theta, \mu) = h_{\sigma_{p+1}}(\theta, \mu)$.

4.1.3 Algorithms

The proposed algorithm, Algorithm 1, operates similar to branch and bound method and can be briefly explained as follows. In each node, continuous relaxation of master problem (4.7) subject to the set of deterministic constraints, $\mathcal{D}$, and set containing valid inequalities, R, is solved. Particularly, in the root node there is no valid inequality in R leading that the problem is reduced to master problem which contains only deterministic constraints. Therefore, it does not have complete description of the original chance constrained problem. If obtained solution results that $\hat{z}$ is integer feasible after solving the master problem, Algorithm 2 is called to ensure the found solution is in $\mathcal{F}$, i.e, $(\hat{w}, \hat{b}, \hat{z}) \in \mathcal{F}$. If not, Algorithm 2 generates valid inequalities to cut off this solution as it is infeasible. If $\hat{z}$ is fractional, then it is optional to call Algorithm 2 which may result in

improvement in the lower bound. Generated cuts are added to the set R. This process repeats until no cuts are found in the current node, then algorithm proceeds to branching if necessary. If found solution in current node is infeasible or has worse objective value than the best feasible solution to the original problem obtained from previously processed nodes, then it is not processed further, i.e, it is pruned. Finally, when all open nodes are processed, the algorithm terminates.

Algorithm 1: Branch-and-cut decomposition algorithm

$t \leftarrow 0,\ I_0(0) \leftarrow \emptyset,\ I_1(0) \leftarrow \emptyset,\ R \leftarrow \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R}^l,\ Open \leftarrow \{0\},\ U \leftarrow +\infty,\ lb \leftarrow -\infty$;
while $Open \neq \emptyset$ **do**
  Step 1: Choose $o \in Open$ and let $Open \leftarrow Open \setminus \{o\}$;
  Step 2: Process node o;
  while *CUTFOUND = TRUE and* $lb < U$ **do**
    Solve (4.7), $fval \leftarrow MP(I_0(o), I_1(o), R)$;
    if *(4.7) is infeasible or* $fval > U$ **then**
      Prune node l;
      Go to Step 1;
    else
      Let $(\hat{w}, \hat{b}, \hat{z})$ be optimal solution to (4.7);
      if $\hat{z} \in \{0,1\}^l$ **then**
        CUTFOUND = $SepCuts(\hat{w}, \hat{b}, \hat{z}, R)$;
        if *CUTFOUND = FALSE* **then** $U \leftarrow fval$;
      else
        $lb \leftarrow fval$;
        CUTFOUND = FALSE;
      end
    end
  end
  Step 3: Branch if necessary;
  if $lb < U$ **then**
    Choose $i \in I$ such that $\hat{z}_i \in (0,1)$;
    $I_0(t+1) \leftarrow I_0(o) \cup \{i\},\ I_1(t+1) \leftarrow I_1(o)$;
    $I_0(t+2) \leftarrow I_0(o),\ I_1(t+2) \leftarrow I_1(o) \cup \{i\}$;
    $t \leftarrow t+2$;
    $Open \leftarrow Open \cup \{t+1, t+2\}$;
  end
end

Algorithm 2: Cut Separation Routine $SepCuts(\hat{w}, \hat{b}, \hat{z}, R)$

Data: $(\hat{w}, \hat{b}, \hat{z}, R)$

Result: If valid inequalities for F are found that are violated by $(\hat{w}, \hat{b}, \hat{z})$, adds these to the description of R and returns TRUE, else returns FALSE.

```
CUTFOUND = FALSE ;
for i ∈ I such that ẑ_i < 1 do
    Call single scenario separation procedure to obtain (viol, θ, μ, β) ;
    if viol = TRUE then
        Using coefficients θ and μ solve separation problem for inequalities
         in form (4.10). If solution (ŵ, b̂, ẑ) violates any of the inequalities,
         add the set of violated inequalities to the R;
        CUTFOUND ← TRUE ;
    end
end
return CUTFOUND and updated R
```

4.2 Solving VaR WW-MSVM with Branch and Cut Decomposition Algorithm

Recall VaR WW-MSVM problem (3.17). Let $f(\boldsymbol{w}) = \frac{1}{2}\sum_{m\in M} w_m^T w_m$. In order to solve VaR WW-MSVM using the proposed algorithm, the subproblems should be well defined. It is clear that (4.4) is feasible. However, it may not be well defined since variables are unbounded in formulation (3.17). In order for (4.4) to be well defined, we can impose bounds to the continuous variables. Note that each feasible solution $(\hat{\boldsymbol{w}}, \hat{\boldsymbol{b}})$ to this problem gives an upper bound as it is a minimization problem, i.e, $f(\boldsymbol{w}) \leq f(\hat{\boldsymbol{w}})$. Therefore, with given feasible solution $(\hat{\boldsymbol{w}}, \hat{\boldsymbol{b}})$, one can find a bound for $w_m,\ m \in M$ such that $w_m^d \in [-\sqrt{2f(\hat{\boldsymbol{w}})},\ \sqrt{2f(\hat{\boldsymbol{w}})}],\ d \in \{1,\ldots,n\},\ m \in M$ where w_m^d corresponds to d-th element of w_m. Any solution outside of this range gives higher objective value than given feasible solution which is out of interest. Therefore, (3.26) can be rewritten as follows:

$$\begin{aligned} \min\ & f(\boldsymbol{w}) \\ \text{s.t.}\ & \mathbb{P}(-L^{WW}(\boldsymbol{w}, \boldsymbol{b}) \geq 2) \geq \alpha \\ & -w_{bound} \leq w_m \leq w_{bound},\ m \in M \\ & -b_{bound} \leq b_m \leq b_{bound},\ m \in M. \end{aligned} \tag{4.11}$$

Here, $w_{bound} = \sqrt{2f(\hat{\boldsymbol{w}})}\mathbb{1}$, where $\mathbb{1}$ is n-dimensional vector of ones, and b_{bound} is a very large number. Let set of deterministic constraints be $\mathcal{D} = \{w_m \in [-w_{bound},\ w_{bound}],\ b_m \in [-b_{bound},\ b_{bound}],\ m \in M\}$ and index set for scenarios $\omega_i^m \in \bar{\Omega},\ i \in I,\ m \in M \setminus \{y_i\}$ be defined as $S = \{(i,m) | i \in I,\ m\ \in M \setminus \{y_i\}\}$. Also let $P_i^m = P(\omega_i^m)$ be region characterized by scenario $\omega_i^m \in \bar{\Omega}$ which is defined as:

$$P_i^m = \{(\boldsymbol{w}, \boldsymbol{b}) | (w_{y_i} - w_m)^T x_i + b_{y_i} - b_m \geq 2\}. \tag{4.12}$$

By introducing binary variables z_i^m for each $i \in I,\ m \in M \setminus \{y_i\}$ such that $z_i^m = 0 \implies (\boldsymbol{w}, \boldsymbol{b}) \in P_i^m$, Problem (4.11) can be reformulated using implication

constraints:

$$\min \ f(\boldsymbol{w}) \tag{4.13a}$$

$$\text{s.t.} \ \ z_i^m = 0 \implies (\boldsymbol{w}, \boldsymbol{b}) \in P_i^m, \ (i,m) \in S \tag{4.13b}$$

$$\sum_{(i,m)\in S} z_i^m \leq p \tag{4.13c}$$

$$(\boldsymbol{w}, \boldsymbol{b}) \in \mathcal{D} \tag{4.13d}$$

$$z_i^m \in \{0,1\}, \ (i,m) \in S, \tag{4.13e}$$

where $p = \lfloor l(k-1)(1-\alpha) \rfloor$. Then, the feasible region of (4.11) is $\mathcal{F} = \{(\boldsymbol{w}, \boldsymbol{b}, z) | (4.13b) - (4.13e)\}$.

4.2.1 Required Subproblems for VaR WW-MSVM

To solve VaR WW-MSVM, first subproblem is introduced as single scenario optimization problem for scenario $\omega_i^m \in \bar{\Omega}$:

$$h_i^m(\theta, \mu) = \min\{\theta^T \boldsymbol{w} + \mu^T \boldsymbol{b} | (\boldsymbol{w}, \boldsymbol{b}) \in P_i^m \cap \bar{\mathcal{D}}\}. \tag{4.14}$$

Note that $\mathcal{D}$ is a compact set, therefore, we choose $\bar{\mathcal{D}} = \mathcal{D}$. Proceeding from here, it is clear that the set $P_i^m \cap \bar{\mathcal{D}} = \{(\boldsymbol{w}, \boldsymbol{b}) | (w_{y_i} - w_m)^T x_i + b_{y_i} - b_m \geq 2, \ w_m \in [-w_{bound}, \ w_{bound}], \ b_m \in [-b_{bound}, \ b_{bound}], \ m \in M\}$ is compact. Also, when $\bar{\mathcal{D}} = \mathcal{D}$, Problem (4.14) is well defined, i.e, the optimal value exists and finite, as $\theta^T \boldsymbol{w} + \mu^T \boldsymbol{b}$ is real-valued and continuous on compact set $P_i^m \cap \bar{\mathcal{D}}$.
Secondly, single scenario separation procedure is introduced to check if found solution violates any of the scenarios and obtain parameters $(viol, \theta, \mu, \beta)$ to generate valid inequalities if violation exists. Note that each $P_i^m, \ i \in I, \ m \in M \backslash \{y_i\}$ is characterized by one inequality. Therefore, in order to ensure scenario ω_i^m is violated, it is sufficient to check $(\hat{w}_{y_i} - \hat{w}_m)^T x_i + \hat{b}_{y_i} - \hat{b} < 2$. Then, if violation exists, $viol = TRUE$, $\beta = 2$, $\theta^T \boldsymbol{w} = (\hat{w}_{y_i} - \hat{w}_m)^T x_i$ and $\mu^T \boldsymbol{b} = \hat{b}_{y_i} - \hat{b}_m$. Otherwise, $viol = FALSE$.

Final subproblem is introduced as master problem $MP(S_0, S_1, R)$:

$$\begin{aligned} \min \ & f(\boldsymbol{w}) \\ \text{s.t} \ & \sum_{(i,m)\in S} z_i^m \leq p, \\ & (\boldsymbol{w}, \boldsymbol{b}, z) \in R, \ (\boldsymbol{w}, \boldsymbol{b}) \in \mathcal{D} \\ & z_i^m \in [0,1], \ (i,m) \in S \\ & z_i^m = 0, (i,m) \in S_0, \ z_i^m = 1, (i,m) \in S_1, \end{aligned} \tag{4.15}$$

where R is a polyhedron that contains $\mathcal{F}$, and $S_0, S_1 \subseteq S$ are such that $S_0 \cap S_1 = \emptyset$.

4.3 Solving VaR CS-MSVM with Branch and Cut Decomposition Algorithm

Recall VaR CS-MSVM Problem (3.26). Let $f(\boldsymbol{w}) = \frac{1}{2}\sum_{m\in M} w_m^T w_m$. By the argument in Section 4.2, in order for Problem (4.4) to be well-defined, (3.26) can be rewritten as follows:

$$\begin{aligned} \min \ & f(\boldsymbol{w}) \\ \text{s.t.} \ & \mathbb{P}(-L^{CS}(\boldsymbol{w}, \boldsymbol{b}) \geq 1) \geq \alpha \\ & -w_{bound} \leq w_m \leq w_{bound}, \ m \in M \\ & -b_{bound} \leq b_m \leq b_{bound}, \ m \in M. \end{aligned} \tag{4.16}$$

Here, $w_{bound} = \sqrt{2f(\hat{\boldsymbol{w}})}\mathbb{1}$, where $\mathbb{1}$ is n-dimensional vector of ones and b_{bound} is a very large number. Let set of deterministic constraints be $\mathcal{D} = \{w_m \in [-w_{bound}, \ w_{bound}], \ b_m \in [-b_{bound}, \ b_{bound}], \ m \in M\}$ and I be the index set for scenarios $\omega_i \in \Omega$. Also let $P_i = P(\omega_i)$ be region characterized by scenario $\omega_i \in \Omega$ which is defined as:

$$P_i = \{(\boldsymbol{w}, \boldsymbol{b}) | (w_{y_i} - w_m)^T x_i + b_{y_i} - b_m \geq 1, \ m \in M \setminus \{y_i\}\}. \tag{4.17}$$

By introducing binary variables z_i for each $i \in I$ such that $z_i = 0 \implies (\boldsymbol{w}, \boldsymbol{b}) \in P_i$, Problem (4.16) can be reformulated using implication constraints:

$$\min \quad f(\boldsymbol{w}) \tag{4.18a}$$

$$\text{s.t.} \quad z_i = 0 \implies (\boldsymbol{w}, \boldsymbol{b}) \in P_i,\ i \in I \tag{4.18b}$$

$$\sum_{i \in I} z_i \leq p \tag{4.18c}$$

$$(\boldsymbol{w}, \boldsymbol{b}) \in \mathcal{D} \tag{4.18d}$$

$$z \in \{0,1\}^l, \tag{4.18e}$$

where $p = \lfloor l(1-\alpha) \rfloor$. Then, the feasible region of (4.16) is $\mathcal{F} = \{(\boldsymbol{w}, \boldsymbol{b}, z) | (4.18b) - (4.18e)\}$.

4.3.1 Required Subproblems for VaR CS-MSVM

To solve VaR CS-MSVM, first subproblem is introduced as single scenario optimization problem for scenario $\omega_i \in \Omega$:

$$h_i(\theta, \mu) = \min\{\theta^T \boldsymbol{w} + \mu^T \boldsymbol{b} | (\boldsymbol{w}, \boldsymbol{b}) \in P_i \cap \bar{\mathcal{D}}\}. \tag{4.19}$$

Note that $\mathcal{D}$ is a compact set, therefore, choosing $\bar{\mathcal{D}} = \mathcal{D}$ preserves feasibility. Proceeding from here, it is clear that the set $P_i \cap \bar{\mathcal{D}} = \{(\boldsymbol{w}, \boldsymbol{b}) | (w_{y_i} - w_m)^T x_i + b_{y_i} - b_m \geq 1,\ m \in M \setminus \{y_i\},\ w_m \in [-w_{bound},\ w_{bound}],\ b_m \in [-b_{bound},\ b_{bound}],\ m \in M\}$ is compact. Also, when $\bar{\mathcal{D}} = \mathcal{D}$, Problem (4.19) is well defined by discussion in Section 4.2.1.

Secondly, single scenario separation procedure is introduced to check if found solution violates any of the scenarios and obtain parameters $(viol, \theta, \mu, \beta)$ so that valid inequalities can be generated. Unlike WW-MSVM, in this problem $P_i,\ i \in I$ is characterized by more than one inequality. By the discussion in Section 4.1.1, if scenario ω_i is violated then there exists at least one $m \in M \setminus \{y_i\}$ such that $(\hat{w}_{y_i} - \hat{w}_m)^T x_i + \hat{b}_{y_i} - \hat{b}_m < 1$ and m^* that maximizes the argument $1 - [(\hat{w}_{y_i} - \hat{w}_m)^T x_i + \hat{b}_{y_i} - \hat{b}_m$ is chosen to obtain parameters $(viol, \theta, \mu, \beta)$. In this case, $viol = TRUE$ and we set $\theta^T \boldsymbol{w} = (\hat{w}_{y_i} - \hat{w}_{m^*})^T x_i$, $\mu^T \boldsymbol{b} = \hat{b}_{y_i} - \hat{b}_{m^*}$ and

$\beta = 1$. Otherwise, $viol = FALSE$. Final subproblem is introduced as master problem $MP(I_0, I_1, R)$:

$$\begin{aligned} \min \;\; & f(\boldsymbol{w}) \\ \text{s.t} \;\; & \sum_{i \in I} z_i \leq p, \\ & (\boldsymbol{w}, \boldsymbol{b}, z) \in R, \; (\boldsymbol{w}, \boldsymbol{b}) \in \mathcal{D} \\ & z_i \in [0,1], \; i \in I \\ & z_i = 0, i \in I_0, \; z_i = 1, i \in I_1, \end{aligned} \tag{4.20}$$

where R is a polyhedron that is described by valid inequalities and contains $\mathcal{F}$, and $I_0, I_1 \subseteq I$ are such that $I_0 \cap I_1 = \emptyset$.

4.4 Strong Big-M Formulation for VaR WW-MSVM

Consider the big-M formulation of VaR WW-MSVM (3.18). As coefficient B is chosen as a very large number, this formulation results in weak continuous relaxation leading poor computational performance. In this section, we propose a procedure to obtain strong coefficients using the argument discussed in [20]. Recall the valid inequalities in the form (4.9). For given θ and μ, let σ_s, $s \in \{1, \ldots, |S|\}$ be a permutation of S obtained by sorting h_i^m, $(i,m) \in S$ in descending order. The value of $h_{\sigma_s}(\theta, \mu)$ is obtained by solving single scenario optimization problem over all scenarios $\omega_i^m \in \bar{\Omega}$. Let a scenario ω_r^q, $(r,q) \in S$ be violated. Then, the value of (θ, μ) is obtained such that $\theta^T \boldsymbol{w} = (w_{y_r} - w_q)^T x_r$ and $\mu^T \boldsymbol{b} = b_{y_r} - b_q$ by the discussion in Section 4.2.1. Therefore, for a violated scenario ω_r^q, the single scenario optimization problem over $(t,d) \in S$ is given as:

$$\begin{aligned} h_t^d(\theta, \mu) = \min \;\; & (w_{y_r} - w_q)^T x_r + b_{y_r} - b_q \\ \text{s.t} \;\; & (w_{y_t} - w_d)^T x_t + b_{y_t} - b_d \geq 2 \\ & -w_{bound} \leq w_m \leq w_{bound}, \; m \in M \\ & -b_{bound} \leq b_m \leq b_{bound}, \; m \in M. \end{aligned} \tag{4.21}$$

Then for Problem (4.21), $\boldsymbol{w} = 0$, $\boldsymbol{b} \in \{\mathbb{R}^k | b_{y_t} - b_d = 2, b_{y_r} - b_q = 2, b_m \in [-b_{bound}, b_{bound}], m \in M\}$ is a feasible solution. Hence, optimal value of Problem (4.21) $h_i^m(\theta, \mu) \leq 2$ for $(t, d) \in S$. Particularly, for scenario ω_r^q, $h_r^q(\bar{\theta}, \bar{\mu}) = 2$ which implies $\sigma_1 = (r, q)$ and $h_{\sigma_1} = 2$. Resulting from this observation, one can conclude that for given $\theta^T \boldsymbol{w} = (w_{y_i} - w_m)^T x_i$ and $\mu^T \boldsymbol{b} = b_{y_i} - b_m$, $\sigma_1 = (i, m)$ and $h_{\sigma_1} = 2$ and the following inequality is valid for $\mathcal{F}$ as it is in the form (4.9):

$$(w_{y_i} - w_m)^T x_i + b_{y_i} - b_m + (h_{\sigma_1}(\theta, \mu) - h_{\sigma_{p+1}}(\theta, \mu)) z_i^m \geq h_{\sigma_1}(\theta, \mu), \tag{4.22}$$

or, equivalently:

$$(w_{y_i} - w_m)^T x_i + b_{y_i} - b_m + (2 - h_{\sigma_{p+1}}(\theta, \mu)) z_i^m \geq 2, \tag{4.23}$$

which is shown to be facet defining for convex hull of $\mathcal{F}$ [49], therefore, provides stronger coefficient for binary variable δ_i^m in Problem (3.18). By following this procedure for all $(i, m) \in S$, we obtain a strong big-M formulation for VaR WW-MSVM.

4.5 Strong Big-M Formulation for VaR CS-MSVM

Using the argument given in Section 4.4, same procedure can be applied to VaR CS-MSVM Problem (3.30). Let σ_i, $i \in I$ be a permutation of I obtained by sorting h_i, $i \in I$ in descending order. Recall that, $P(\omega_i)$ is characterized by more than one equality for each $\omega_i \in \Omega$. Let inequality q in $P(\omega_r)$, $r \in I$ be violated such that $(w_{y_r} - w_q)^T x_r + b_{y_r} - b_q < 1$. Then, we obtain $\theta^T \boldsymbol{w} = (w_{y_r} - w_q)^T x_r$ and $\mu^T \boldsymbol{b} = b_{y_r} - b_q$. Hence, the single scenario optimization problem over $t \in I$ is given as:

$$\begin{aligned} h_t(\theta, \mu) = \min \ & (w_{y_r} - w_q)^T x_r + b_{y_r} - b_q \\ \text{s.t} \ & (w_{y_t} - w_d)^T x_t + b_{y_t} - b_d \geq 1, \ d \in M \setminus \{y_t\} \\ & - w_{bound} \leq w_m \leq w_{bound}, \ m \in M \\ & - b_{bound} \leq b_m \leq b_{bound}, \ m \in M. \end{aligned} \tag{4.24}$$

For Problem (4.24), $\boldsymbol{w} = 0$, $\boldsymbol{b} \in \{\mathbb{R}^k | b_{y_t} - b_d = 1, d \in M \setminus \{y_t\}, b_m \in [-b_{bound}, b_{bound}], m \in M\}$ is a feasible solution. Therefore, optimal value of Problem (4.24) $h_t(\theta, \mu) \leq 1$ for $t \in I$. Particularly, for scenario ω_r, $h_r(\theta, \mu) = 1$ which implies $\sigma_1 = r$ and $h_{\sigma_1} = 1$. Similar to discussion in Section 4.4, for given $\theta^T \boldsymbol{w} = (w_{y_i} - w_m)^T x_i$ and $\mu^T \boldsymbol{b} = b_{y_i} - b_m$, $\sigma_1 = i$ and $h_{\sigma_1} = 1$ and the following inequality is valid for $\mathcal{F}$ as it is in the form (4.9):

$$(w_{y_i} - w_m)^T x_i + b_{y_i} - b_m + (h_{\sigma_1}(\theta, \mu) - h_{\sigma_{p+1}}(\theta, \mu)) z_i \geq h_{\sigma_1}(\theta, \mu), \tag{4.25}$$

or, equivalently:

$$(w_{y_i} - w_m)^T x_i + b_{y_i} - b_m + (1 - h_{\sigma_{p+1}}(\theta, \mu)) z_i \geq 1, \tag{4.26}$$

which is shown to be facet defining for convex hull of $\mathcal{F}$ [49], therefore, provides stronger coefficient for binary variable δ_i^m in Problem (3.30). By following this procedure for all $(i, m) \in S$, we obtain a strong big-M formulation for VaR CS-MSVM.

Chapter 5

Computational Study

In this chapter, we provide the computational results of risk-averse multi-class SVMs. For this study, two groups of artificial datasets are generated. First group consists of 14 artificial datasets used to measure the performance of risk-averse multi-class SVMs under the presence of noise, outliers and imbalance class distribution. Second group has 6 artificial datasets of different sizes to obtain a comparative study on the computational performance of strong big-M formulation, branch and cut decomposition algorithm and regular big-M formulation. Using the first group, in order to investigate the influence of ν, that is risk-aversion level, and the probability of occurrence of a class on the classification performance, we used different risk and probability levels. In the second group, we assume the probability of each sample is the same and consider two different levels of risk-aversion. Finally, we conclude this chapter with the experimental results for real-life datasets to illustrate the performance of risk-averse multi-class SVMs when existence of the outliers is an issue.

5.1 Artificial Datasets

To observe the performance of risk-averse multi-class SVMs, we generate 14 datasets consisting of 3 cases: first case contains 6 datasets with different number of samples in class 1; second case contains 3 datasets with different outlier levels and the last one contains 5 datasets with different noise locations. All datasets consist of 3 classes where the size of class 2 is equal to the size of class 3 and the size of class 1 depends on the data. Similarly, probability of observing class 2 is equal to the probability of observing class 3 where probability of observing class 1 changes depending on the setting. Description of the datasets generated to measure the performance of the risk-averse multi-class SVMs can be found in Table 5.1.

Data	Class 1			Class 2		Class 3	
Name	Size	# of Outliers	# of Noise	Size	# of Outliers	Size	# of Outliers
Ratio 1	100	-	-	100	-	100	-
Ratio 2	70	-	-	100	-	100	-
Ratio 3	50	-	-	100	-	100	-
Ratio 4	40	-	-	100	-	100	-
Ratio 5	20	-	-	100	-	100	-
Ratio 6	10	-	-	100	-	100	-
Outlier 1	120	20	-	120	20	120	20
Outlier 2	120	20	-	120	20	100	-
Outlier 3	120	20	-	100	-	100	-
Noise 1	100	-	3	100	-	100	-
Noise 2	100	-	6	100	-	100	-
Noise 3	100	-	3	100	-	100	-
Noise 4	5	-	5	100	-	100	-
Noise 5	5	-	5	100	-	100	-

Table 5.1: Datasets used to analyze the performance of risk-averse multi-class SVMs.

In Table 5.1, the first column of each class denotes the total number of samples in the corresponding class, the second column denotes the number of outliers in the class and last column denotes the number of samples that can be considered as noise. Particularly, in dataset Outlier 1, class 1 has 120 samples in total where 20 of them are placed as outliers. For Outlier 2 dataset, we place 20 outliers in each class 1 and class 2 while for Outlier 3 dataset, we have 20 outliers in each class. In Noise datasets, the samples marked as noise are located differently to investigate the impact of noise location and number of noisy samples on the performance. Specifically, in Noise 1 dataset, noisy samples are close to class 3 and in Noise 2 dataset, we generate additional samples that are close to class 2 while in Noise 3 dataset, we have only 3 noisy samples between class 2 and class 3. Different from these datasets, Noise 4 and Noise 5 datasets contain only 5 samples in class 1 in order to detect whether risk-averse multi-class SVMs are able to classify these samples correctly or they treat them as noise of other classes. For this case, the probability of observing class 1 and risk level parameter ν take values in the sets $\{0.01, 0.05, 0.1, 0.25, 0.33, 0.44, 0.55, 0.66, 0.77, 0.88, 0.99\}$ and $\{0.05, 0.1, 0.15, 0.2\}$, respectively.

We generate 6 datasets of sizes $30, 51, 150, 300, 510$ and 750 consisting of 3 classes with equal class sizes to perform a comparative study on the performances of strong big-M formulation, branch and cut decomposition algorithm and regular big-M formulation. Also, to analyze the impact of class size to computation time, we generate additional 2 datasets of size 150 with 4 and 5 classes where class sizes are equal. For these datasets, samples of class t are generated from Gaussian distribution with mean $\mu_t,\ t \in 1, \ldots, 5$ where $\mu_1 = [2, 0]$, $\mu_2 = [-2, -2]$, $\mu_3 = [-2, 2]$, $\mu_4 = [5, 0]$, $\mu_5 = [-5, 0]$ and with covariance matrix $\Sigma = 0.2I$. Here, I corresponds to an identity matrix in $\mathbb{R}^{2\times 2}$. For this study, it is assumed that all classes have same probability and the parameter ν is taken as 0.05 and 0.1.

For Ratio datasets, samples of class 1, class 2 and class 3 are generated from Gaussian distribution with mean $\mu_1,\ \mu_2,\ \mu_3$, respectively, and covariance matrix Σ, where $\mu_1 = [2, 0]$, $\mu_2 = [-2, -2]$, $\mu_3 = [-2, 2]$ and $\Sigma = 0.2I$, respectively. Similar to these datasets, for Noise datasets, class 1, class 2 and class 3 are generated from Gaussian distribution with mean $\mu_1 = [2, 0]$, $\mu_2 = [-2, -2]$, $\mu_3 = [-2, 2]$ and covariance matrix $\Sigma = 0.02I$. However, for Outlier datasets, we

generate samples inside circular regions with given class centers and radius. In this setting, class centers are $c_1 = [3, 0]$, $c_2 = [-4, -4]$, $c_3 = [-4, 3]$ for class 1, class 2 and class 3, respectively and radius is $r = 2$. Outliers are generated within an annulus having the same center as the class of outliers and with inner radius of 2 and outer radius of 2. Therefore, all samples are in 2-dimensional space so that the geometric analysis of the behavior of risk functions VaR and CVaR in multi-class SVMs can be provided.

5.2 Computational Results

In this section, we provide the computational results obtained from artificial datasets. For that purpose, first, we present the comparative study on the efficiency of the solution methods for VaR MSVMs in terms of objective value and optimality gap within the time limit of 3 hours in Section 5.2.1. Then, we compare the performance of VaR MSVM and CVaR MSVM methods under different noise, outlier and imbalance ratios with changing values of risk level and class 1 probability to conclude their stability to different cases. In Section 5.2.2, using the solutions obtained from the stability analysis, we provide the figures resulting from VaR MSVMs and CVaR MSVMs to analyze the behavior VaR and CVaR together with Weston and Watkins and Crammer and Singer methods. Finally in Section 5.2.3, we complete our analysis with the experimental results of original formulations, CVaR MSVMs and VaR MSVMs for real-life datasets to compare the stability of risk-averse models to presence of outliers.

5.2.1 Comparison of Solution Methods for VaR Multi-Class SVMs

In this study, we aim to give a solution methodology for VaR MSVM. For this purpose, we compare our proposed method, that is strong big-M formulation, to branch and cut decomposition algorithm [20] and big-M formulation where M is

taken as 100 for all settings. It is assumed that all classes have equal probability. We implement all methods in CPLEX. For comparison, we implement branch and cut algorithm with and without optional fractional cuts. Also, as CPLEX disables dynamic search and presolve features when user cuts are implemented, we also obtain results for regular big-M formulation with both disabled and enabled features. To make a concrete comparison, we have used datasets of different sizes and different number of classes. For this comparison, the risk-aversion parameter ν, which corresponds to the upper quantile of the loss distribution, i.e $\nu = 1 - \alpha$, is taken as 0.05 and 0.1. We report the objective value and optimality gap within 3 hours limit. The obtained results for VaR WW-MSVM and VaR CS-MSVM under different dataset sizes can be found in Table 5.2 and Table 5.3, respectively. The results illustrating the impact of the number of classes on the performance of the solution methods can be found in Table 5.4 and Table 5.5. In these tables, the second and third display the results of branch and cut decomposition algorithm with disabled and enabled fractional user cuts, respectively. Next two columns present the results of regular big-M formulation with enabled and disabled CPLEX features. The last column shows the results of strong big-M formulation. For each column, first sub-column denotes the best objective value found within the time limit; second sub-column corresponds to optimality gap; and the last sub-column represents the solution time.

Data (3 classes)		Branch & Cut (Fractional Disabled)			Branch & Cut (Fractional Enabled)			Big-M (M=100)			Big-M* (M=100)			Strong Big-M		
# Samples	ν	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time
30	0.05	0.7927	-	0.77s	0.7927	-	0.77s	0.7927	-	0.07s	0.7927	-	0.11s	0.7927	-	0.03s
30	0.1	0.6549	-	0.92s	0.6549	-	0.94s	0.6549	-	0.13s	0.6549	-	0.15s	0.6549	-	0.04s
51	0.05	1.0418	-	1.31s	1.0418	-	1.36s	1.0418	-	0.13s	1.0418	-	0.15s	1.0418	-	0.04s
51	0.1	0.7332	-	1.77s	0.7332	-	1.80s	0.7332	-	0.28s	0.7332	-	0.39s	0.7332	-	0.08s
150	0.05	0.8394	-	10.32s	0.8394	-	10.89s	0.8394	-	6.49s	0.8394	-	8.65s	0.8394	-	0.68s
150	0.1	0.7157	-	453s	0.7157	-	450s	0.7157	-	235s	0.7157	-	319s	0.7157	-	29.57s
300	0.05	0.8684	-	194s	0.8684	-	168s	0.8684	-	614s	0.8684	-	827s	0.8684	-	10.46s
300	0.1	0.7348	14.43%	3hr	0.7333	13.40%	3hr	0.7278	9.24%	3hr	0.7278	18.33%	3hr	0.7278	-	1618s
510	0.05	1.0015	10.25%	3hr	0.9982	9.98%	3hr	0.9899	14.26%	3hr	0.9899	25.84%	3hr	0.9899	-	531s
510	0.1	0.7975	39.97%	3hr	0.7963	42.08%	3hr	0.7756	55.57%	3hr	0.7756	67.44%	3hr	0.7756	15.90%	3hr
750	0.05	0.9972	28.20%	3hr	0.9955	29.29%	3hr	0.9824	51.66%	3hr	0.9824	65.52%	3hr	0.9824	4.87%	3hr
750	0.1	0.8177	47.37%	3hr	0.8158	46.23%	3hr	0.7918	100%	3hr	0.7918	100%	3hr	0.7893	29.81%	3hr

Table 5.2: Comparison of Branch and Cut Algorithm, Big-M Formulation and Strong Big-M Formulation under different dataset sizes for VaR WW-MSVM. * denotes Big-M formulation results when CPLEX features (presolve and dynamic search) are disabled.

Data (3 classes)		Branch & Cut (Fractional Disabled)			Branch & Cut (Fractional Enabled)			Big-M (M=100)			Big-M* (M=100)			Strong Big-M		
# Samples	ν	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time
30	0.05	0.2192	-	0.33s	0.2192	-	0.33s	0.2192	-	0.04s	0.2192	-	0.08s	0.2192	-	0.02s
30	0.1	0.1886	-	0.33s	0.1886	-	0.62s	0.1886	-	0.06s	0.1886	-	0.10s	0.1886	-	0.02s
51	0.05	0.3410	-	0.40s	0.3410	-	0.39s	0.3410	-	0.06s	0.3410	-	0.24s	0.3410	-	0.03s
51	0.1	0.2278	-	0.82s	0.2278	-	0.96s	0.2278	-	0.07s	0.2278	-	0.28s	0.2278	-	0.05s
150	0.05	0.2531	-	2.65s	0.2531	-	3.16s	0.2531	-	0.92s	0.2531	-	2.92s	0.2531	-	0.24s
150	0.1	0.2099	-	12.06s	0.2099	-	20.91s	0.2099	-	4.53s	0.2099	-	23s	0.2099	-	0.46s
300	0.05	0.2482	-	12.07s	0.2482	-	24.12s	0.2482	-	4.74s	0.2482	-	4.74s	0.2482	-	0.42s
300	0.1	0.2124	9.55%	3hr	0.2102	-	1547s	0.2102	-	143.22s	0.2102	-	1087s	0.2102	-	10.44s
510	0.05	0.3054	-	9737s	0.3054	-	1003s	0.3054	-	279.37s	0.3054	-	977s	0.3054	-	27.09s
510	0.1	0.2497	33.30%	3hr	0.2462	21.19%	3hr	0.2429	15.31%	3hr	0.2429	34.67%	3hr	0.2429	-	146s
750	0.05	0.2881	16.13%	3hr	0.2832	-	6271	0.2832	-	1539s	0.2832	19.16%	3hr	0.2832	-	65.82s
750	0.1	0.2470	47.37%	3hr	0.2464	40.79%	3hr	0.2392	27.92%	3hr	0.2397	71.12%	3hr	0.2392	-	4534s

Table 5.3: Comparison of Branch and Cut Algorithm, Big-M Formulation and Strong Big-M Formulation under different dataset sizes for VaR CS-MSVM. * denotes Big-M formulation results when CPLEX features (presolve and dynamic search) are disabled.

Data (150 samples)		Branch & Cut (Fractional Disabled)			Branch & Cut (Fractional Enabled)			Big-M (M=100)			Big-M* (M=100)			Strong Big-M		
# of Classes	ν	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time
3	0.05	0.8394	-	10.32s	0.8394	-	10.89s	0.8394	-	6.49s	0.8394	-	8.65s	0.8394	-	0.68s
3	0.1	0.7157	-	453s	0.7157	-	450s	0.7157	-	235s	0.7157	-	319s	0.7157	-	29.57s
4	0.05	1.7656	-	2097s	1.7656	-	1834s	1.7656	-	2155s	1.7656	-	2883s	1.7656	-	83.68s
4	0.1	1.4633	81.14%	3hr	1.4593	79.12%	3hr	1.4185	51.24%	3hr	1.4185	62.49%	3hr	1.4185	49.62%	3hr
5	0.05	3.2913	90.07%	3hr	3.2439	89.91%	3hr	3.1822	59.97%	3hr	3.1822	64.72%	3hr	3.1822	35.89%	3hr
5	0.1	2.6705	100%	3hr	2.6698	100%	3hr	2.5961	88.21%	3hr	2.6137	92.69%	3hr	2.5961	86.95%	3hr

Table 5.4: Comparison of Branch and Cut Algorithm, Big-M Formulation and Strong Big-M Formulation under different number of classes for VaR WW-MSVM. * denotes Big-M formulation results when CPLEX features (presolve and dynamic search) are disabled.

Data (150 samples)		Branch & Cut (Fractional Disabled)			Branch & Cut (Fractional Enabled)			Big-M (M=100)			Big-M* (M=100)			Strong Big-M		
# of Classes	ν	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time	Obj	Gap	Time
3	0.05	0.2531	-	2.65s	0.2531	-	3.16s	0.2531	-	0.92s	0.2531	-	2.92s	0.2531	-	0.24s
3	0.1	0.2099	-	12.06s	0.2099	-	20.91s	0.2099	-	4.53s	0.2099	-	23s	0.2099	-	0.46s
4	0.05	0.5525	-	3.66s	0.5525	-	4.35s	0.5525	-	2.4s	0.5525	-	8.33s	0.5525	-	0.55s
4	0.1	0.4696	-	145s	0.4696	-	73s	0.4696	-	59s	0.4696	-	174s	0.4696	-	5.67s
5	0.05	1.2039	-	6.30s	1.2039	-	12.98s	1.2039	-	8.41s	1.2039	-	49s	1.2039	-	1.71s
5	0.1	0.9684	-	3456s	0.9684	-	297s	0.9684	-	934s	0.9684	-	2999s	0.9684	-	31s

Table 5.5: Comparison of Branch and Cut Algorithm, Big-M Formulation and Strong Big-M Formulation under different number of classes for VaR CS-MSVM. * denotes Big-M formulation results when CPLEX features (presolve and dynamic search) are disabled.

Table 5.2 presents the results of solution methods for VaR WW-MSVM to show the impact of dataset size together with risk-aversion level on the computational performance when each dataset has 3 classes with equal sizes. The results show that strong big-M formulation outperforms the other methods in each criterion. For the datasets which are solved to optimality by all methods, strong big-M formulation provides significantly faster results than the others. With the increase in the sample size, while branch and cut decomposition methods and regular big-M methods cannot solve the problems, strong big-M formulation either gives optimal solution or yields lower optimality gap. Particularly, for the datasets where regular big-M formulations cannot improve the lower bound, strong big-M formulation surpasses the other methods in terms of objective value together with optimality gap. Also, the experiments on branch and cut decomposition algorithm show that enabling fractional cuts improves lower bound and gives better objective value. For small sized datasets, regular big-M formulation, with or without CPLEX features, is faster than the branch and cut decomposition methods. However, with increase in sample size, we observe that branch and cut algorithm surpasses the regular big-M formulation in optimality gap, yet, regular big-M formulation results in better objective value. Finally, increase in ν leads to decrease in the computational performance due to larger feasible region.

Table 5.3 provides the comparison of solution methods for VaR CS-MSVM to illustrate the impact of dataset size and risk-aversion level on the computational performance when each dataset has 3 classes with equal sizes. The results show that strong big-M formulation outperforms the other methods in terms of each criterion. Also, branch and cut decomposition algorithms cannot beat regular big-M formulation. For small sized datasets, it can be observed that enabling fractional cuts affects the solution time negatively, however, as dataset size increases, it improves lower bound and objective value. As a final remark, it is concluded that VaR CS-MSVM outperforms VaR WW-MSVM in terms of solution time and optimality gap when they are compared under the same datasets. Recall that a sample may commit margin violations with respect to several separating hyperplanes. VaR WW-MSVM considers each violation as a different realization of random loss, while VaR CS-MSVM considers the maximum of margin violations committed by a sample as a realization of random loss. Therefore,

VaR WW-MSVM solves the problem considering more scenarios which affects the solution time.

Table 5.4 demonstrates the results of solution methods for VaR WW-MSVM to emphasize the impact of number of classes on the computational performance when each dataset has 150 samples. The results show that the computational performance of the solution methods are considerably degrades when number of classes increases due to increasing number of scenarios. Particularly, branch and cut decomposition algorithms give considerably poor results compared to the other methods. Similar to previous analysis, strong big-M formulation outperforms the other methods in each criterion.

Table 5.5 displays the results of solution methods for VaR CS-MSVM for number of classes 3, 4 and 5. Unlike the results for VaR WW-MSVM, Table 5.5 illustrates that all instances can be solved within the given time limit. The results present that, while number of classes increases, the proposed big-M formulation surpasses the other methods in terms of solution time. Also, it is observed that number of classes influences the computational performance of regular big-M formulation more compared to the other methods. For increasing number of classes, branch and cut decomposition algorithms perform better than regular big-M formulation and enabling fractional cuts considerably improves the solution time.

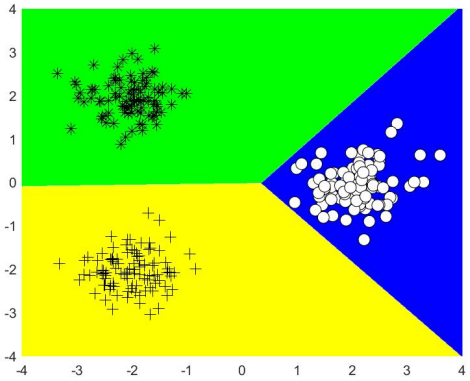
5.2.2 Geometric Analysis of Risk-Averse Multi-Class SVMs

In this section, we analyze VaR and CVaR in multi-class SVMs. For this analysis, we compare WW-MSVM, CS-MSVM, CVaR WW-MSVM, CVaR CS-MSVM, VaR WW-MSVM and VaR CS-MSVM formulations for all datasets given in Table 5.1 under different probabilities of class 1 and ν values. With this analysis, we aim to provide a comparative study that illustrates the behavior of risk measures under different values of probability and risk-level when presence of outliers, noise and imbalance class distribution is an issue. In all the figures below, blue region represents class 1, yellow region represents class 2 and green region represents class 3.

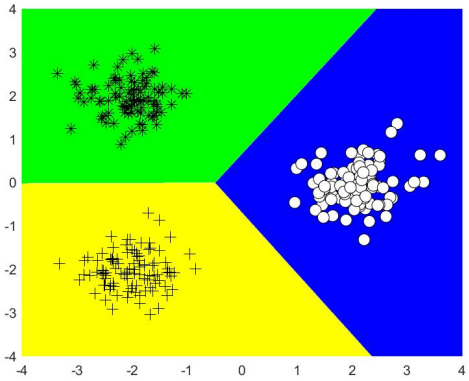
5.2.2.1 Comparison of WW-MSVM and CVaR WW-MSVM

In this section, we analyze the sensitiveness of WW-MSVM and CVaR WW-MSVM to the probability and size of classes and noise. In all the analysis provided below, the risk-aversion level ν is set to 0.1.

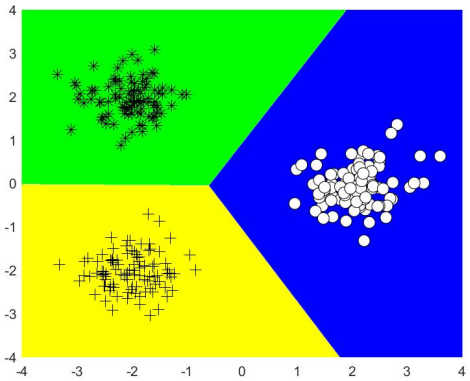
Case Ratio 1 dataset: In this case, we analyze the impact of class probabilities on the performance of WW-MSVM and CVaR WW-MSVM when class sizes are equal.



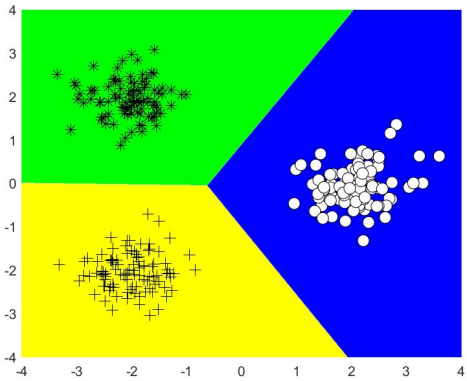
(a) WW-MSVM,
0.1 class 1 probability



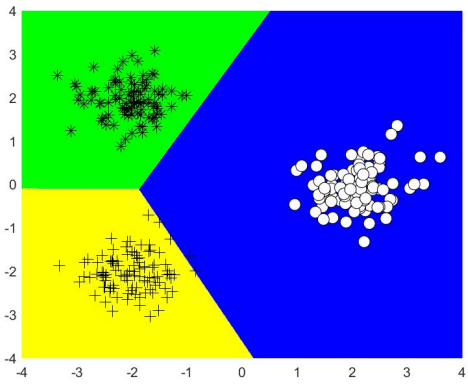
(b) CVaR WW-MSVM,
0.1 class 1 probability



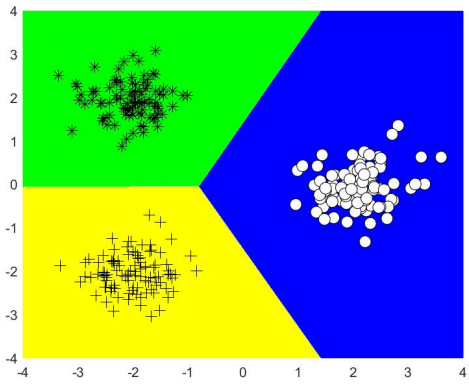
(c) WW-MSVM,
0.33 class 1 probability



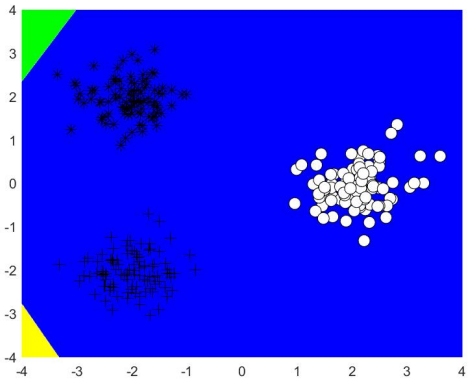
(d) CVaR WW-MSVM,
0.33 class 1 probability



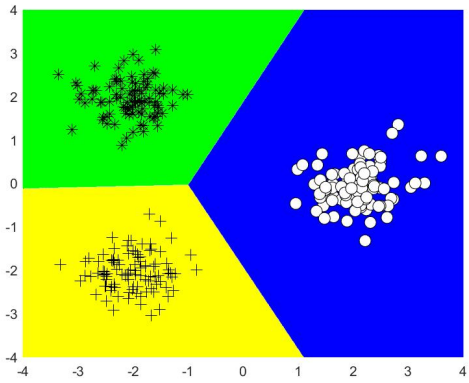
(e) WW-MSVM,
0.77 class 1 probability



(f) CVaR WW-MSVM,
0.77 class 1 probability



(g) WW-MSVM,
0.88 class 1 probability

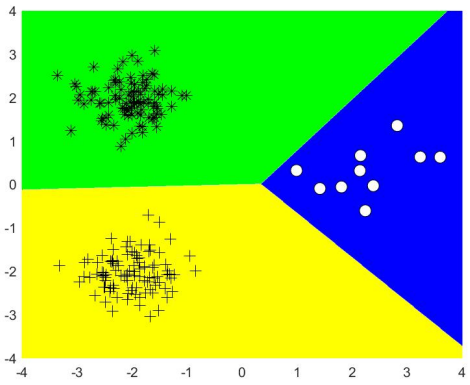


(h) CVaR WW-MSVM,
0.88 class 1 probability

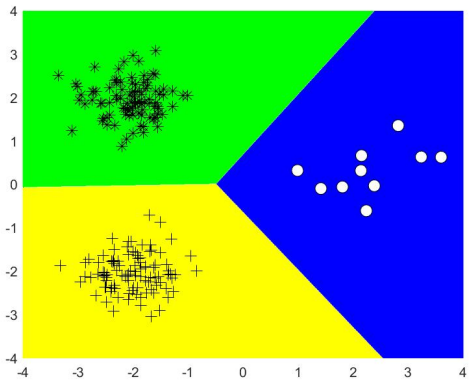
Figure 5.1: Comparison of WW-MSVM and CVaR WW-MSVM with $\nu = 0.1$ under different class 1 probabilities for Ratio 1 dataset.

As seen in Figure 5.1, as probability of class 1 increases, both WW-MSVM and CVaR WW-MSVM tend to reserve a larger region for class 1. However, while there is an apparent expansion in the region of class 1 in WW-MSVM, this expansion is subtle in CVaR WW-MSVM. Furthermore, when the probability of class 1 is high, WW-MSVM provides classification regions that result in misclassification, yet, CVaR WW-MSVM separates classes without any misclassification. These conclusions can be extended to the comparison of CS-MSVM and CVaR CS-MSVM under the same setting as seen in Figure A.1

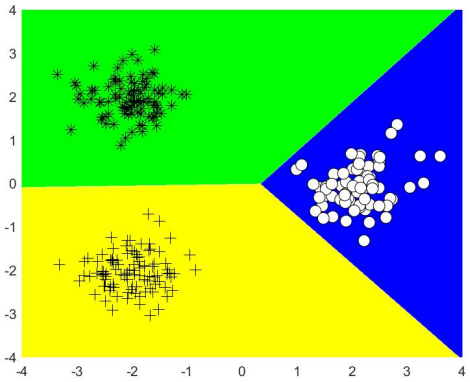
Case Ratio 2 and 6 datasets: Here, we analyze the performance of WW-MSVM and CVaR WW-MSVM when the size of class 1 changes. We use Ratio 2 and 6 datasets and set the probability of class 1 to two different values, 0.1 and 0.88.



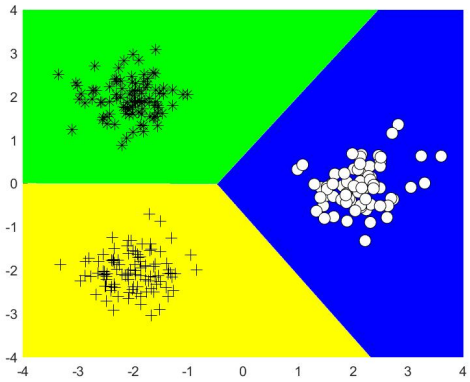
(a) WW-MSVM, Ratio 6 dataset



(b) CVaR WW-MSVM, Ratio 6 dataset



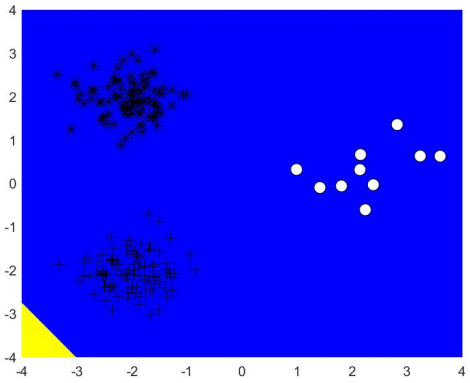
(c) WW-MSVM,
Ratio 2 dataset



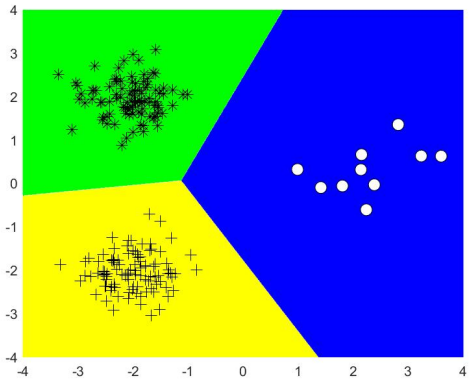
(d) CVaR WW-MSVM,
Ratio 2 dataset

Figure 5.2: Comparison of WW-MSVM and CVaR WW-MSVM with $\nu = 0.1$ under 0.1 class 1 probability for Ratio 2 and 6 datasets

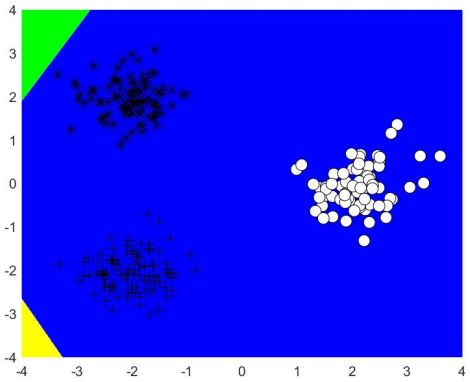
As observed from Figure 5.2, when the probability of class 1 is low, both WW-MSVM and CVaR WW-MSVM are less sensitive to the size of class 1.



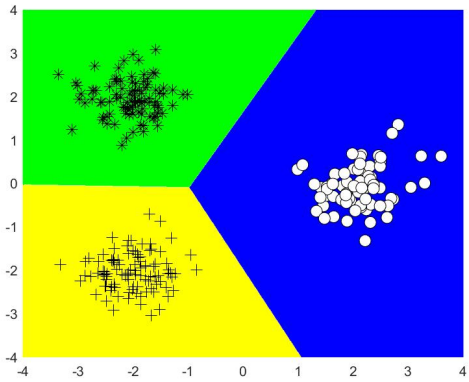
(a) WW-MSVM,
Ratio 6 dataset



(b) CVaR WW-MSVM,
Ratio 6 dataset



(c) WW-MSVM,
Ratio 2 dataset

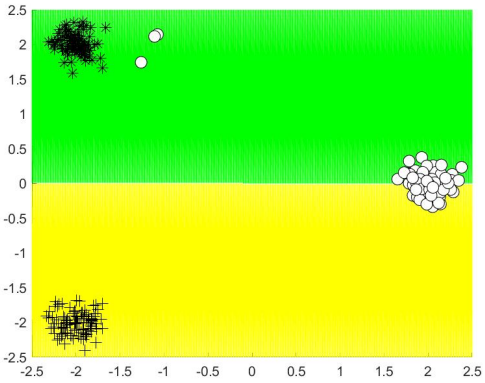


(d) CVaR WW-MSVM,
Ratio 2 dataset

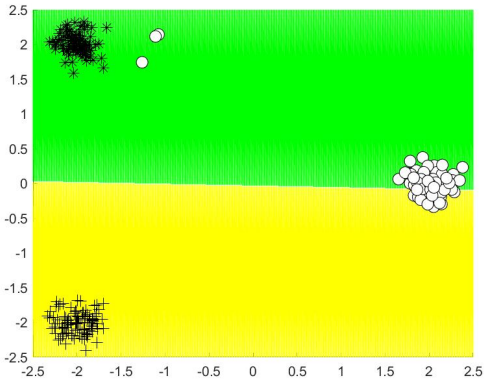
Figure 5.3: Comparison of WW-MSVM and CVaR WW-MSVM with $\nu = 0.1$ under 0.88 class 1 probability for Ratio 2 and 6 datasets

In Figure 5.3, we analyze the impact of the size of class 1 on the classification regions when the probability of class 1 is high. Again, both WW-MSVM and CVaR WW-MSVM are less sensitive to the size of class 1.

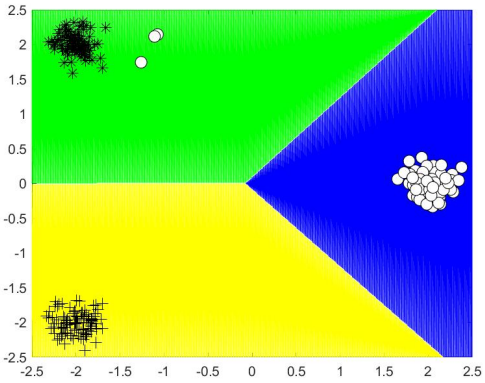
Case Noise 1 dataset: This case provides the comparison of WW-MSVM and CVaR WW-MSVM under different class 1 probabilities for Noise 1 dataset where each class has 100 samples and class 1 has 3 noisy samples located close to class 3.



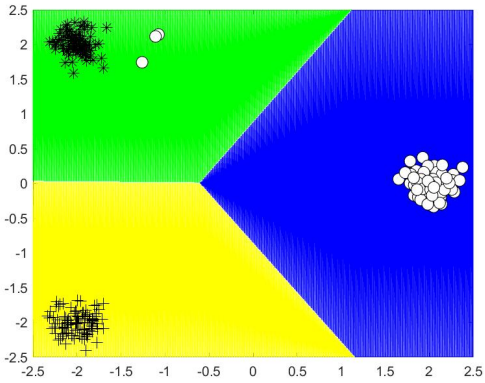
(a) WW-MSVM,
0.01 class 1 probability



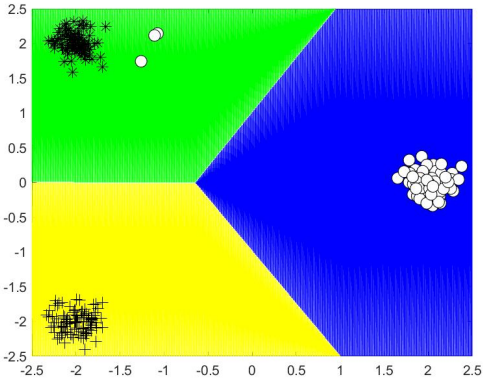
(b) CVaR WW-MSVM,
0.01 class 1 probability



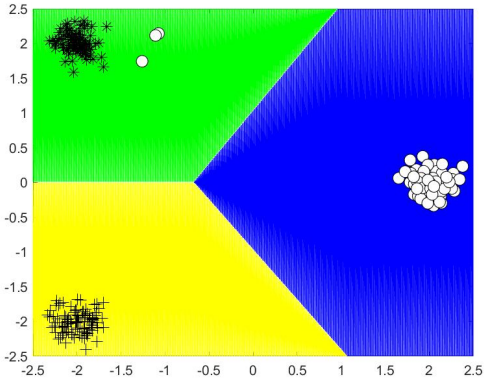
(c) WW-MSVM,
0.1 class 1 probability



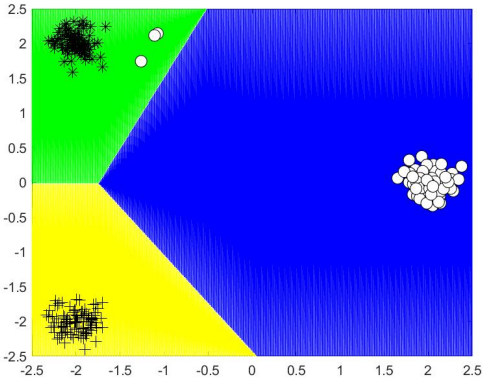
(d) CVaR WW-MSVM,
0.1 class 1 probability



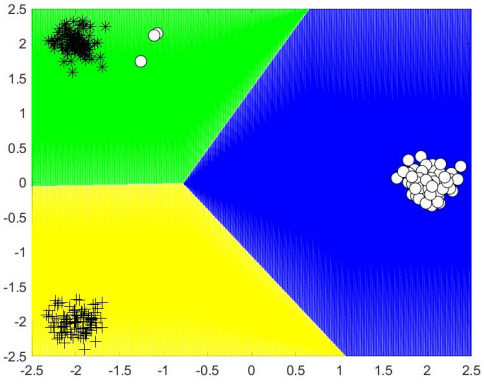
(e) WW-MSVM,
0.33 class 1 probability



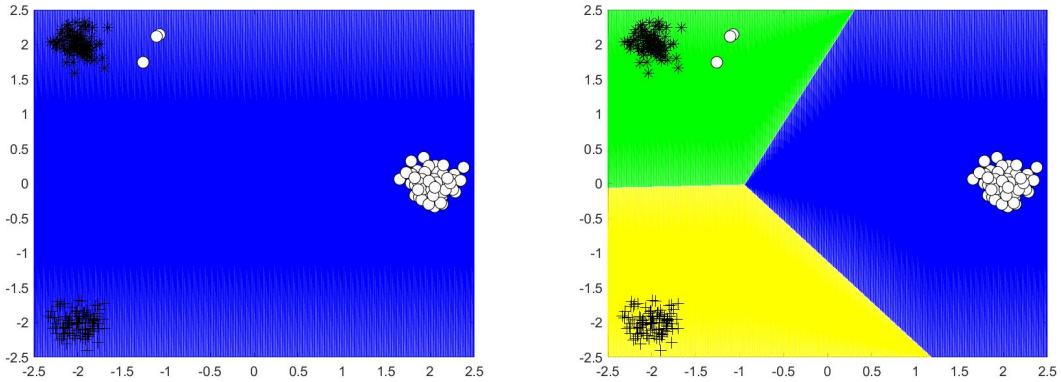
(f) CVaR WW-MSVM,
0.33 class 1 probability



(g) WW-MSVM,
0.77 class 1 probability



(h) CVaR WW-MSVM,
0.77 class 1 probability



(i) WW-MSVM,
0.88 class 1 probability

(j) CVaR WW-MSVM,
0.88 class 1 probability

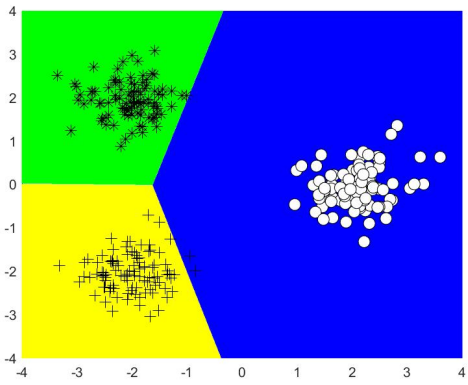
Figure 5.4: Comparison of WW-MSVM and CVaR WW-MSVM with $\nu = 0.1$ under different class 1 probabilities for Noise 1 dataset.

As seen in Figure 5.4, when the probability of class 1 increases, WW-MSVM becomes more sensitive to the noise. For a probability of 0.01, WW-MSVM is unable to classify class 1, whereas, when probability is 0.88, it classifies all samples as class 1. However, CVaR WW-MSVM is less sensitive to the presence of noise and it can separate the classes for all probability values except 0.01. Same conclusion can be deduced for CS-MSVM and CVaR CS-MSVM as indicated in Figure A.4.

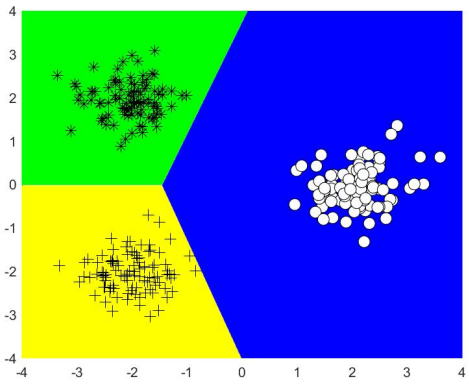
5.2.2.2 Comparison of CVaR WW-MSVM and CVaR CS-MSVM

In this section, we compare CVaR WW-MSVM and CVaR CS-MSVM under different levels of risk-aversion, different class sizes and probabilities, and existence of noise and outliers.

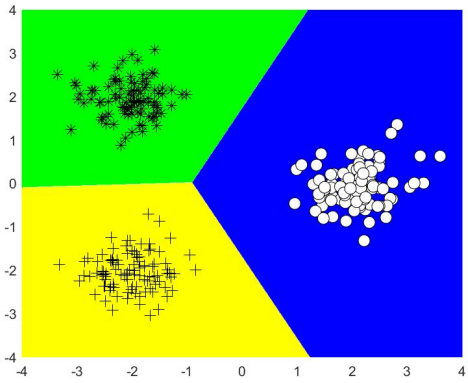
Case Ratio 1 dataset: This case presents the effect of ν, that is level of risk-aversion, on the performance of CVaR WW-MSVM and CVaR CS-MSVM when probability of class 1 is 0.88 and class sizes are equal.



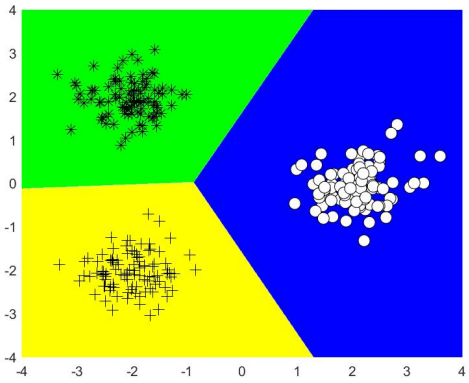
(a) CVaR WW-MSVM, $\nu = 0.2$



(b) CVaR CS-MSVM, $\nu = 0.2$



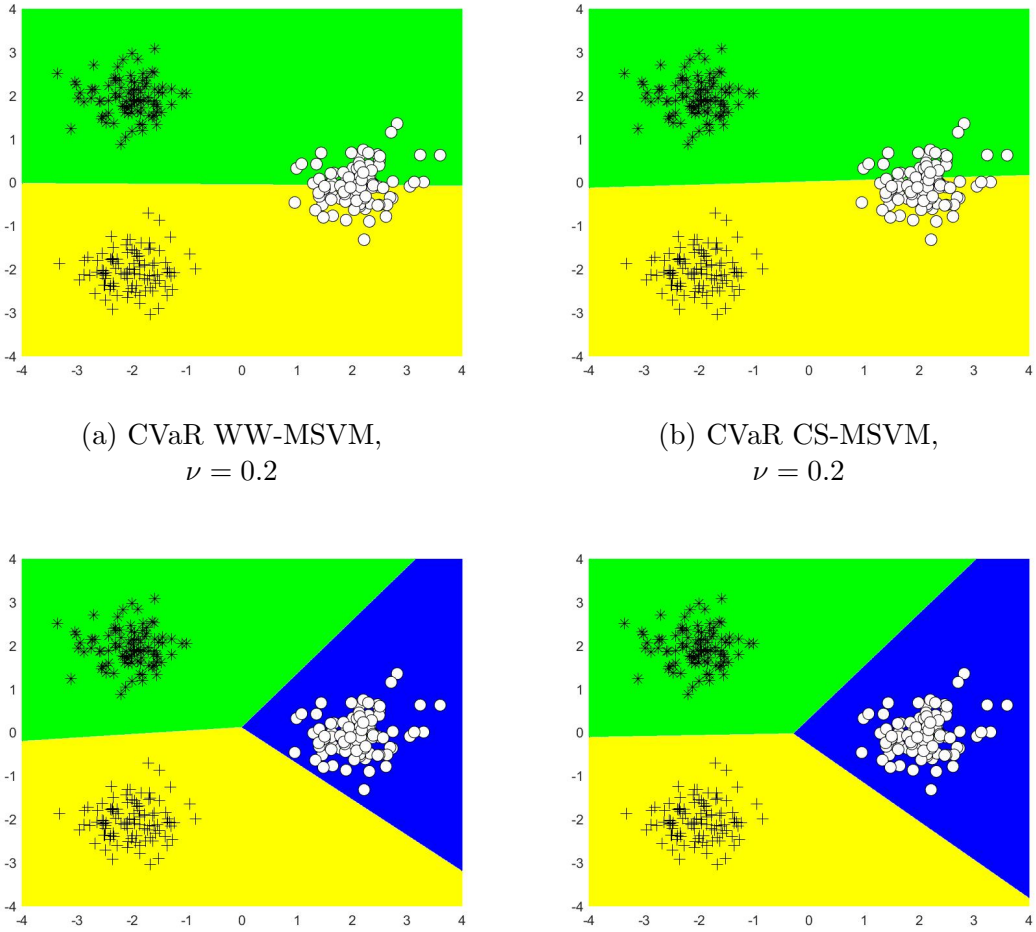
(c) CVaR WW-MSVM, $\nu = 0.05$



(d) CVaR CS-MSVM, $\nu = 0.05$

Figure 5.5: Comparison of CVaR WW-MSVM and CVaR CS-MSVM with different ν values under 0.88 class 1 probability for Ratio 1 dataset.

As seen in Figure 5.5, as level of risk-aversion increases, that is ν decreases from 0.2 to 0.05, both CVaR WW-MSVM and CVaR CS-MSVM get less responsive to high class 1 probability, that is both formulations provide stable classification regions.



(a) CVaR WW-MSVM, $\nu = 0.2$

(b) CVaR CS-MSVM, $\nu = 0.2$

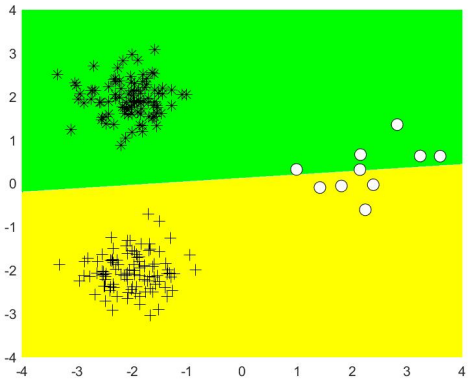
(c) CVaR WW-MSVM, $\nu = 0.05$

(d) CVaR CS-MSVM, $\nu = 0.05$

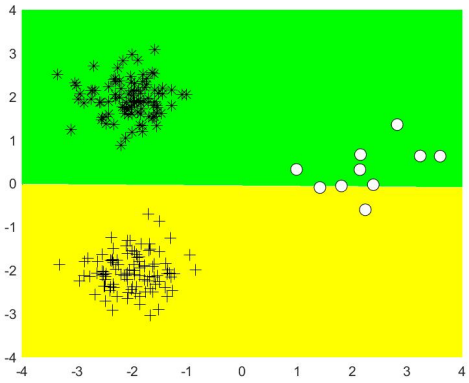
Figure 5.6: Comparison of CVaR WW-MSVM and CVaR CS-MSVM with different ν values under 0.01 class 1 probability for Ratio 1 dataset.

As observed in Figure 5.6, for low levels of risk-aversion such as $\nu = 0.2$, both CVaR WW-MSVM and CVaR CS-MSVM are unable to classify class 1. However, as ν decreases from 0.2 to 0.05, both CVaR WW-MSVM and CVaR CS-MSVM get less sensitive to low class 1 probability and provide stable classification regions similar to Figure 5.5. Therefore, the impact of the probability of a class on the classification performance of CVaR MSVMs reduces as risk-aversion level increases.

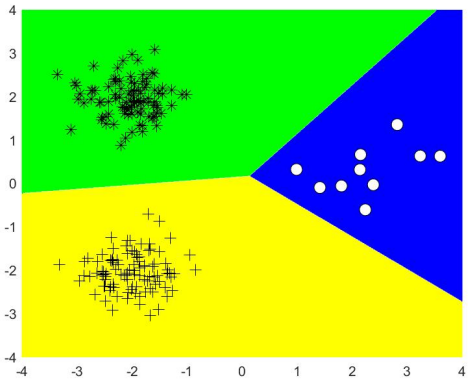
Case Ratio 6 dataset: This case examines the impact of ν on the performance of CVaR WW-MSVM and CVaR CS-MSVM under unequal class sizes and class 1 probabilities of 0.01 and 0.88.



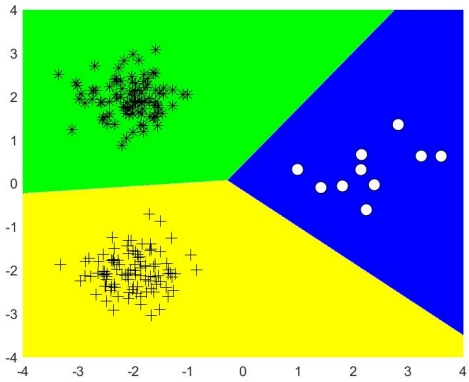
(a) CVaR CS-MSVM, $\nu = 0.2$



(b) CVaR WW-MSVM, $\nu = 0.2$



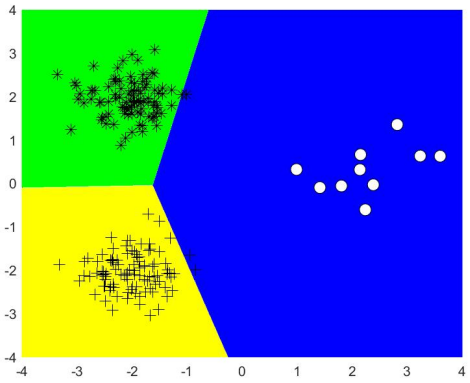
(c) CVaR WW-MSVM, $\nu = 0.05$



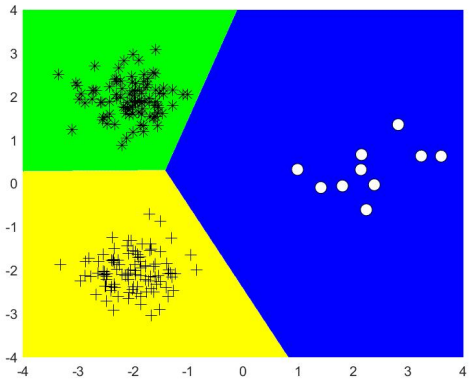
(d) CVaR CS-MSVM, $\nu = 0.05$

Figure 5.7: Comparison of CVaR WW-MSVM and CVaR CS-MSVM with different ν values under 0.01 class 1 probability for Ratio 6 dataset.

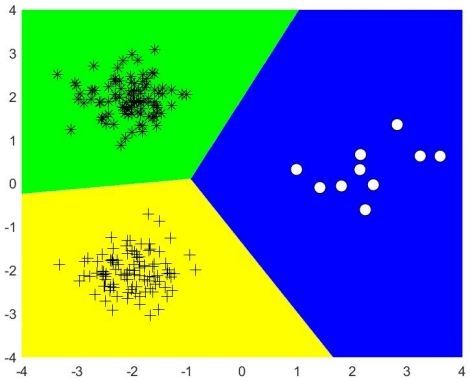
As observed in Figure 5.7, as ν decreases from 0.2 to 0.05, in other words, as level of risk-aversion increases, both CVaR WW-MSVM and CVaR CS-MSVM expand the region of class 1 when the probability of class 1 is low. As ν denotes the risk-aversion level, we can observe how CVaR WW-MSVM and CVaR CS-MSVM perceive risk. Here, misclassification of the samples belonging to the class with low probability is perceived as risk.



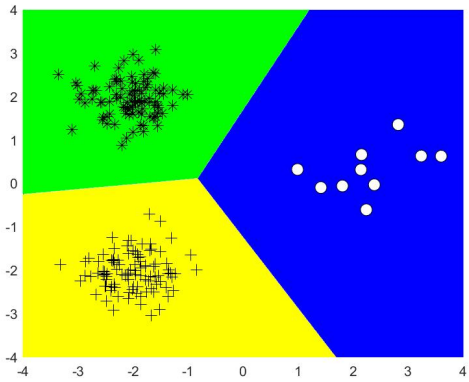
(a) CVaR WW-MSVM, $\nu = 0.2$



(b) CVaR CS-MSVM, $\nu = 0.2$



(c) CVaR WW-MSVM, $\nu = 0.05$

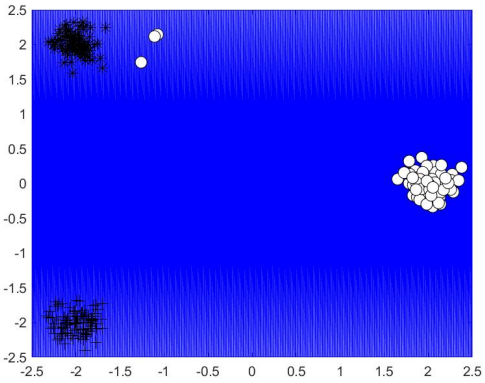


(d) CVaR CS-MSVM, $\nu = 0.05$

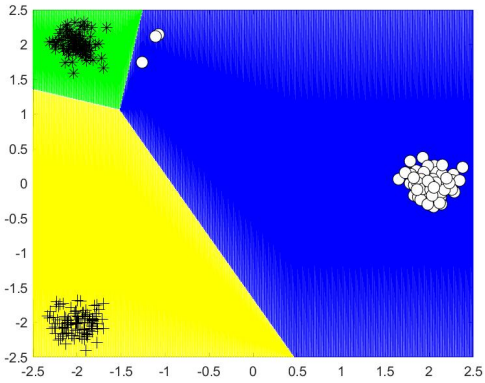
Figure 5.8: Comparison of CVaR WW-MSVM and CVaR CS-MSVM with different ν values under 0.88 class 1 probability for Ratio 6 dataset.

In Figure 5.8, we investigate the impact of ν on the classification regions when the probability of class 1 is high. In both CVaR WW-MSVM and CVaR CS-MSVM, the region of class 1 shrinks when ν decreases from 0.2 to 0.05. In other words, both models expand the regions of the classes with low probability as risk-aversion level increases. Therefore, the perception of risk for CVaR MSVMs is supported.

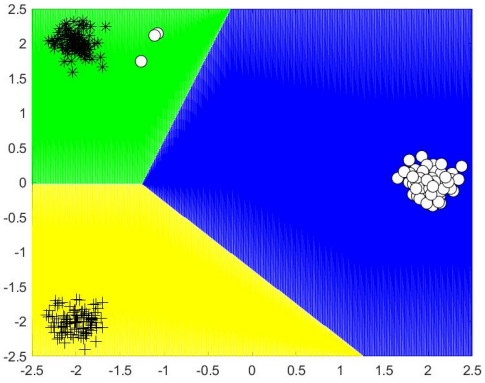
Case Noise 1 dataset: In this case, for class 1 probability of 0.88, we analyze the impact of ν on the construction of the separating hyperplanes when class 1 has 3 noisy samples located close to class 3.



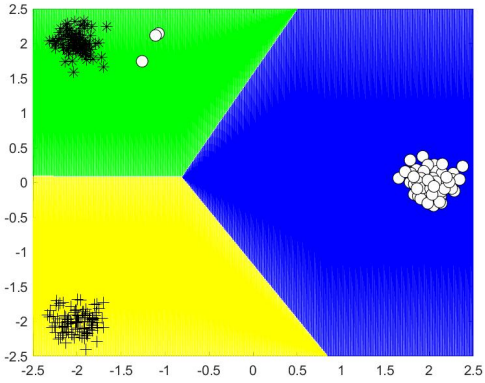
(a) CVaR WW-MSVM, $\nu = 0.2$



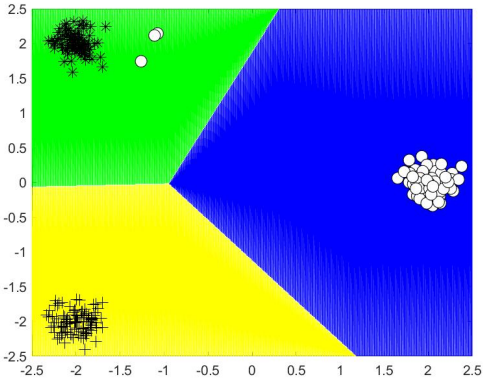
(b) CVaR CS-MSVM, $\nu = 0.2$



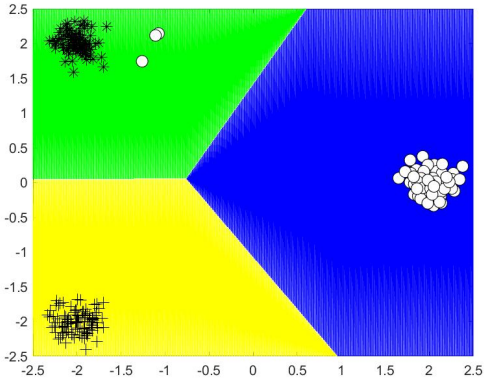
(c) CVaR WW-MSVM, $\nu = 0.15$



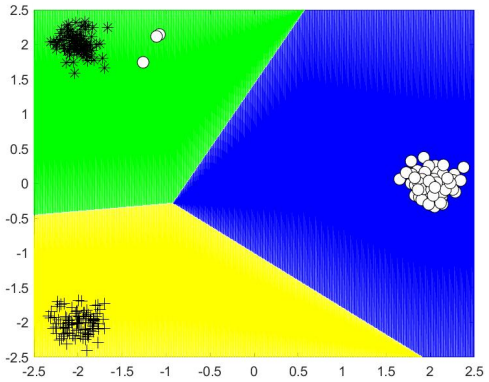
(d) CVaR CS-MSVM, $\nu = 0.15$



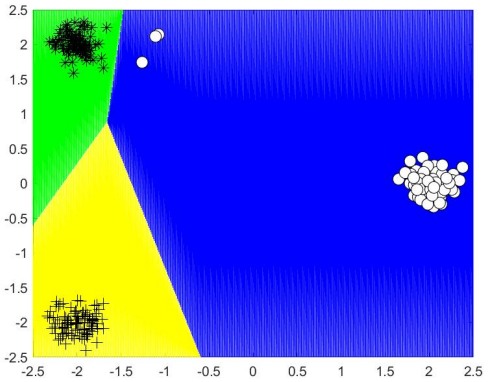
(e) CVaR WW-MSVM, $\nu = 0.1$



(f) CVaR CS-MSVM, $\nu = 0.1$



(g) CVaR WW-MSVM, $\nu = 0.05$

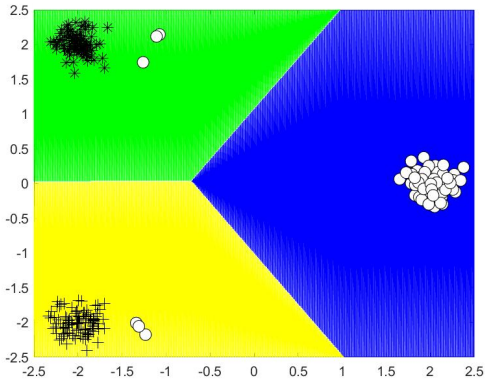


(h) CVaR CS-MSVM, $\nu = 0.05$

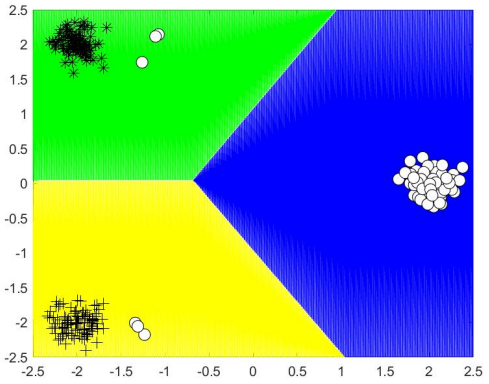
Figure 5.9: Comparison of CVaR WW-MSVM and CVaR CS-MSVM with different ν values under 0.88 class 1 probability for Noise 1 dataset.

As seen in Figure 5.9, when risk-aversion level is low, that is $\nu = 0.2$, CVaR WW-MSVM fails to separate the classes and CVaR CS-MSVM produces classification regions considering noisy samples. As risk-aversion level increases to 0.1, the region of class 1 shrinks in both models. Particularly, in CVaR WW-MSVM, the region of class 1, blue region, is larger implying noise location affects the classification regions. However, for high risk-aversion level, that is $\nu = 0.05$, CVaR CS-MSVM overfits the dataset. Hence, when the probability of class 1 is high, CVaR CS-MSVM is more stable for moderate levels of risk-aversion. On the other hand, for very high levels of risk-aversion, CVaR WW-MSVM gets more stable.

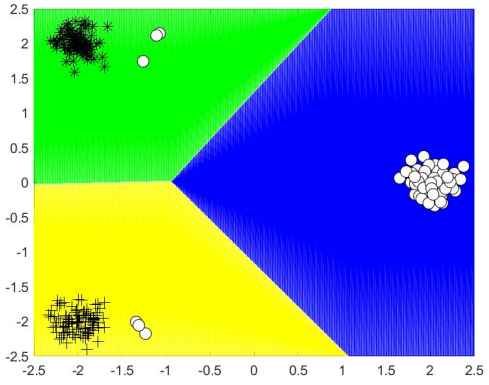
Case Noise 2 dataset: This case investigates the impact of class probability and ν on the performance of CVaR WW-MSVM and CVaR CS-MSVM for Noise 2 dataset where class 1 contains 6 noisy samples evenly distributed close to the remaining 2 classes.



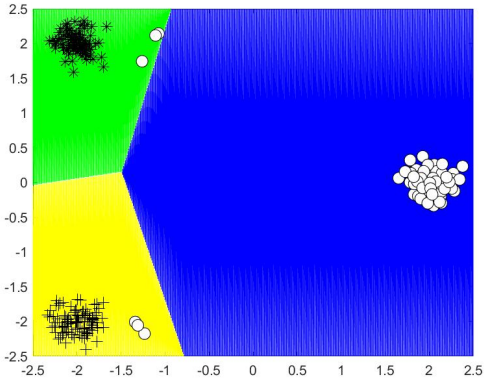
(a) CVaR WW-MSVM,
0.33 class 1 probability



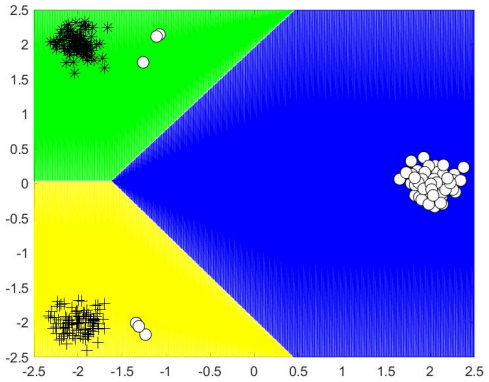
(b) CVaR CS-MSVM,
0.33 class 1 probability



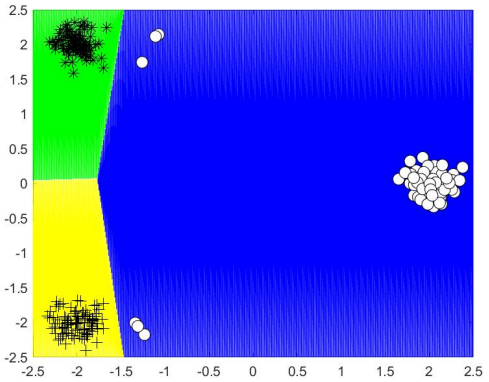
(c) CVaR WW-MSVM,
0.77 class 1 probability



(d) CVaR CS-MSVM,
0.77 class 1 probability



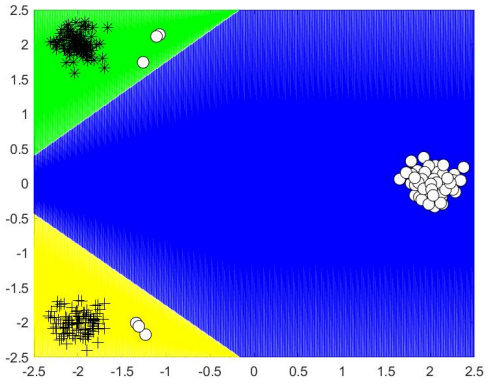
(e) CVaR WW-MSVM, 0.88 class 1 probability



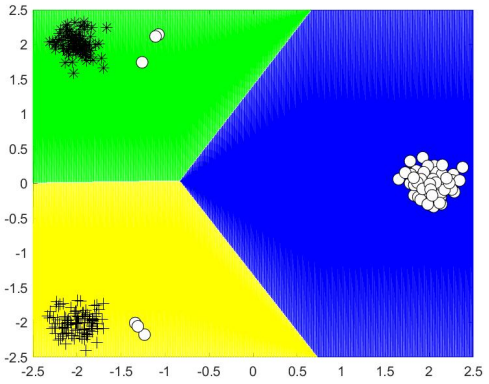
(f) CVaR CS-MSVM, 0.88 class 1 probability

Figure 5.10: Comparison of CVaR WW-MSVM and CVaR CS-MSVM with $\nu = 0.1$ under different class 1 probabilities for Noise 2 dataset.

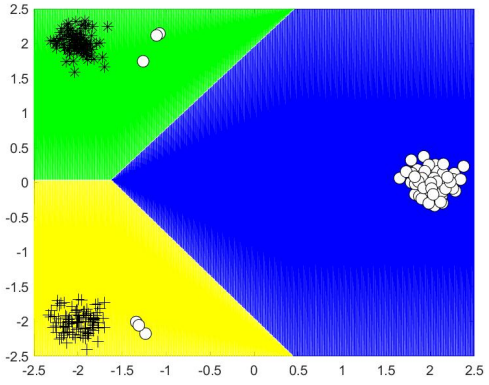
As observed in Figure 5.10, as the probability of class 1 increases, the expansion in the region of class 1 is more apparent for CVaR CS-MSVM such that high class 1 probability results in overfitting. Recall Figure 5.9f that when $\nu = 0.1$, CVaR CS-MSVM ignores the noisy samples for Noise 1 dataset when the probability of class 1 is 0.88. Therefore, the number of noisy samples is a more critical parameter for CVaR CS-MSVM which may lead to overfitting. On the other hand, the probability of class 1 has more restrainable effect on the classification regions produced by CVaR WW-MSVM. From this observation, we can conclude that CVaR WW-MSVM is more stable to different noise levels for different class 1 probabilities when $\nu = 0.1$.



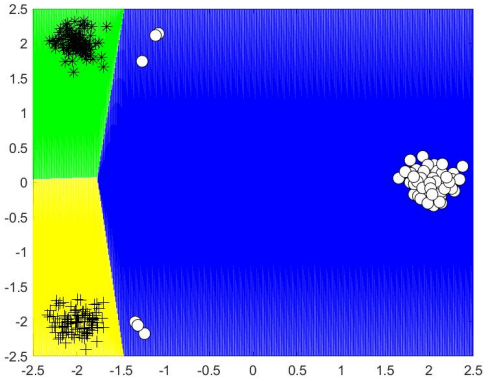
(a) CVaR WW-MSVM, $\nu = 0.15$



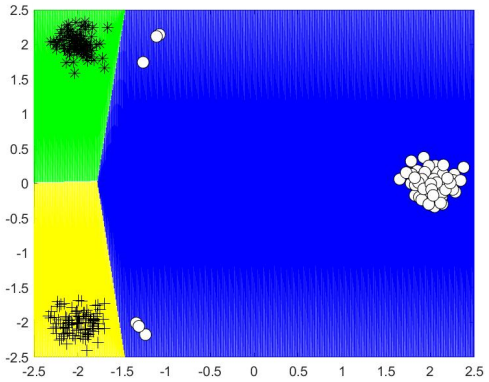
(b) CVaR CS-MSVM, $\nu = 0.15$



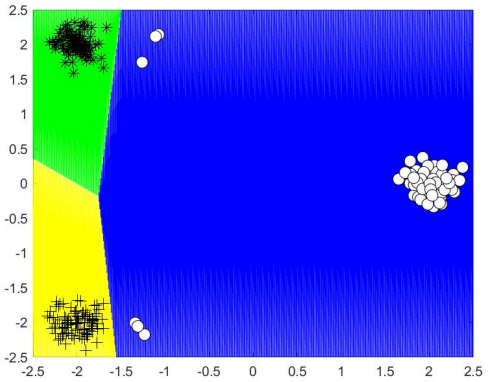
(c) CVaR WW-MSVM, $\nu = 0.1$



(d) CVaR CS-MSVM, $\nu = 0.1$



(e) CVaR WW-MSVM, $\nu = 0.05$

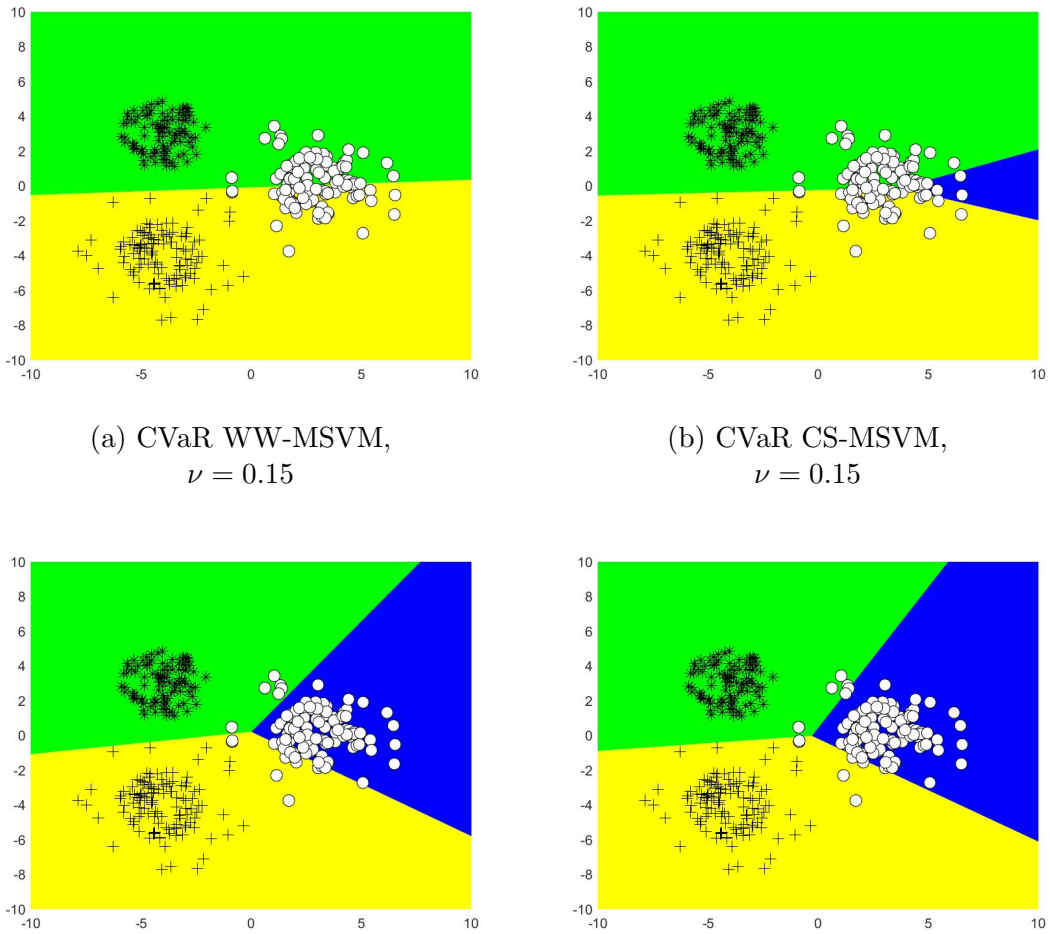


(f) CVaR CS-MSVM, $\nu = 0.05$

Figure 5.11: Comparison of CVaR WW-MSVM and CVaR CS-MSVM with different ν values under 0.88 class 1 probability for Noise 2 dataset.

In Figure 5.11, we examine the influence of ν on overfitting when the probability of class 1 is high. For low level of risk-aversion, that is $\nu = 0.15$, CVaR CS-MSVM results balanced classification regions while it is clear that CVaR WW-SVM is affected by the presence of noise. As level of risk-aversion increases, both CVaR WW-MSVM and CVaR CS-MSVM overfit the dataset, however, unlike CVaR CS-MSVM, we observe overfitting in CVaR WW-MSVM when risk-aversion level is high, i.e, $\nu = 0.05$. As illustrated in Figure 5.9g, CVaR WW-MSVM with $\nu = 0.05$ ignores noisy samples for Noise 1 dataset with high class 1 probability. Therefore, the number of noisy samples has an impact on the performance of CVaR WW-MSVM similar to CVaR CS-MSVM. Yet, CVaR WW-MSVM is less responsive to noise compared to CVaR CS-MSVM.

Case Outlier 2 dataset: This case examines the performance of CVaR WW-MSVM and CVaR CS-MSVM under different ν values for Outlier 2 dataset containing 20 outliers in both class 1 and class 2 when probability of class 1 is both low and high.



(a) CVaR WW-MSVM, $\nu = 0.15$

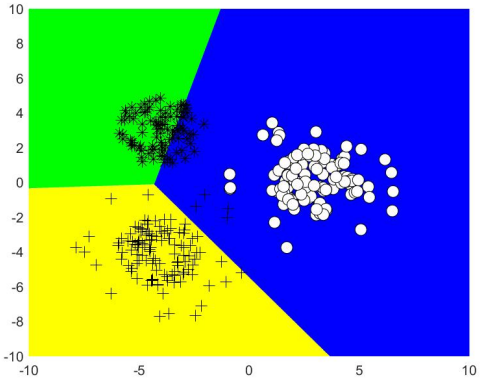
(b) CVaR CS-MSVM, $\nu = 0.15$

(c) CVaR WW-MSVM, $\nu = 0.05$

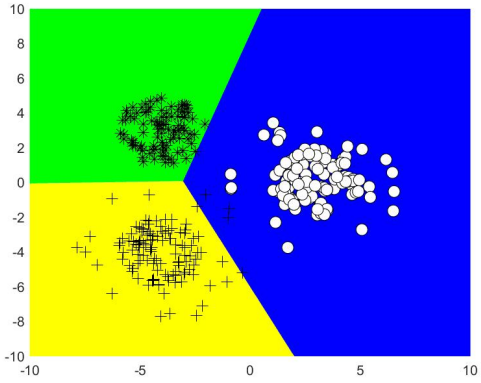
(d) CVaR CS-MSVM, $\nu = 0.05$

Figure 5.12: Comparison of CVaR WW-MSVM and CVaR CS-MSVM with different ν values under 0.01 class 1 probability for Outlier 2 dataset.

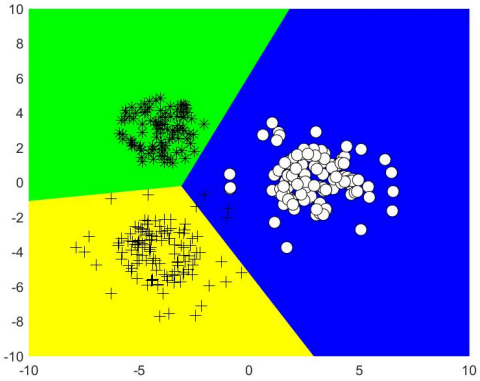
As presented in Figure 5.12, when level of risk aversion is low, i.e ν is 0.15, CVaR WW-MSVM is unable to classify class 1 while CVaR CS-MSVM produces considerably small region for class 1 since outliers of class 2 lead to higher loss values for samples belonging to class 1. As ν decreases, the presence of outliers is less effective on the classification performance of both CVaR WW-MSVM and CVaR CS-MSVM. Also, CVaR CS-MSVM is more stable to outliers.



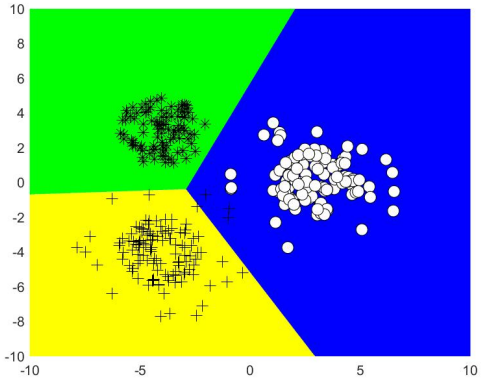
(a) CVaR WW-MSVM, $\nu = 0.15$



(b) CVaR CS-MSVM, $\nu = 0.15$



(c) CVaR WW-MSVM, $\nu = 0.05$



(d) CVaR CS-MSVM, $\nu = 0.05$

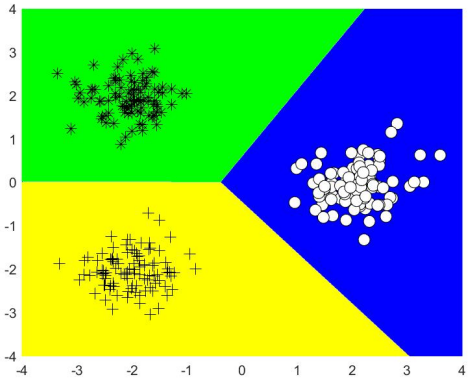
Figure 5.13: Comparison of CVaR WW-MSVM and CVaR CS-MSVM with different ν values under 0.88 class 1 probability for Outlier 2 dataset.

As presented in Figure 5.13, as ν decreases, that is risk-aversion increases, both CVaR WW-MSVM and CVaR CS-MSVM get less responsive to outliers, however, complete separation of the classes is not achieved in both models. Furthermore, CVaR CS-MSVM is more stable to outliers than CVaR WW-MSVM.

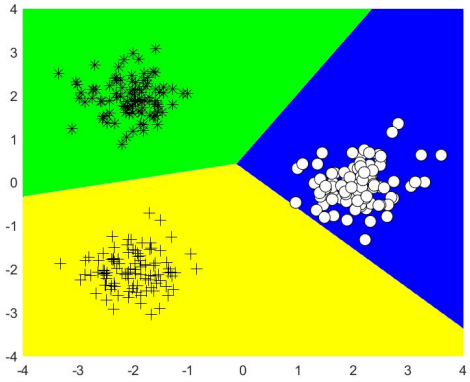
5.2.2.3 Comparison of CVaR WW-MSVM and VaR WW-MSVM

In this section, we compare CVaR WW-MSVM and VaR WW-MSVM in order to analyze their response probability values, noise and outliers.

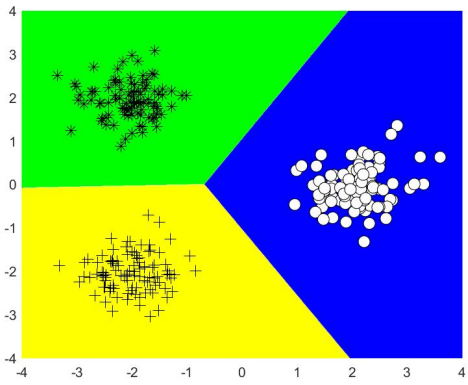
Case Ratio 1 dataset: This case illustrates the effect of class 1 probability on the performances of CVaR WW-MSVM and VaR WW-MSVM when risk-aversion level is high and classes are of equal size.



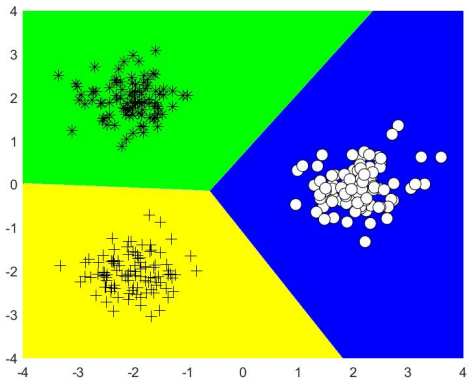
(a) CVaR WW-MSVM Model, 0.05 class 1 probability



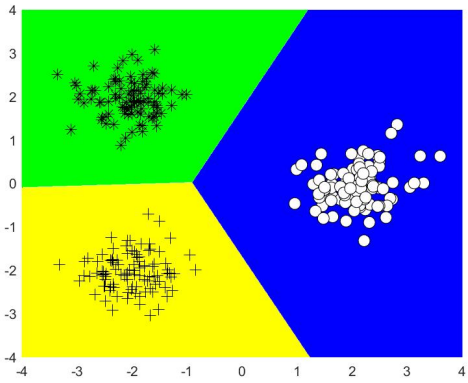
(b) VaR WW-MSVM Model, 0.05 class 1 probability



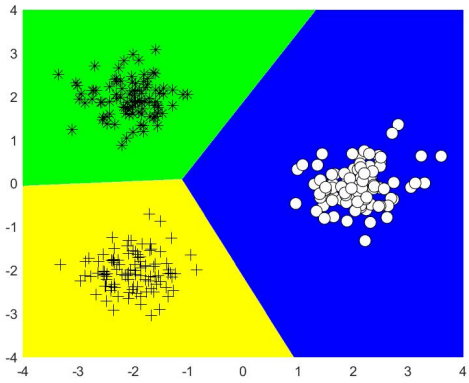
(c) CVaR WW-MSVM Model, 0.33 class 1 probability



(d) VaR WW-MSVM Model, 0.33 class 1 probability



(e) CVaR WW-MSVM Model, 0.88 class 1 probability

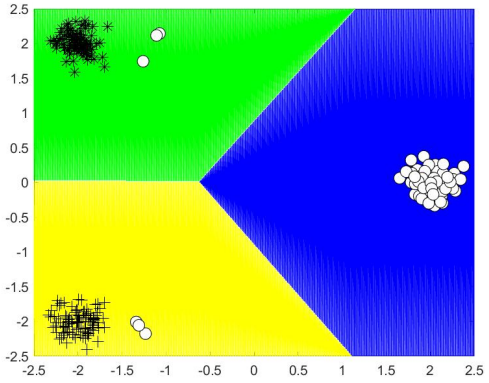


(f) VaR WW-MSVM Model, 0.88 class 1 probability (not optimal)

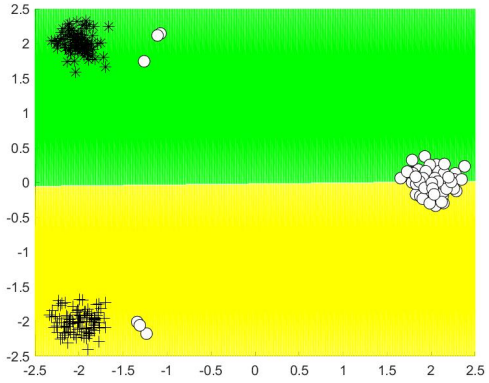
Figure 5.14: Comparison of CVaR WW-MSVM and VaR WW-MSVM with $\nu = 0.05$ under different class 1 probabilities for Ratio 1 dataset.

As seen in Figure 5.14, as the probability of class 1 increases, VaR WW-MSVM produces larger region for class 1 similar to CVaR WW-MSVM. Particularly, when the probability of class 1 is low, VaR WW-MSVM results smaller region for class 1; when the probability of class 1 is high, VaR WW-MSVM gives larger region to class 1 compared to CVaR WW-MSVM. Therefore, VaR WW-MSVM is more sensitive to class probabilities. This conclusion can be extended to the comparison CVaR CS-MSVM and VaR CS-MSVM provided in Figure B.1.

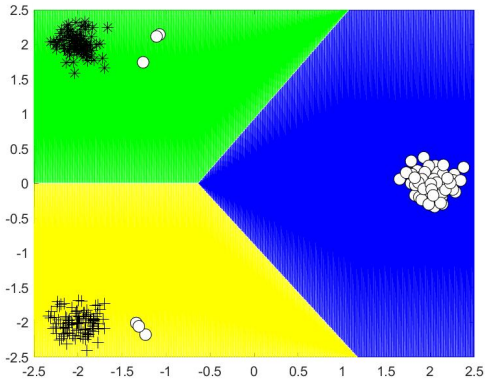
Case Noise 2 dataset: This case demonstrates the effect of ν on the performance of VaR WW-MSVM and presents the comparison of CVaR WW-MSVM and VaR WW-MSVM for Noise 2 dataset for class 1 probabilities of 0.1 and 0.88, respectively.



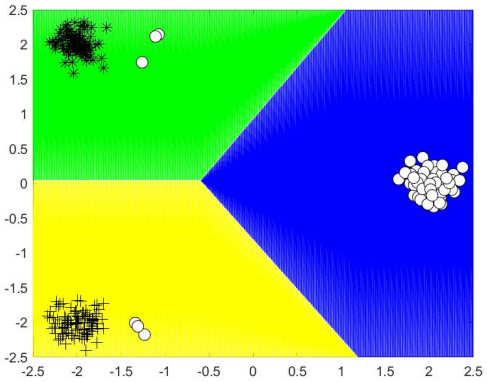
(a) CVaR WW-MSVM, $\nu = 0.1$



(b) VaR WW-MSVM, $\nu = 0.1$



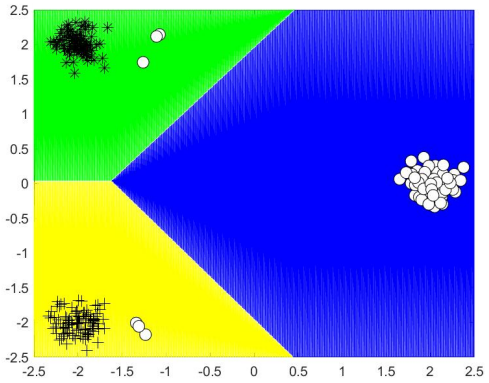
(c) CVaR WW-MSVM, $\nu = 0.05$



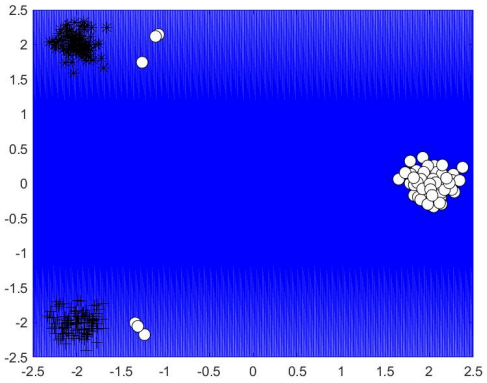
(d) VaR WW-MSVM, $\nu = 0.05$

Figure 5.15: Comparison of CVaR WW-MSVM and VaR WW-MSVM with different ν values under 0.1 class 1 probability for Noise 2 dataset.

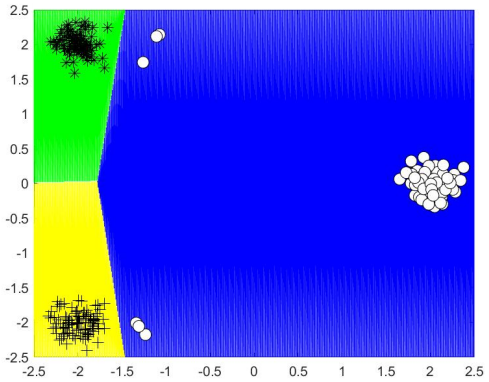
As presented in Figure 5.15, as risk-aversion level increases, the expansion in the region of class 1, that is blue region, is apparent in VaR WW-MSVM while we observe subtle change in CVaR WW-MSVM. Therefore, VaR WW-MSVM is more responsive to the risk-aversion parameter ν when the probability of class 1 is low. This results can be extended to the comparison of CVaR CS-MSVM and VaR CS-MSVM as in Figure B.2.



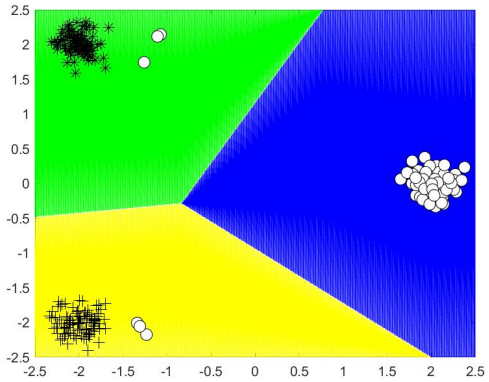
(a) CVaR WW-MSVM, $\nu = 0.1$



(b) VaR WW-MSVM, $\nu = 0.1$



(c) CVaR WW-MSVM, $\nu = 0.05$

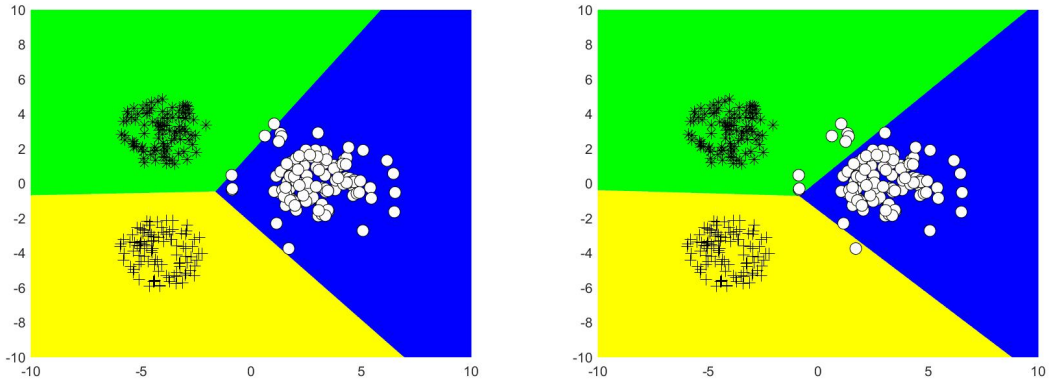


(d) VaR WW-MSVM, $\nu = 0.05$

Figure 5.16: Comparison of CVaR WW-MSVM and VaR WW-MSVM with different ν values under 0.88 class 1 probability for Noise 2 dataset.

As given in Figure 5.16, as ν decreases, CVaR WW-MSVM results in overfitting, while VaR WW-MSVM provides stable results to noise when the probability of class 1 is high. Same conclusion can be made for CVaR CS-MSVM and VaR CS-MSVM as given in Figure B.3.

Case Outlier 3 dataset: This case investigates the performance of CVaR WW-MSVM and VaR WW-MSVM in the presence of outliers when probability of class 1 is low and risk-aversion level, ν, is set to 0.05.



(a) CVaR WW-MSVM, $\nu = 0.05$

(b) VaR WW-MSVM, $\nu = 0.05$

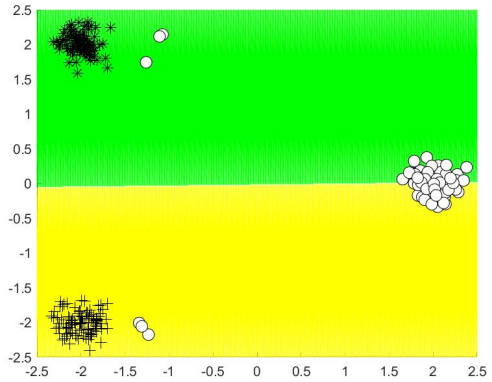
Figure 5.17: Comparison of CVaR WW-MSVM and VaR WW-MSVM with different ν values under 0.05 class 1 probability for Outlier 3 dataset.

Figure 5.17 indicates that VaR WW-MSVM is superior to CVaR WW-MSVM in ignoring outliers. Therefore, VaR WW-MSVM is more stable to outliers for low class 1 probability as ν decreases. This result can be extended to CVaR CS-MSVM and VaR CS-MSVM as provided in Figure B.4.

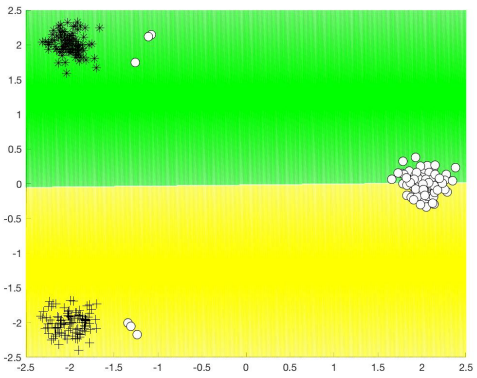
5.2.2.4 Comparison of VaR WW-MSVM and VaR CS-MSVM

In this section, we compare VaR WW-MSVM and VaR CS-MSVM under noise and outliers.

Case Noise 2 dataset: This case analyzes the performances of VaR WW-MSVM and VaR CS-MSVM under different ν values in the presence of noise for low and high class 1 probabilities, respectively.

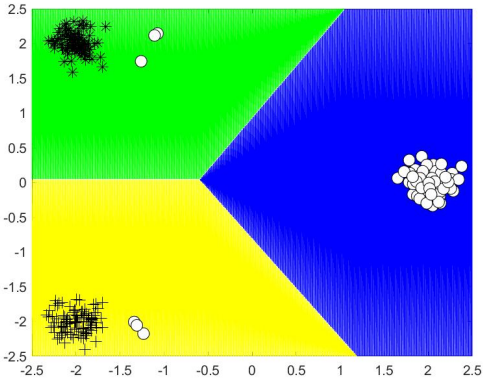


(a) VaR WW-MSVM, $\nu = 0.1$

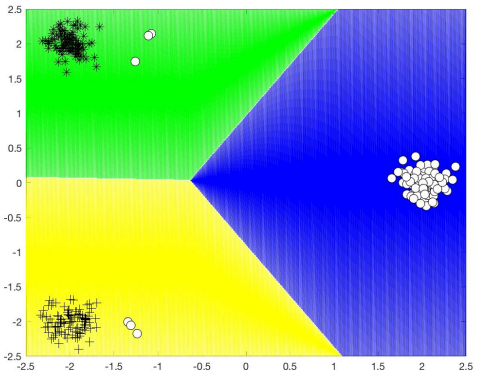


(b) VaR CS-MSVM, $\nu = 0.1$

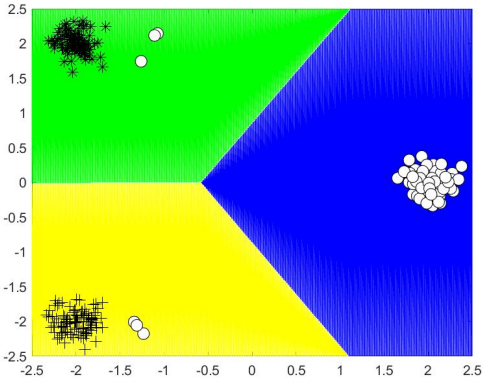
As observed in Figure 5.18, for risk-aversion level of 0.1, both VaR WW-MSVM and VaR CS-MSVM are unable to classify class 1. As risk-aversion level increases, both VaR WW-MSVM and VaR CS-MSVM provides stable results to noise. Therefore, for low class 1 probability, both VaR WW-MSVM and VaR CS-MSVM are stable to noise when level of risk-aversion is high.



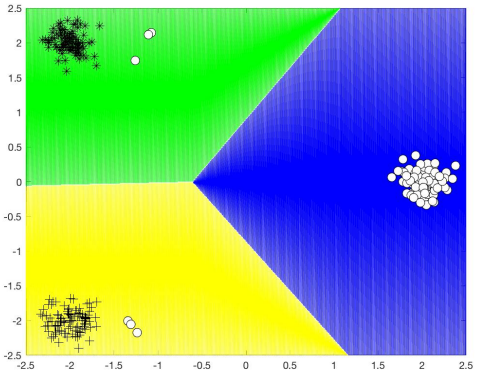
(c) VaR WW-MSVM, $\nu = 0.05$



(d) VaR CS-MSVM, $\nu = 0.05$

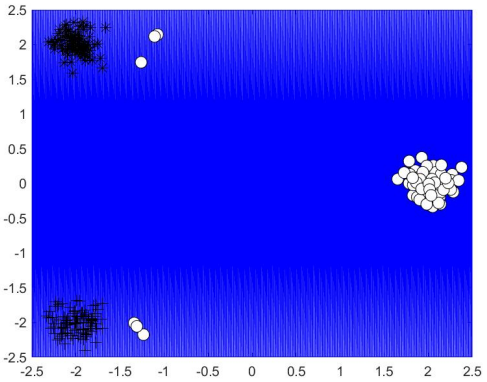


(e) VaR WW-MSVM, $\nu = 0.01$

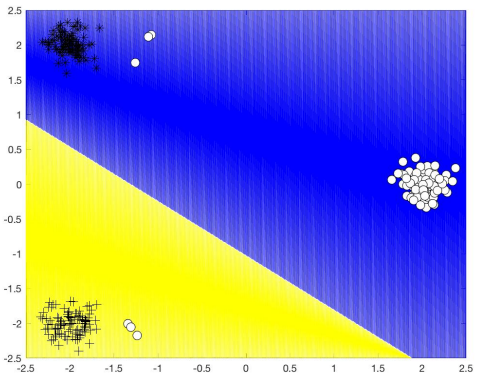


(f) VaR CS-MSVM, $\nu = 0.01$

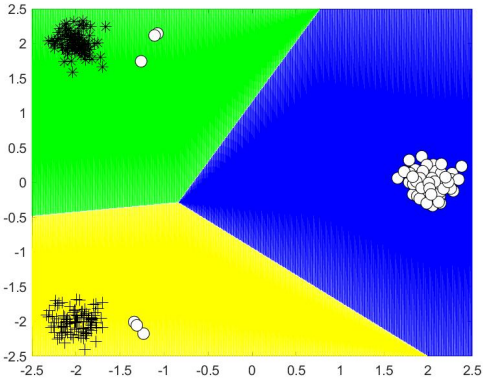
Figure 5.18: Comparison of VaR WW-MSVM and VaR CS-MSVM with different ν values under 0.1 class 1 probability for Noise 2 dataset.



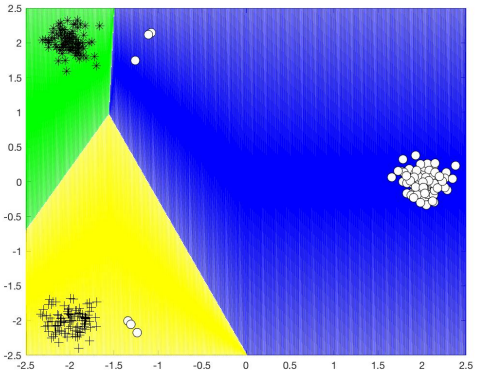
(a) VaR WW-MSVM, $\nu = 0.1$



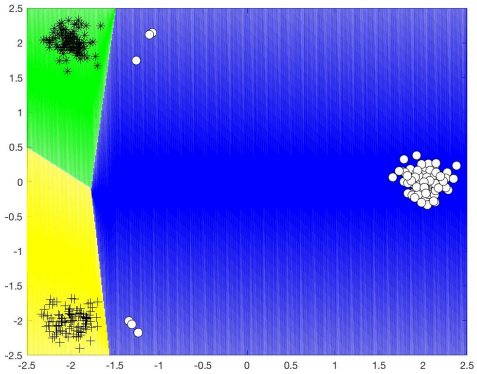
(b) VaR CS-MSVM, $\nu = 0.1$



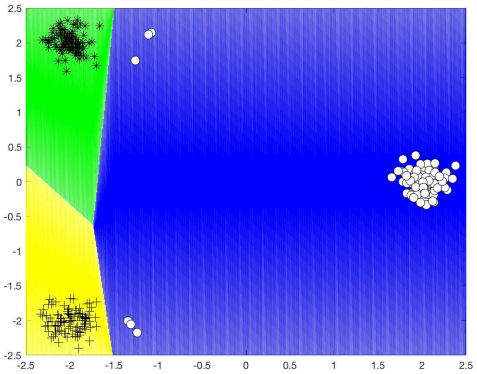
(c) VaR WW-MSVM, $\nu = 0.05$



(d) VaR CS-MSVM, $\nu = 0.05$



(e) VaR WW-MSVM, $\nu = 0.01$

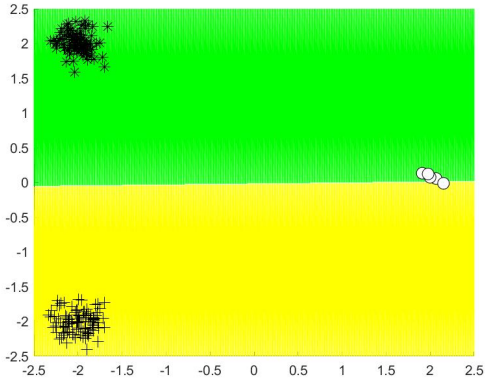


(f) VaR CS-MSVM, $\nu = 0.01$

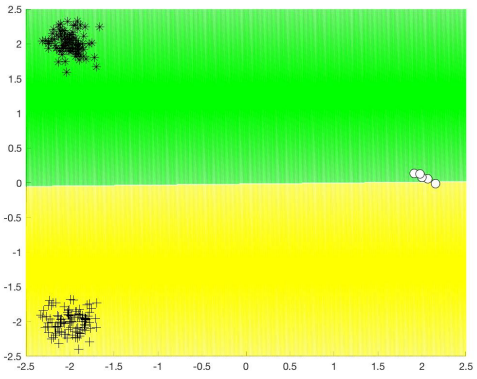
Figure 5.19: Comparison of VaR WW-MSVM and VaR CS-MSVM with different ν values under 0.88 class 1 probability for Noise 2 dataset.

Figure 5.19 illustrates the impact of ν on VaR models for Noise 2 dataset with high class 1 probability. When $\nu = 0.1$, both VaR WW-MSVM and VaR CS-MSVM results poor classification performance. When $\nu = 0.05$, VaR WW-MSVM produces stable results to noise while we observe overfitting in VaR CS-MSVM. However, when risk-aversion level is high, i.e, ν is taken as 0.01, both VaR WW-MSVM and VaR CS-MSVM overfit the dataset. Therefore, low values of ν may lead to overfitting when class with noisy samples have high probability of occurrence. Furthermore, VaR WW-MSVM is more stable to noise.

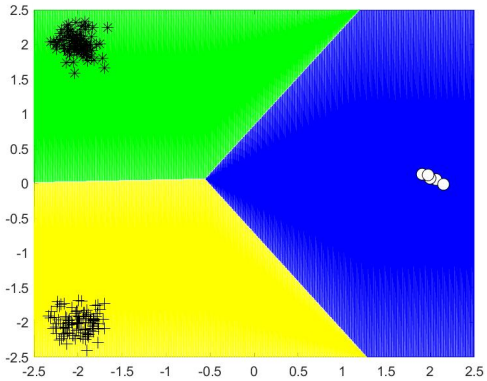
Case Noise 4 dataset: This case investigates the effect of ν on VaR models under 0.1 class probability for Noise 4 datasets where class 1 contains only 5 samples. The aim of this analysis is to observe whether VaR models can separate the comparably small-sized class from the others for different levels of risk-aversion.



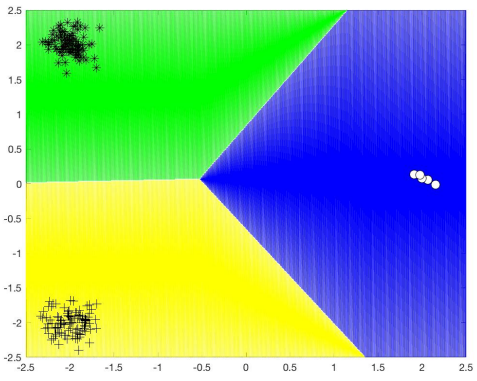
(a) VaR WW-MSVM, $\nu = 0.05$



(b) VaR CS-MSVM, $\nu = 0.05$



(c) VaR WW-MSVM, $\nu = 0.01$

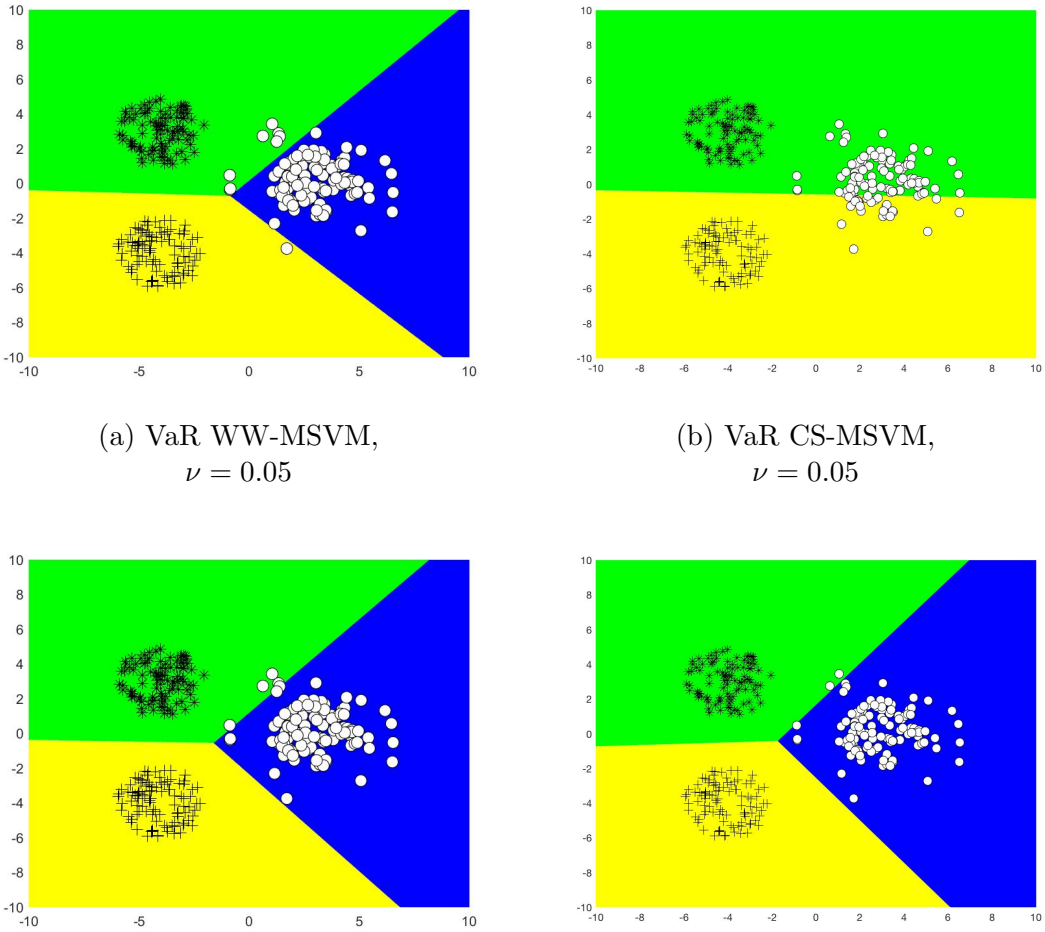


(d) VaR CS-MSVM, $\nu = 0.01$

Figure 5.20: Comparison of VaR WW-MSVM and VaR CS-MSVM with different ν values under 0.05 class 1 probability for Noise 4 dataset.

As observed in Figure 5.20, the separation of small-sized class with low probability depends on the value of ν.

Case Outlier 3 dataset: In this case, impact of ν on the performance of VaR WW-MSVM and VaR CS-MSVM is examined in presence of outliers when the probability of class 1 is low and high, respectively.



(a) VaR WW-MSVM, $\nu = 0.05$

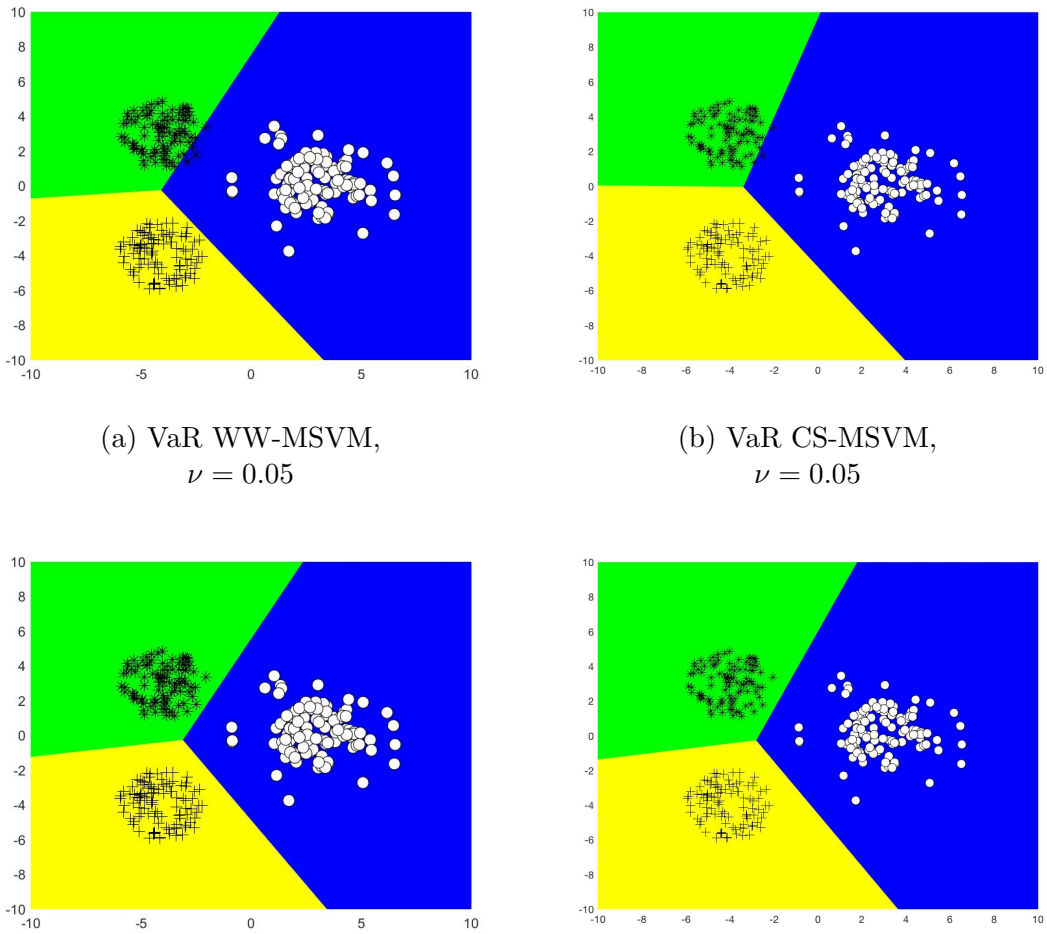
(b) VaR CS-MSVM, $\nu = 0.05$

(c) VaR WW-MSVM, $\nu = 0.01$

(d) VaR CS-MSVM, $\nu = 0.01$

Figure 5.21: Comparison of VaR WW-MSVM and VaR CS-MSVM with different ν values under 0.05 class 1 probability for Outlier 3 dataset.

As given in Figure 5.21, as risk-aversion level increases, the separation of the classes achieved in both models but VaR WW-MSVM ignores more outliers. Therefore, VaR WW-MSVM is superior to VaR CS-MSVM in avoiding outliers for different values of ν when probability of class 1 is low.



(a) VaR WW-MSVM, $\nu = 0.05$

(b) VaR CS-MSVM, $\nu = 0.05$

(c) VaR WW-MSVM, $\nu = 0.01$

(d) VaR CS-MSVM, $\nu = 0.01$

Figure 5.22: Comparison of VaR WW-MSVM and VaR CS-MSVM with different ν values under 0.88 class 1 probability for Outlier 3 dataset.

As given in Figure 5.22, as risk-aversion level increases, that is ν decreases to 0.01, both VaR WW-MSVM and VaR CS-MSVM produce more stable results. However, both models are unable to ignore outliers when probability of class 1 is high.

5.2.2.5 Results

In this section, we present the results obtained from the study on the performance of risk-averse MSVMs. The analysis under different class probabilities, different levels of risk-aversion, and in presence of noise and outliers provides the results below:

- In all models, increase in a class probability results expansion in the region of the corresponding class. However, the expansion detected in CVaR models are more restricted. Therefore, CVaR MSVM is less sensitive to change in class probability compared to original formulations and VaR MSVM.
- For all models, class size is not a dominant parameter in construction of the classification regions.
- For increasing values of probability, CVaR MSVM is more stable to noise compared to the original formulations.
- For high class 1 probability, CVaR MSVM narrows down the region of class 1 as ν decreases because ν denotes the upper bound for the fraction of margin errors. Hence, as ν decreases, CVaR models are forced to narrow down the region of class 1 in order to make less margin error.
- For CVaR MSVM, the risk is defined as misclassifying the samples belonging to the class with lower observation probability.
- For low levels of risk-aversion, CVaR WW-MSVM results poor performance in separating classes compared to CVaR CS-MSVM when probability of a class is significantly higher than the others.
- For high levels of risk-aversion, CVaR CS-MSVM is more responsive to number of noisy samples belonging to class with high probability which may lead to overfitting. However, from this observation it can be concluded that, by using ν and class probability, it is possible to control CVaR MSVM's sensitivity to noise.

- Both CVaR WW-MSVM and CVaR CS-MSVM are unable to ignore outliers regardless of the probability of the class containing outliers.
- Lower values of ν may result overfitting in CVaR MSVMs, while it provides stability to noise in VaR MSVMs when probability of the class containing noise is high.
- For high levels of risk-aversion, VaR MSVM is superior to CVaR MSVM in avoiding outliers when probability of the class containing outliers is low.
- When probability of the class containing noise is high, VaR CS-MSVM results poor performance in avoiding noise independent of the value of ν, while stability to noise is achieved in VaR WW-MSVM for high levels of risk-aversion. Therefore, VaR WW-MSVM is more stable for high probability of the class with noise.

5.2.3 Computational Results for Real-Life Datasets

In this section, we analyze the classification performance of original multi-class SVMs and risk-averse multi-class SVMs on real-life datasets. To present a grounded study on the classification performance of risk-averse multi-class SVMs, we obtain experimental results of CVaR MSVMs, VaR MSVMs and original formulations for the real-life datasets: Breast Tissue, Iris and Wine [50]. The descriptions of the datasets can be found in Table 5.6.

Table 5.6: Descriptions of real-life datasets.

Datasets	# of Samples	# of Attributes	# of Classes	Size of Class 1	Size of Class 2	Size of Class 3	Size of Class 4
Breast Tissue	106	9	4	21	49	14	22
Iris	150	4	3	50	50	50	-
Wine	178	12	3	59	71	48	-

To measure the stability of risk-averse multi-class SVMs on real-life datasets,

we generate different levels of outliers to each dataset. Before adding outliers, we first normalize the datasets by Z-score normalization method, i.e, each attribute has mean 0 and standard deviation 1. Then, we randomly select the fraction of 0%, 1%, 5%, 10% of the dataset and multiply the attributes of these samples by 1000 to generate outliers as described in [13]. 5-fold cross validation is used to evaluate training and testing performances. In this setting, 2/3 of the training dataset is used for training the model while the remaining 1/3 is used for validation, i.e, to find the parameter ν that results in the highest validation accuracy from the set $\{0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15\}$. Table 5.7 illustrates the computational results of WW-MSVM and CS-MSVM for Iris dataset when 0%, 1%, 5% and 10% fraction of the original dataset is used as outliers.

Table 5.7: Computational results for Iris dataset with different outlier levels.

Percentage of Outliers	WW-MSVM		CVaR WW-MSVM		VaR WW-MSVM	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
No Outlier	96.54%	96.67%	98.52%	97.33%	95.06%	95.33%
1%	98.52%	96.00%	98.51%	96.00%	96.04%	93.33%
5%	96.05%	94.00%	95.56%	93.33%	93.09%	94.67%
10%	73.58	72.00%	73.83%	70.00%	90.67%	88.72%
	CS-MSVM		CVaR CS-MSVM		VaR CS-MSVM	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
No Outlier	96.54%	98.27%	97.43%	97.33%	95.06%	96.00%
1%	94.82%	95.33%	97.53%	95.33%	96.54%	96.67%
5%	96.04%	94.67%	94.82%	94.67%	95.80%	96.67%
10%	74.32%	73.33%	73.82%	70.00%	93.33%	94.00%

The computational results indicate that all models give similar accuracy rates when outlier levels are 0%, 1%, 5%. With the increase in outlier percentage, test accuracy decreases both in original formulations and CVaR MSVMs. On the other hand, VaR MSVMs are less responsive to the increase of the outlier percentage.

Particularly, when percentage of the outliers is 10%, VaR MSVMs result slight decrease in accuracy while the performance of the other models substantially degrades. When WW-MSVM and CS-MSVM are compared, original models yield similar accuracy rates. as in CVaR WW-MSVM and CVaR CS-MSVM. However, VaR CS-MSVM provides more stable results in presence of outliers for Iris dataset compared to VaR WW-MSVM.

Table 5.8: Computational results for Iris dataset when only one class has outliers.

Percentage of Outliers	WW-MSVM		CVaR WW-MSVM		VaR WW-MSVM	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
No Outlier	96.54%	96.67%	98.52%	97.33%	95.06%	95.33%
1%	82.72%	81.33%	82.47%	80.00%	95.80%	94.67%
5%	79.01%	77.33%	77.78%	77.33%	91.39%	89.67%
10%	79.51%	74.67%	76.04%	75.33%	90.12%	85.72%
	CS-MSVM		CVaR CS-MSVM		VaR CS-MSVM	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
No Outlier	96.54%	98.27%	97.43%	97.33%	95.06%	96.00%
1%	82.72%	81.33%	82.72%	80.00%	95.30%	96.00%
5%	79.51%	78.00%	76.05%	75.33%	91.35%	88.00%
10%	78.03%	76.00%	75.56%	73.33%	89.38%	84.67%

Table 5.8 illustrates the performances of the models when only one class has (here, class 2) the outliers. It is aimed to analyze the impact of the outlier distribution to over-all performance. As results present, the accuracy of original formulations and CVaR MSVM significantly decreases with the increase in the percentage of outliers. Similar to the other models, the performance of VaR MSVM degrades as well. But the decrease rate is much smaller and VaR MSVM results considerably higher test accuracy rates compared to the other models. When performance of WW-MSVM and CS-MSVM is compared, it is noticed that both models give similar results.

Table 5.9: Computational results for Breast Tissue dataset with different outlier levels.

Percentage of Outliers	WW-MSVM		CVaR WW-MSVM		VaR WW-MSVM	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
No Outlier	95.16%	89.65%	95.50%	88.70%	92.38%	85.93%
1%	94.47%	88.70%	94.46%	89.61%	91.35%	89.61%
5%	93.78%	87.75%	93.78%	88.66%	91.69%	89.96%
10%	86.16%	86.88%	97.93%	87.75%	90.37%	88.06%
	CS-MSVM		CVaR CS-MSVM		VaR CS-MSVM	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
No Outlier	96.20%	89.61%	95.85%	85.93%	95.16%	84.89%
1%	91.69%	88.66%	100%	85.02%	93.78%	89.65%
5%	93.42%	89.61%	96.69%	85.89%	92.38%	90.56%
10%	88.59%	87.83%	97.93%	87.75%	92.74%	89.42%

Table 5.9 presents the performances of the models for Breast Tissue dataset under different outlier levels. As observed from the results, the classification performance of all models are almost insensitive to the percentage of outlier. All models provide similar accuracy rates as percentage of outliers increases. Particularly, when outliers constitute 10% of the dataset, VaR MSVM provides marginally higher test accuracy rates compared to other models.

Table 5.10: Computational results for Wine dataset with different outlier levels.

Percentage of Outliers	WW-MSVM		CVaR WW-MSVM		VaR WW-MSVM	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
No Outlier	97.91%	96.54%	100%	97.71%	99.17%	97.09%
1%	99.59%	96.54%	100%	96.05%	98.34%	97.22%
5%	98.13%	96.01%	99.79%	96.63%	98.13%	97.19%
10%	98.96%	94.11%	100%	96.30%	98.33%	97.16%
	CS-MSVM		CVaR CS-MSVM		VaR CS-MSVM	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
No Outlier	98.13%	98.23%	100%	97.71%	100%	97.71%
1%	99.38%	97.75%	100%	97.22%	99.17%	97.19%
5%	100%	97.19%	100%	96.78%	98.13%	97.22%
10%	99.79%	97.16%	100%	96.56%	99.59%	97.16%

Table 5.10 demonstrates the performances of all models for Wine dataset under different outlier levels. The results indicate that all models are almost indifferent to the percentage of outliers. Therefore, the outlier generation scheme may be ineffective for this dataset.

Table 5.11: Computational results for Wine dataset when only the majority class has outliers.

Percentage of Outlier	WW-MSVM		CVaR WW-MSVM		VaR WW-MSVM	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
No Outlier	97.91%	96.54%	100%	97.71%	99.17%	97.09%
1%	100%	98.89%	99.59%	96.67%	98.34%	96.65%
5%	98.75%	95.49%	99.59%	96.05%	97.09%	97.19%
10%	98.34%	94.97%	99.17%	94.93%	95.88%	96.08%
	CS-MSVM		CVaR CS-MSVM		VaR CS-MSVM	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
No Outlier	98.13%	98.23%	100%	97.71%	100%	97.71%
1%	99.37%	97.09%	100%	97.22%	99.17%	97.22%
5%	98.12%	96.36%	100%	94.38%	99.52%	97.22%
10%	97.91%	95.65%	100%	95.56%	99.38%	98.31%

Table 5.11 presents the results for Wine dataset when outliers are observed only in the majority class. The aim is to investigate the impact of outliers in the majority class on the classification performance. Similar to Breast Tissue dataset, the classification performance of all models slightly degrades when percentage of outliers increases. The results illustrates that while CVaR MSVM and original formulations yields similar test accuracy rates, VaR MSVM provides marginally higher rates for increasing numbers of outliers in majority class.

Table 5.12: Computational results for Wine dataset when only the minority class has outliers.

Percentage of Outlier	WW-MSVM		CVaR WW-MSVM		VaR WW-MSVM	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
No Outlier	97.91%	96.54%	100%	97.71%	99.17%	97.09%
1%	98.75%	96.67%	99.79%	97.78%	98.76%	98.33%
5%	99.59%	96.67%	99.79%	97.75%	98.13%	98.89%
10%	97.83%	94.38%	99.38%	97.19%	98.27%	97.75%
	CS-MSVM		CVaR CS-MSVM		VaR CS-MSVM	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
No Outlier	98.13%	98.23%	100%	97.71%	100%	97.71%
1%	99.59%	97.22%	100%	97.22%	98.55%	96.63%
5%	99.59%	97.75%	100%	97.75%	100%	97.77%
10%	98.96%	97.19%	100%	97.18%	98.75%	96.60%

Table 5.12 shows the results for Wine dataset when outliers are observed only in the minority class. Similar to the results for Wine dataset, all models are almost indifferent to the percentage of outliers.

Chapter 6

Conclusion

In this study, we develop risk-averse multi-class SVM by using financial risk measures CVaR and VaR. The aim of this study is to construct a robust classifier for multi-class problems such that the impact of outliers and noise is minimized. For this purpose, we consider Weston and Watkins MSVM and Crammer and Singer MSVM models which follow the *all-together* scheme. Following the results that risk-aversion contributes to the stability to outliers and noise (see [12] and [13]), we implement the financial risk measures CVaR and VaR to CS-MSVM and WW-MSVM. Due to mathematical properties of CVaR and quadratic nature of SVM, both CVaR WW-MSVM and CVaR CS-MSVM are formulated as quadratic programming. Unlike CVaR, VaR is not a convex function which leads that VaR MSVM is computationally intractable. In order to evaluate the performance of VaR MSVM, we propose a strong big-M formulation and compare it with branch and cut decomposition algorithm [20] and regular big-M formulation. For the computational analysis, we generate two groups of artificial datasets: first group is used to assess the computational performance of the solution methods for VaR MSVM in terms of optimality gap and objective value; the second group is used to analyze the behavior of CVaR and VaR under the presence of noise, outliers and imbalanced class distributions for different values of class probabilities and risk-aversion levels.

The computational results on artificial datasets indicate that the proposed strong

big-M formulation outperforms the other methods in terms of solution time and optimality gap. Particularly, VaR CS-MSVM results shorter computation time compared to VaR WW-MSVM. In addition, we provide a comparative study to analyze the performance of risk-averse MSVM under different class probabilities and risk-aversion levels where existence of outliers, noise and imbalance class distribution are issue. The outcomes of this analysis show that imbalanced class distributions have unsubstantial effect on the performance of all models. In addition, we observe that class probability and risk-aversion level are dominant parameters affecting the behavior of CVaR MSVMs and VaR MSVMs such that they determine the priority of the models. The higher the value of a class probability, the wider the region of the corresponding class. In this analysis, it is observed that risk-aversion level has a restrictive role by controlling the fraction of margin errors. Also, results under the presence of noise and outliers present that CVaR and VaR provide more stable classification regions comparet to the original formulations. However, for high levels of risk-aversion, CVaR MSVMs may overfit the datasets while VaR MSVMs remain stable to outliers and noise. Therefore, VaR MSVMs are less sensitive to outliers and noise in datasets compared to CVaR MSVMs and original formulations.

In order to further validate our results, we implement all models using real-life datasets. To see the impact of outliers on the real-life examples, we artificially generate outliers using the original datasets. For Iris dataset, the presence of outliers considerably influences the out of sample performance of CVaR MSVMs and original formulations. The accuracy rates in these models significantly degrade when percentage of outliers increases, while VaR MSVMs provide more stable results. Moreover, when outliers are observed in only one class of Iris dataset, the accuracy rates in CVaR MSVMs and original formulation decrease substantially, whereas, VaR MSVMs result considerably higher accuracy rates. For Breast Tissue and Wine datasets, we observe slight differences among the models meaning that our outlier generation scheme may be ineffective for these datasets or they may be well-separable. Therefore, depending on the dataset and outlier distribution, VaR MSVMs are more robust to outliers and noise in datasets.

This study provides implementation of risk-aversion to MSVM by using widely

known financial risk measures CVaR and VaR. With the risk defined as misclassification of samples, we construct robust risk-averse classifiers for multi-class problems which can be applied to various fields such as medicine, banking and marketing.

Bibliography

[1] "Introduction to support vector machines." `https://goo.gl/QGiH8R`. Accessed: 2018-11-28.

[2] A. K. Jain, R. P. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 2006.

[4] M. Hussain, S. K. Wajid, A. Elzaart, and M. Berbar, "A comparison of svm kernel functions for breast cancer detection," in *8th International Conference Computer Graphics, Imaging and Visualization*, pp. 145–150, IEEE, 2011.

[5] G. Singh, R. Gupta, A. Rastogi, M. D. Chandel, and A. Riyaz, "A machine learning approach for detection of fraud based on svm," *International Journal of Scientific Engineering and Technology (ISSN: 2277-1581), Volume*, no. 1, pp. 194–198, 2012.

[6] H. C. Kim, S. Pang, H. M. Je, D. Kim, and S. Y. Bang, "Constructing support vector machine ensemble," *Pattern recognition*, vol. 36, no. 12, pp. 2757–2767, 2003.

[7] D. Sculley and G. M. Wachman, "Relaxed online svms for spam filtering," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 415–422, ACM, 2007.

[8] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics*, vol. 11, no. 1, pp. 1–21, 1969.

[9] G. S. Maddala and K. Lahiri, *Introduction to econometrics*, vol. 2. Macmillan New York, 1992.

[10] C. M. Salgado, C. Azevedo, H. Proença, and S. M. Vieira, "Noise versus outliers," in *Secondary Analysis of Electronic Health Records*, pp. 163–183, Springer, 2016.

[11] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, ACM, 1992.

[12] A. Takeda and M. Sugiyama, "v-support vector machine as conditional value-at-risk minimization," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 1056–1063, ACM, 2008.

[13] P. Tsyurmasto, M. Zabarankin, and S. Uryasev, "Value-at-risk support vector machine: Stability to outliers," *Journal of Combinatorial Optimization*, vol. 28, no. 1, pp. 218–232, 2014.

[14] X. Zhang, "Using class-center vectors to build support vector machines," in *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, pp. 3–11, IEEE, 1999.

[15] Q. Song, W. Hu, and W. Xie, "Robust support vector machine with bullet hole image classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 32, no. 4, pp. 440–448, 2002.

[16] C. Lin and S. Wang, "Fuzzy support vector machines," *IEEE transactions on neural networks*, vol. 13, no. 2, pp. 464–471, 2002.

[17] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *Journal of Risk*, vol. 2, pp. 21–41, 2000.

[18] Y. A. Toukourou and F. Dufresne, "On integrated chance constraints in alm for pension funds," *ASTIN Bulletin: The Journal of the IAA*, vol. 48, no. 2, pp. 571–609, 2018.

[19] W. M. Raike, "Dissection methods for solutions in chance constrained programming problems under discrete distributions," *Management science*, vol. 16, no. 11, pp. 708–715, 1970.

[20] J. Luedtke, "A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support," *Mathematical Programming*, vol. 146, no. 1-2, pp. 219–244, 2014.

[21] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," pp. 11–30, 2015.

[22] V. Vapnik, *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.

[23] V. N. Vapnik and A. Y. Chervonenkis, "The necessary and sufficient conditions for consistency in the empirical risk minimization method," *Pattern Recognition and Image Analysis*, vol. 1, pp. 283–305, 1989.

[24] I. Guyon, B. Boser, and V. Vapnik, "Automatic capacity tuning of very large vc-dimension classifiers," in *Advances in neural information processing systems*, pp. 147–155, 1993.

[25] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[26] A. Ng, "Stanford University, CS229 Lecture Notes in Support Vector Machines." `http://cs229.stanford.edu/notes/cs229-notes3.pdf`. Accessed: 2018-11-28.

[27] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

[28] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural computation*, vol. 12, no. 5, pp. 1207–1245, 2000.

[29] D. J. Crisp and C. J. Burges, "A geometric interpretation of v-svm classifiers," in *Advances in neural information processing systems*, pp. 244–250, 2000.

[30] U. Kressel, "Pairwise classification and support vector machines. advances in kernel methods: Suppoert vector learning," 1999.

[31] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.

[32] J. Weston and C. Watkins, "Multi-class support vector machines," tech. rep., Citeseer, 1998.

[33] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of machine learning research*, vol. 2, no. Dec, pp. 265–292, 2001.

[34] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[35] P. Jorion, "Value at risk: The new benchmark for controlling market risk," 1996.

[36] A. A. Gaivoronski and G. Pflug, "Value-at-risk in portfolio optimization: properties and computational approach," *Journal of risk*, vol. 7, no. 2, pp. 1–31, 2005.

[37] P. Artzner, F. Delbaen, E. Jean-Marc, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, pp. 203 – 228, 07 1999.

[38] P. Artzner, F. Delbaen, E. Jean-Marc, and D. Heath, "Thinking coherently," *Risk*, vol. 10, pp. 68–71, 1997.

[39] R. T. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *Journal of banking & finance*, vol. 26, no. 7, pp. 1443–1471, 2002.

[40] P. Embrechts, S. I. Resnick, and G. Samorodnitsky, "Extreme value theory as a risk management tool," *North American Actuarial Journal*, vol. 3, no. 2, pp. 30–41, 1999.

[41] G. C. Pflug, "Some remarks on the value-at-risk and the conditional value-at-risk," in *Probabilistic constrained optimization*, pp. 272–281, Springer, 2000.

[42] J. Gotoh and A. Takeda, *A linear classification model based on conditional geometric score*. Inst. of Technology, 2004.

[43] G. P. McCormick, "Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems," *Mathematical programming*, vol. 10, no. 1, pp. 147–175, 1976.

[44] F. Qiu, S. Ahmed, S. S. Dey, and L. A. Wolsey, "Covering linear programming with violations," *INFORMS Journal on Computing*, vol. 26, no. 3, pp. 531–546, 2014.

[45] Y. Song, J. R. Luedtke, and S. Küçükyavuz, "Chance-constrained binary packing problems," *INFORMS Journal on Computing*, vol. 26, no. 4, pp. 735–747, 2014.

[46] X. Bai, J. Sun, X. Sun, and X. Zheng, "An alternating direction method for chance-constrained optimization problems with discrete distributions," *Technical report, School of Management, Fudan University*, 2012.

[47] J. Luedtke, S. Ahmed, and G. L. Nemhauser, "An integer programming approach for linear programs with probabilistic constraints," *Mathematical programming*, vol. 122, no. 2, pp. 247–272, 2010.

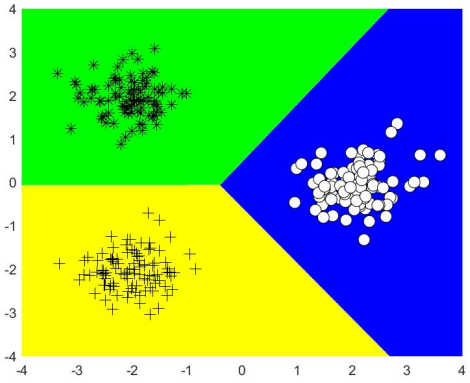
[48] O. Günlük and Y. Pochet, "Mixing mixed-integer inequalities," *Mathematical Programming*, vol. 90, no. 3, pp. 429–457, 2001.

[49] A. Atamtürk, G. L. Nemhauser, and M. W. Savelsbergh, "The mixed vertex packing problem," *Mathematical Programming*, vol. 89, no. 1, pp. 35–53, 2000.

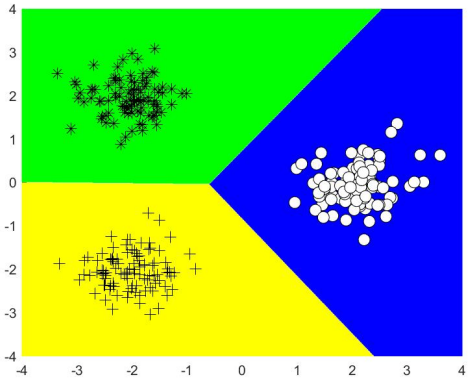
[50] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017.

Appendix A

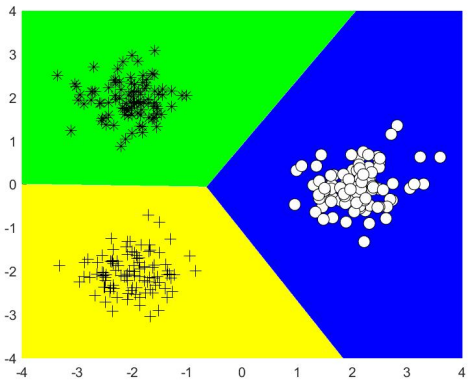
Comparison of CS-MSVM and CVaR CS-MSVM



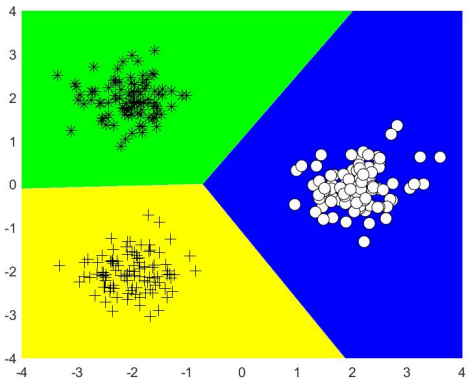
(a) CS-MSVM,
0.1 class 1 probability



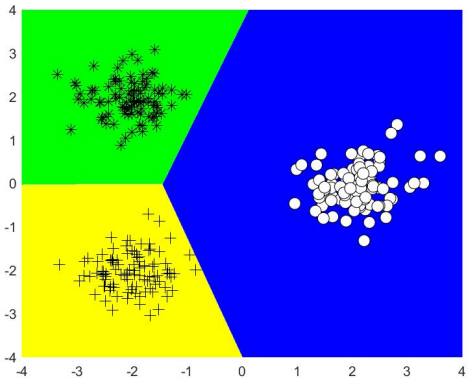
(b) CVaR CS-MSVM,
0.1 class 1 probability



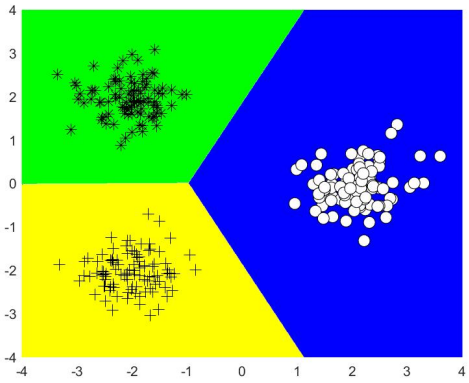
(c) CS-MSVM,
0.33 class 1 probability



(d) CVaR CS-MSVM,
0.33 class 1 probability

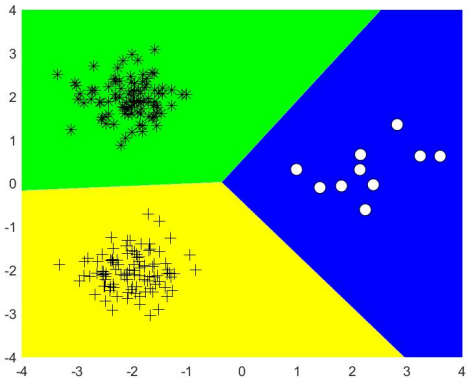


(e) CS-MSVM,
0.88 class 1 probability

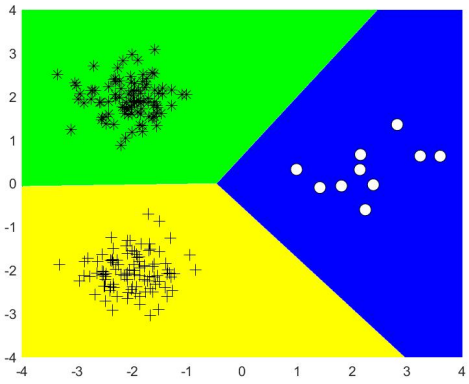


(f) CVaR CS-MSVM,
0.88 class 1 probability

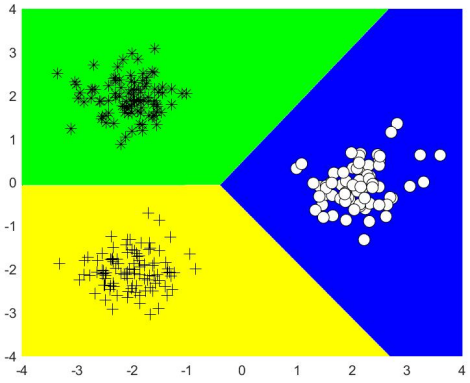
Figure A.1: Comparison of CS-MSVM and CVaR CS-MSVM with $\nu = 0.1$ under different class 1 probabilities for Ratio 1 dataset.



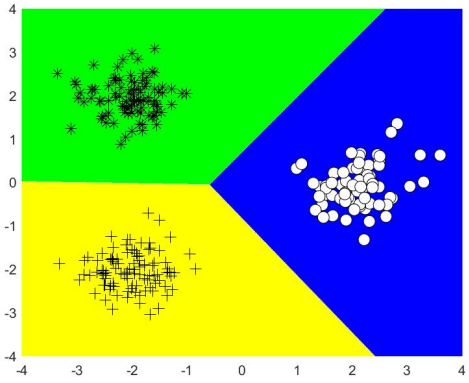
(a) CS-MSVM,
Ratio 6 dataset



(b) CVaR CS-MSVM,
Ratio 6 dataset

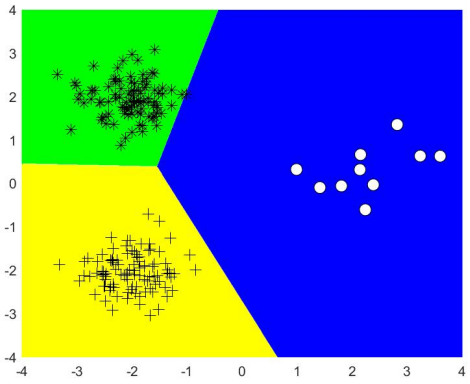


(c) CS-MSVM,
Ratio 2 dataset

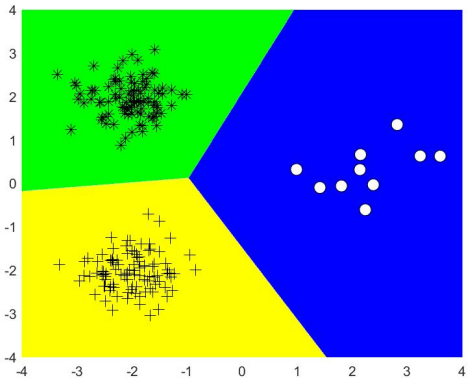


(d) CVaR CS-MSVM,
Ratio 2 dataset

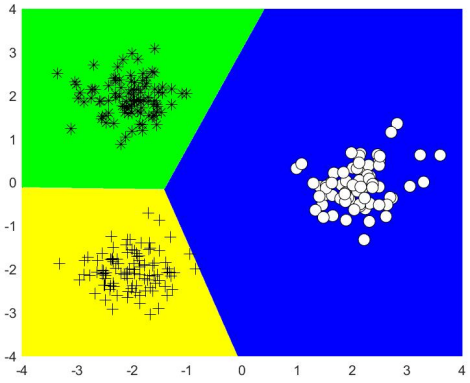
Figure A.2: Comparison of CS-MSVM and CVaR CS-MSVM with $\nu = 0.1$ under 0.1 class 1 probability for Ratio 2 and 6 datasets.



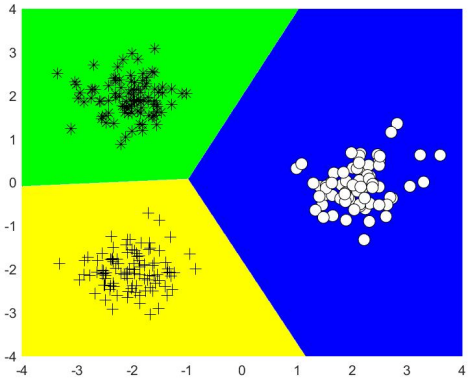
(a) CS-MSVM,
Ratio 6 dataset



(b) CVaR CS-MSVM,
Ratio 6 dataset

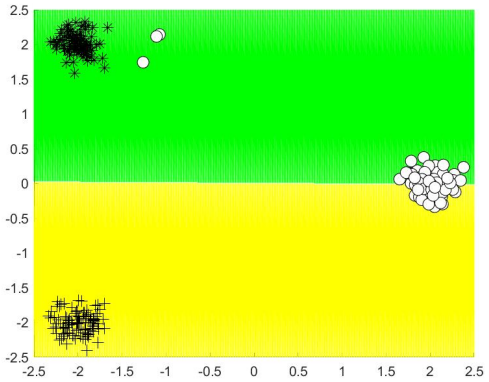


(c) CS-MSVM,
Ratio 2 dataset

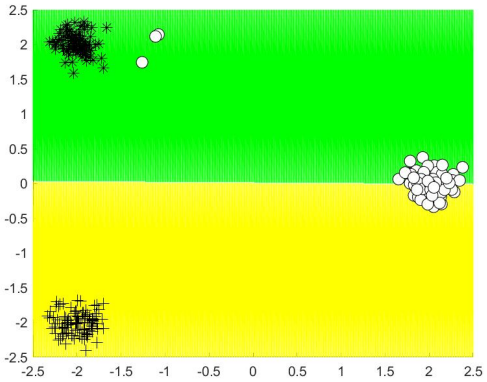


(d) CVaR CS-MSVM,
Ratio 2 dataset

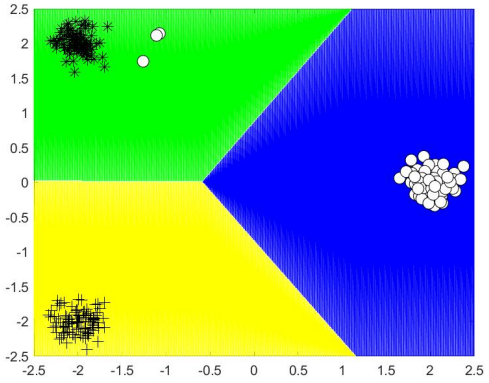
Figure A.3: Comparison of CS-MSVM and CVaR CS-MSVM with different ν values under 0.88 class 1 probability for Ratio 2 and 6 datasets.



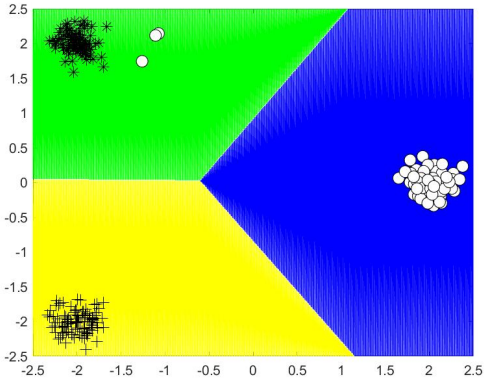
(a) CS-MSVM,
0.01 class 1 probability



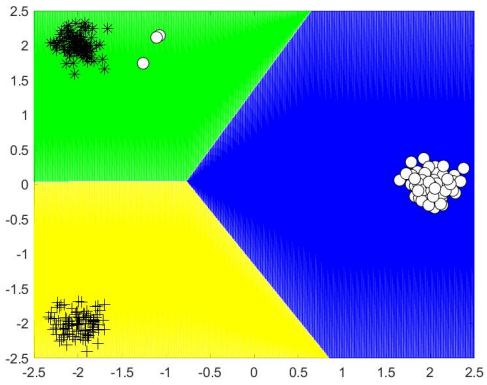
(b) CVaR CS-MSVM,
0.01 class 1 probability



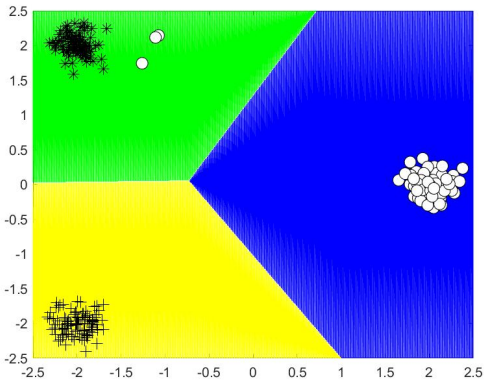
(c) CS-MSVM,
0.1 class 1 probability



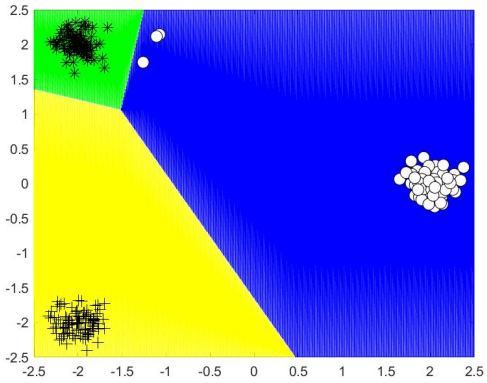
(d) CVaR CS-MSVM,
0.1 class 1 probability



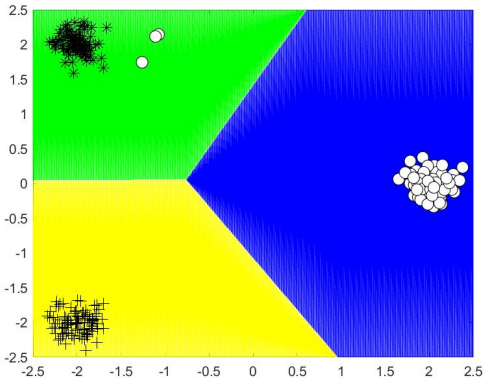
(e) CS-MSVM,
0.77 class 1 probability



(f) CVaR CS-MSVM,
0.77 class 1 probability



(g) CS-MSVM,
0.88 class 1 probability

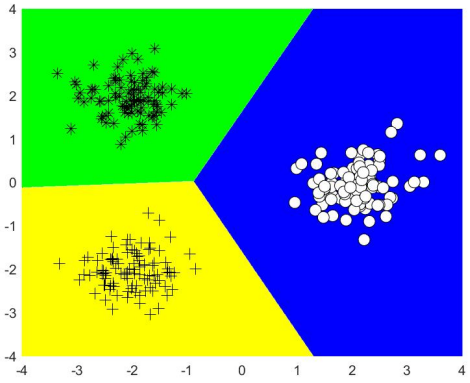


(h) CVaR CS-MSVM,
0.88 class 1 probability

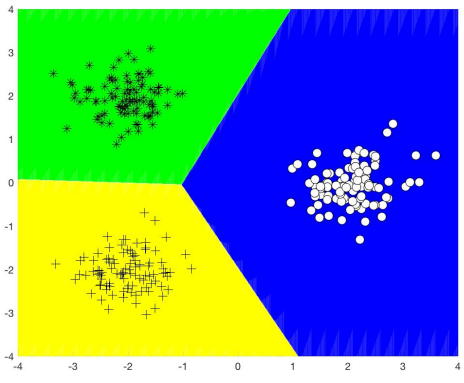
Figure A.4: Comparison of CS-MSVM and CVaR CS-MSVM with $\nu = 0.1$ under different class 1 probabilities for Noise 1 dataset.

Appendix B

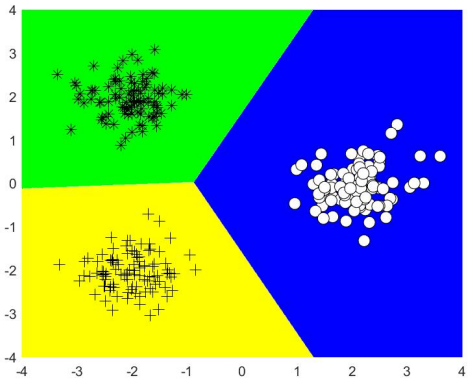
Comparison of CVaR CS-MSVM and VaR CS-MSVM



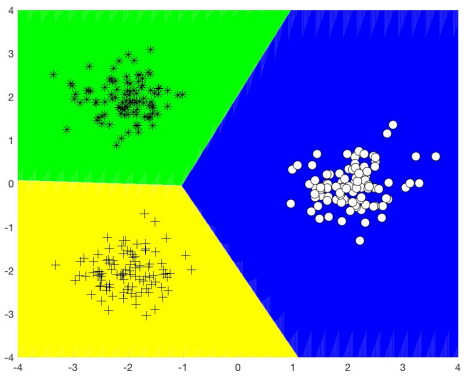
(a) CVaR CS-MSVM,
0.10 class 1 probability



(b) VaR CS-MSVM,
0.10 class 1 probability

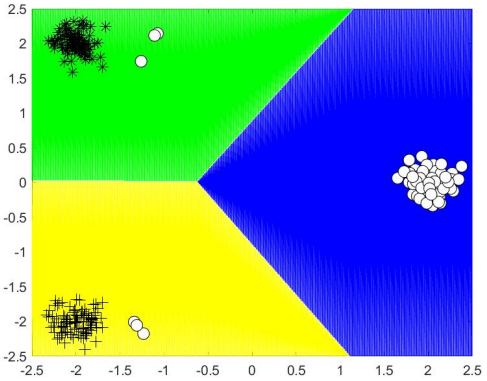


(c) CVaR CS-MSVM,
0.88 class 1 probability

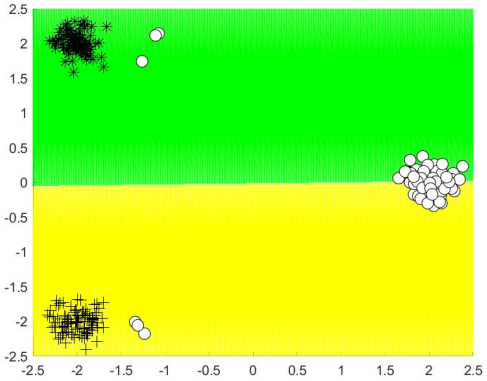


(d) VaR CS-MSVM,
0.88 class 1 probability

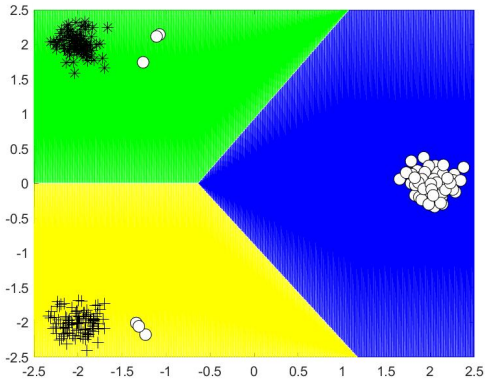
Figure B.1: Comparison of CVaR CS-MSVM and VaR CS-MSVM with $\nu = 0.05$ under different class 1 probabilities for Ratio 1 dataset.



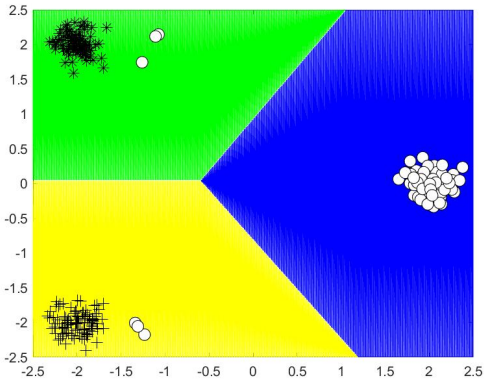
(a) CVaR WW-MSVM,
$\nu = 0.1$



(b) VaR WW-MSVM,
$\nu = 0.1$

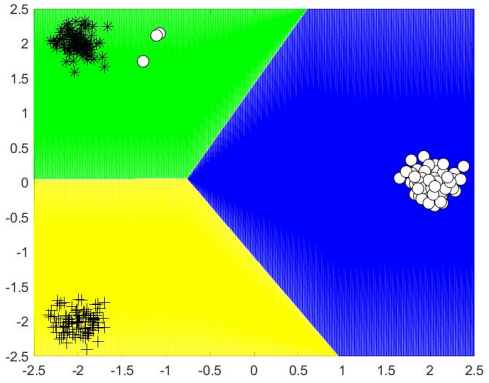


(c) CVaR WW-MSVM, $\nu = 0.05$

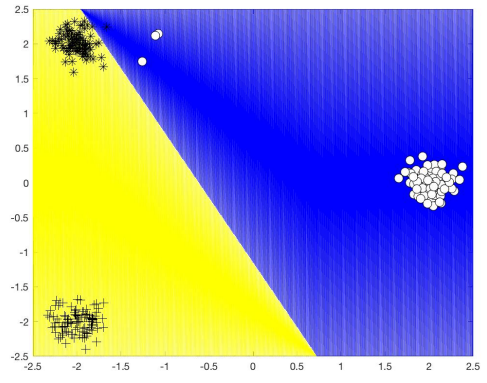


(d) VaR WW-MSVM, $\nu = 0.05$

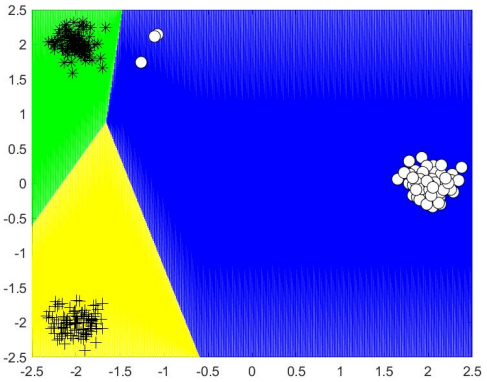
Figure B.2: Comparison of CVaR WW-MSVM and VaR WW-MSVM with different ν values under 0.1 class 1 probability for Noise 2 dataset.



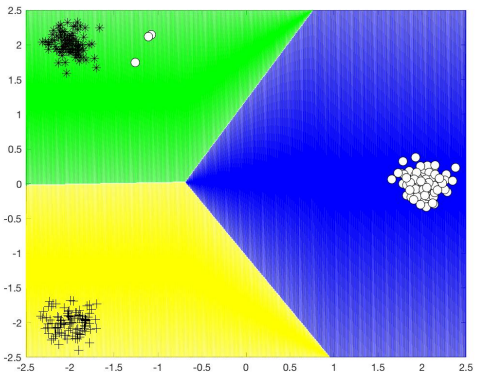
(a) CVaR CS-MSVM, $\nu = 0.1$



(b) VaR CS-MSVM, $\nu = 0.1$

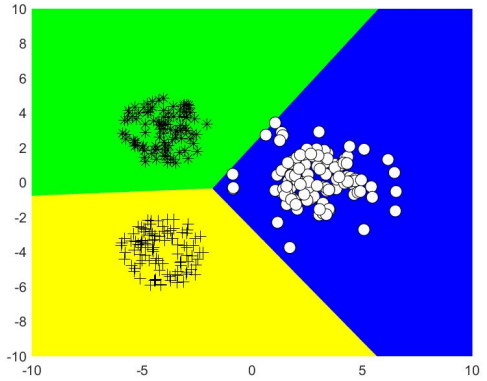


(c) CVaR CS-MSVM, $\nu = 0.05$

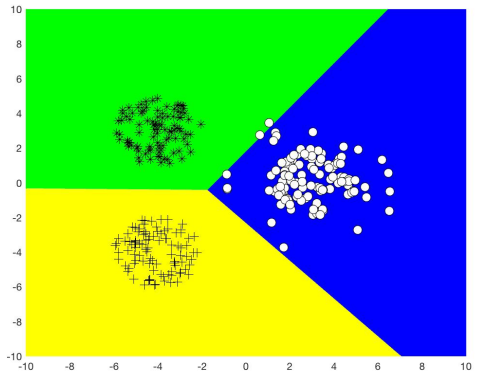


(d) VaR CS-MSVM, $\nu = 0.05$

Figure B.3: Comparison of CVaR CS-MSVM and VaR CS-MSVM with different ν values under 0.88 class 1 probability for Noise 1 dataset.



(a) CVaR CS-MSVM, $\nu = 0.05$

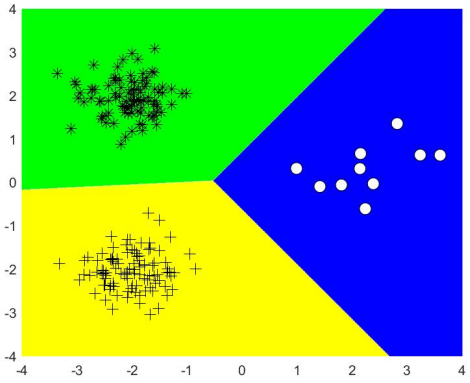


(b) VaR CS-MSVM, $\nu = 0.05$

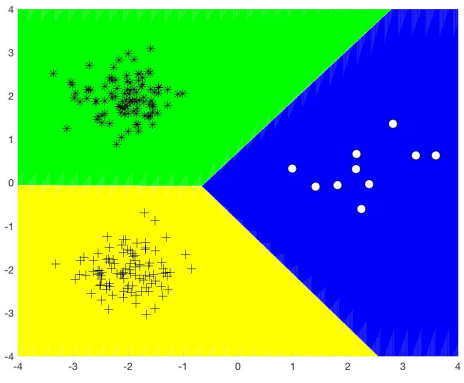
Figure B.4: Comparison of CVaR CS-MSVM and VaR CS-MSVM with different $\nu = 0.05$ under 0.1 class 1 probability for Outlier 3 dataset.

Appendix C

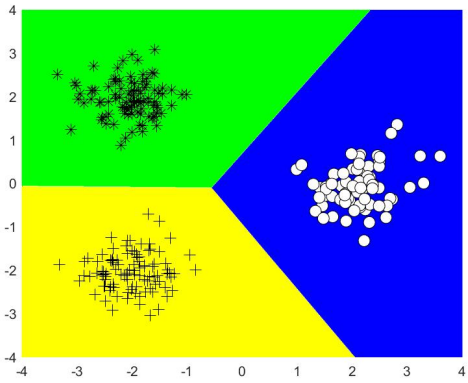
Comparison of VaR WW-MSVM and VaR CS-MSVM



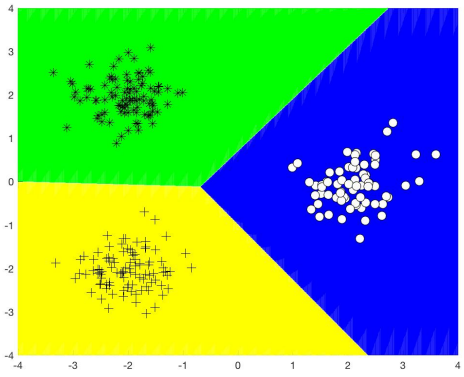
(a) VaR WW-MSVM, Ratio 6 dataset



(b) VaR CS-MSVM, Ratio 6 dataset



(c) VaR WW-MSVM,
Ratio 2 dataset



(d) VaR CS-MSVM,
Ratio 2 dataset

Figure C.1: Comparison of CS-MSVM and VaR CS-MSVM with $\nu = 0.1$ under 0.1 class 1 probability for Ratio 2 and 6 datasets.