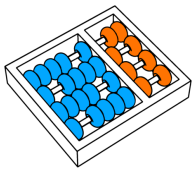
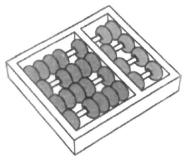


Daniel Bastos Moraes

"Low False Positive Learning with Support Vector Machines"

"*Máquina de Vetores de Suporte com Restrição de Falsos Positivos*"

CAMPINAS
2014



University of Campinas
Institute of Computing

Universidade Estadual de Campinas
Instituto de Computação

Daniel Bastos Moraes

"Low False Positive Learning with Support Vector Machines"

Supervisor(s)/Orientador(es)
Prof. Dr. Anderson de Rezende Rocha
Prof. Dr. Jacques Wainer

"Máquina de Vetores de Suporte com Restrição de Falsos Positivos"

MSc Dissertation presented to the Post Graduate Program of the Institute of Computing of the University of Campinas to obtain a Mestre degree in Computer Science.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Computação da Universidade Estadual de Campinas para obtenção do título de Mestre em Ciência da Computação.

This volume corresponds to the final version of the Dissertation defended by Daniel Bastos Moraes, under the supervision of Prof. Dr. Anderson de Rezende Rocha.

Este exemplar corresponde à versão final da Dissertação defendida por Daniel Bastos Moraes, sob orientação de Prof. Dr. Anderson de Rezende Rocha.

Supervisor's signature

Assinatura do Orientador(a)

CAMPINAS
2014

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

M791L

Moraes, Daniel Bastos, 1987-
Low false positive learning with support vector machines / Daniel Bastos Moraes. – Campinas, SP : [s.n.], 2014.

Orientador: Anderson de Rezende Rocha.
Coorientador: Jacques Wainer.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Aprendizado do computador. 2. Algoritmos. I. Rocha, Anderson de Rezende,1980-. II. Wainer, Jacques,1958-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Máquina de vetores de suporte com restrição de falsos positivos
Palavras-chave em inglês:
Machine learning
Algorithms
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Anderson de Rezende Rocha [Orientador]
Moacir Pereira Ponti Junior
Ricardo da Silva Torres
Data de defesa: 19-02-2014
Programa de Pós-Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 19 de fevereiro de 2014, pela Banca examinadora composta pelos Professores Doutores:

Prof. Dr. Moacir Pereira Ponti Junior
ICMC / USP

Prof. Dr. Ricardo da Silva Torres
IC / UNICAMP

Anderson de Rezende Rocha
IC / UNICAMP

Institute of Computing /*Instituto de Computação*
University of Campinas /*Universidade Estadual de Campinas*

Low False Positive Learning with Support Vector Machines

Daniel Bastos Moraes[1]

February 19, 2014

Examiner Board/*Banca Examinadora*:

- Prof. Dr. Anderson de Rezende Rocha (Supervisor/*Orientador*)
- Prof. Dr. Jacques Wainer (Supervisor/*Orientador*)
- Prof. Dr. Ricardo da Silva Torres
  Institute of Computing - UNICAMP
- Prof. Dr. Moacir Ponti Jr.
  External, ICMC - USP
- Prof. Dr. Siome Klein Goldenstein
  Substitute, Institute of Computing - UNICAMP
- Prof. Dr. João Paulo Papa
  External Substitute, UNESP - Bauru

[1]Financial support: CNPq and CAPES scholarship 2011–2013

Abstract

Most machine learning systems for binary classification are trained using algorithms that maximize the accuracy and assume that false positives and false negatives are equally bad. However, in many applications, these two types of errors may have very different costs. For instance, in medical screening applications, falsely determining that a patient is healthy is much more serious than falsely determining that she has a certain medical condition. In this work, we consider the problem of controlling the false positive rate on Support Vector Machines, since its traditional formulation does not offer such assurance. To solve this problem, we define a feature space sensitive area, where the probability of having false positives is higher, and use a second classifier (k-Nearest Neighbors) in this area to better filter errors and improve the decision-making process. We compare the proposed solution to other state-of-the-art methods for low false positive classification using 33 standard datasets in the literature. The solution we propose shows better performance in the vast majority of the cases using the standard Neyman-Pearson measure.

Resumo

A maioria dos sistemas de aprendizado de máquina para classificação binária é treinado usando algoritmos que maximizam a acurácia e assume que falsos positivos e falsos negativos são igualmente ruins. Entretanto, em muitas aplicações, estes dois tipos de erro podem ter custos bem diferentes. Por exemplo, em aplicações de triagem médica, determinar erroneamente que um paciente é saudável é muito mais sério que determinar erroneamente que ele tem uma certa condição médica. Neste trabalho, nós abordamos o problema de controlar a taxa de falsos positivos em Máquinas de Vetores de Suporte (SVMs), uma vez que sua formulação tradicional não provê garantias desse tipo. Para resolver esse problema, definimos uma área sensível no espaço de características onde a probabilidade de falsos positivos é mais alta e usamos um segundo classificador (k-vizinhos mais próximos) nesta área para melhor filtrar os erros e melhorar o processo de tomada de decisão. Nós comparamos a solução proposta com outros métodos do estado da arte para classificação com baixa taxa de falsos positivos usando 33 conjuntos de dados comuns na literatura. A solução proposta mostra melhor performance na grande maioria dos casos usando a métrica padrão de Neyman-Pearson.

Acknowledgements

First of all I would like to thank God for my existence, for keeping me healthy, for giving the strength to wake up every day and overcome the many challenges that are presented to me, and for allowing me to accomplish this great achievement.

A research work takes time, focus, patience, and resources. Therefore I thank:

My advisor Anderson Rocha, example of professionalism and competence, to whom I own much of my formation. Thank you very much for all the guidance, encouragement, and continuous support.

My co-advisor Jacques Wainer, for the great contribution in this work and for being always available when I needed.

My colleagues from the RECOD lab and the other colleagues from the Institute of Computing, for the friendship and sharing of ideas and difficulties; the Institute of Computing, for the opportunity to perform an excellent graduate course; and CAPES, for the financial support, indispensable to fulfill this work.

I also thank those who are the grace of my life:

To my parents Moraes and Graça, for the patience, and for the continuous strive to give their best so that I could have the best education and followed the path of righteousness; and to my brother Rafael, for being always present during difficult times. You are my pillars.

To my wonderful girlfriend Luciana, for the understanding and complete support. I have no doubts that you are the right person to be forever by my side.

To my friends Andrei Braga, Diogo Araújo and Filipe Lopes, for helping me whenever I requested. Your contribution was essential for the accomplishment of this work.

To my good friends Filipe Mendonça, Helder Prado, Henrique Prado, Isaac Augusto, Marcelo José, Rafael Araújo, and Tiago Mendoça, for the honest and constant fellowship.

"Stay hungry, stay foolish."
Steve Jobs

Contents

List of Tables

List of Figures

Chapter 1

Introduction

One of the pilars of science is the experimentation and collection of data about the world. This raw data must be manipulated and understood in order to extract information, i.e. make it useful for us. One way to do this is through the use of theoretical *a priori* models, which are usually built manually, without data analysis.

Besides theoretical *a priori* models, the data can also be analyzed with statistical methods so that an experimental model can be generated. Machine learning, which has been widely used in several problems from different areas, including biology, computer science, medicine, and many others, can provide useful statistical methods. This is possible because machine learning algorithms are able to learn patterns from data, in order to try to predict future observations.

There are several kinds of machine learning algorithms. Some of them are supervised learning, unsupervised learning and reinforcement learning. This work is focused on supervised learning algorithms, more specifically in classification problems.

1.1 Classification

In a classification problem, the task is to find a classification function which is capable of correctly distinguishing new samples. In a typical classification setting, we are given a sample of training vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, each belonging to one of a set of classes, indicated by the respective labels $y_1, \ldots, y_n \in \{1, \ldots, m\}$. The aim of a classifier is then to find a function $f : \mathbb{R}^d \to \{1, \ldots, m\}$ that accurately predicts the label when presented with a new sample [31].

For instance, consider a binary classification problem, that is, a classification problem where $m = 2$. In Figure 1.1 we have two sets of points in a space of two dimensions ($d = 2$) and a classification function that correctly separates the samples. In that particular example, the problem is linearly separable, so we could solve it just by adjusting a straight-

line between the samples. Thus, in this case the binary classifier could simply be defined by the straight-line equation, $f(\mathbf{x}) = \text{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b = 0)$, where $\mathbf{w} \in \mathbb{R}^d$ is a vector of coefficients and b is a parameter that indicates the offset of $\mathbf{w}$ with respect to the origin. The problem then boils down to finding the line that better separates the samples, that is, the best $(\mathbf{w}, b)$.

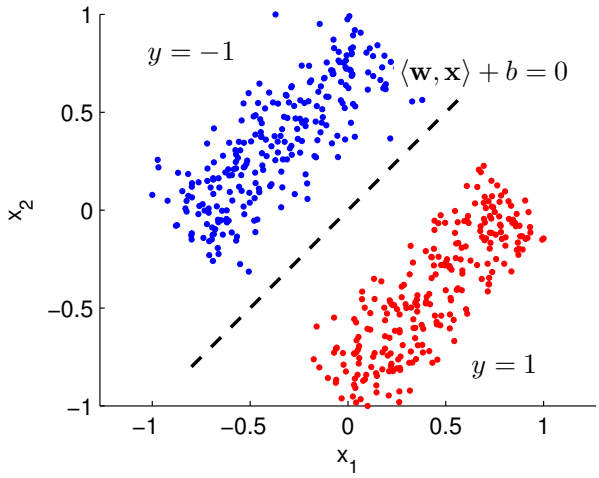


Figure 1.1: A two-dimensional data with positive (red) and negative (blue) samples.

Sometimes the classifier may need to separate the data through a non-linear decision boundary. This usually happens when the data is not linearly separable, but, even when it is, a non-linear decision boundary could be a better model to the data. However, in supervised learning algorithms, complex non-linear decision boundaries could cause a problem called *overfitting* [15], that occurs when the model describes noise instead of the underlying relationship of the data. The reason is that the training data usually has some noise, and too complex models will able to perfectly separate the training data and carry the noise with it. To avoid this problem most classifiers try to adjust simpler decision boundaries that "should be more generalizable". Figure 1.2 ilustrates this, where some points were incorrectly classified in order to define a simpler and more generalizable decision boundary.

1.2 Classification with a false positive constraint

Most machine learning systems are trained using algorithms that maximize accuracy and assume that false positives and false negatives are equally bad. However, there are several applications that are sensitive to false positives, such as spam filtering, face recognition, and computer-aided diagnosis. In these applications, the errors from one class are much

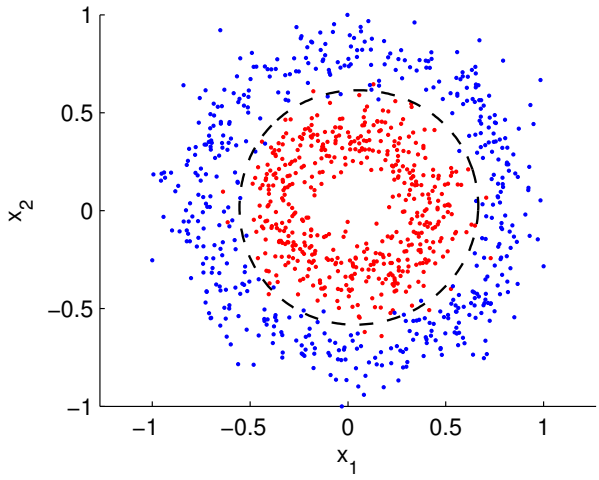


Figure 1.2: A non-linear decision boundary adjusted to a binary problem.

more costly than errors from the other class, and keeping the false positive rate under the maximal tolerance is usually a concern prior to achieve high classification accuracy.

In a spam filtering system, for example, incorrectly classifying a good mail as spam is much worse than misclassifying a few pieces of spam [41]. In computer-based diagnosis, especially if the automated system is being used for triage of patients, falsely determining that a case is normal is much more serious than falsely determining that the case is abnormal. If a case is flagged as abnormal, usually a more costly diagnostics will be applied to the case, but a case flagged as normal will not be further investigated. Thus a case falsely determined as normal will remain wrongly determined, whereas a case falsely determined as abnormal will be further examined, which eventually would determine it as indeed normal.

We will call the situation of wrongly flagging a case as normal (which has higher cost) as a **false positive**[1], which is also called in the literature as a **false alarm**. Formally, for a given classifier f and a new sample $\mathbf{x}_i \in \mathbb{R}^d$ where d is the feature space dimensionality, its class is denoted by y_i while $f(\mathbf{x}_i)$ denotes the predicted class of $\mathbf{x}_i$ by f. A false positive is a point $\mathbf{x}_i$ such that $f(\mathbf{x}_i) = +1$, but $y_i = -1$. A false negative is a point $\mathbf{x}_j$ such that $f(\mathbf{x}_j) = -1$, but $y_j = +1$. The **false positive rate** of the classifier f is then:

$$\mathrm{FP}(f) = \frac{|\{\mathbf{x}_i \mid \mathbf{x}_i \text{ is a false positive}\}|}{\{\mathbf{x}_i \mid y_i = -1\}|} \tag{1.1}$$

Similarly, the false negative rate $\mathrm{FN}(f)$ is the ratio of the number of false negatives divided by the number of positive cases.

[1]This is potentially confusing because the medical literature treats the normal case as *negative* and thus in the medical literature one would like to limit the false negative to a very low value. We will follow the computer science literature that prefers to call the costly mistake as false positive.

Although there are many real problems in which the false positive rate must be controlled, the primary goal of most classifiers is to achieve high classification accuracy. Support Vector Machines (SVM) is a good example of that. It is a powerful algorithm for binary classification, known for its ability to handle high dimensional data efficiently. It has been widely used in many applications providing state-of-the-art accuracy to many classification problems. However, the traditional support vector classifier formulation penalizes errors in both classes equally, and offers no assurance regarding the false positive rate. Thus, in problems such as spam filtering, for which a false positive rate constraint must be complied, the traditional SVM can be useless.

Observing the aforementioned limitations, some extensions to SVM have been proposed in order to make it able to control errors in an asymmetric form. The most common techniques for that are the Bias-Shifting (BS) [10, 11, 17, 40] and the Cost-Sensitive SVM (CS-SVM) [10, 11, 34, 18, 25]. While the former tries to control the false alarms by shifting the SVM's decision boundary toward the sensitive class (positive in our case), the latter tries to adjust cost parameters (slack variables) from the SVM's formulation in order to make misclassifications from the sensitive class more costly than in the other class. The CS-SVM offers state-of-the-art results on the problem of low false positive classification, and several studies have been made in this direction. However, it can be very inefficient on larger datasets and finding these cost parameters may be impracticable on problems for which a lot of data is required, specially in the dawning age of big data. The BS, on the other hand, gives results that are close to the CS-SVM and is as efficient as the traditional SVM.

Although some researchers have considered solutions for the low false positive problem that are not based on SVMs [5, 38, 39] (more papers referenced in the related work section), SVM-based solutions have demonstrated to be more effective in many situations and are the focus of this work.

In this work, we propose the Risk Area SVM (RASVM), a novel method to efficiently solve the low false positive classification problem. It is an extension of the traditional support vector machine classifier which is able to control the false positive rate, given a user-specified maximum allowed threshold. The RASVM selects a sensitive region close to the SVM's decision boundary with a high incidence of false positives. Within that region, which we call *risk area*, the decision to classify a sample as positive is based on inspecting its k-nearest neighbors (k-NN) and a new data point will be only classified as positive if all its k-nearest neighbors are also positive.

The idea of combining k-NN within a region around the SVM's decision boundary in order to control false positives was first introduced in [1] to solve a problem of automatic triage. This work extends upon and further explores those ideas to build a more robust and generalized method for controlling false positives. The requirement of keeping the

false positive rate bounded below a certain level while minimizing the false negative rate is also called the *Neyman-Pearson* classification paradigm [33, 32]. The requirement can also be stated as of maximizing the accuracy (correct predictions), while keeping the false positive rate bounded. Thus, given a user specified threshold α, our objective is to:

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & \text{FN}(f), \\ \text{subject to} \quad & \text{FP}(f) \leq \alpha. \end{aligned} \tag{1.2}$$

1.3 Objectives

In this dissertation, we present new methods for controlling the false positive rate on SVM. Our goal is to extend the traditional support vector classifier, making it able to control the false positive rate, given a user-specified threshold.

Our main objectives are to:

- study the existing methods for controlling false positives on SVM;
- develop new methods for controlling false positives on SVM;
- implement the state-of-the-art methods;
- compare our methods with the ones on the literature.

1.4 Outline of this dissertation

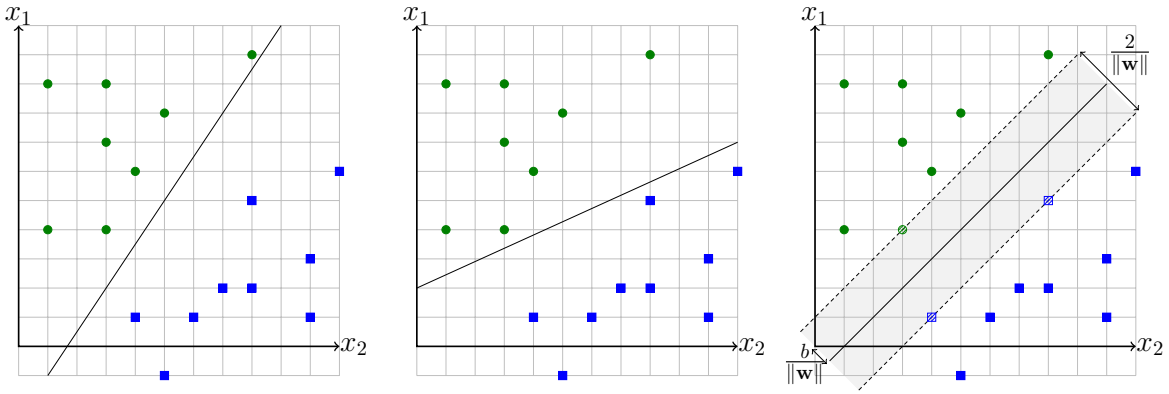
This text is organized as follows: In Chapter 2, we briefly review SVMs, introducing the C-SVM, the 2C-SVM, and the ν-SVM. In Chapter 3, we discuss alternatives to SVM based on the Neyman-Pearson classification. In Chapter 4, we describe our method and its variations. In Chapter 5, we describe the experiment settings common to all methods evaluated herein. In Chapter 6, we compare our results against the results of state-of-the-art methods. Finally, we conclude on Chapter 7 with a discussion and some directions for future research.

Chapter 2

Support Vector Machines

In machine learning, Support Vector Machines (SVM) are among the most effective methods for binary classification [31], and was originally developed by Cortes and Vapnik [8]. The idea is to find the maximum-margin hyperplane $(\mathbf{w}, b)$ in a high-dimensional space $\mathcal{H}$ that accurately separates the positive instances from the negative ones. This concept is motivated by the statistical learning theory [37], which says that the probabilistic test error is minimized when the margin is maximized.

Figure 2.1 illustrates an example, where there are multiple hyperplanes that separates the positive from the negative class. The SVM, however, chooses the maximum-margin hyperplane, which is shown in Figure 2.1 (c). Filled figures represent training data and striped figures the support vectors. Positive data represent circles while squares denote the negative data points.



(a) Separating hyperplane. (b) Separating hyperplane. (c) Maximum-margin hyperplane.

Figure 2.1: Examples of two possible separating hyperplanes and the maximum-margin hyperplane used in SVM.

2.1 Maximum-margin hyperplane linear classifier

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ denote a sample of training vectors, each belonging to one of two classes, indicated by the respective labels $y_1, \ldots, y_n \in \{+1, -1\}$. To find the maximum-margin hyperplane we need to select $(\mathbf{w}, b)$ such that it corretly separates the positive from the negative class, and the shortest distance between the hyperplane and its closest point $\mathbf{x}_i$ is maximum. We have then the following problem:

$$\begin{aligned} \max_{\gamma,\mathbf{w},b} \quad & \gamma = \min_{i=1,\ldots,n} \gamma_i \\ \text{subject to} \quad & y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) \geq \gamma, \quad \text{for } i = 1, \ldots, n, \\ & ||\mathbf{w}|| = 1, \end{aligned} \tag{2.1}$$

where γ_i is the distance between $(\mathbf{w}, b)$ and $\mathbf{x}_i$. This is a non-convex problem, and is difficult to be solved numerically. However, it can be shown that the parameters of the maximum-margin hyperplane could be derived by solving the folowing optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}||\mathbf{w}||^2 \\ \text{subject to} \quad & y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) \geq 1, \quad \text{for } i = 1, \ldots, n, \end{aligned} \tag{2.2}$$

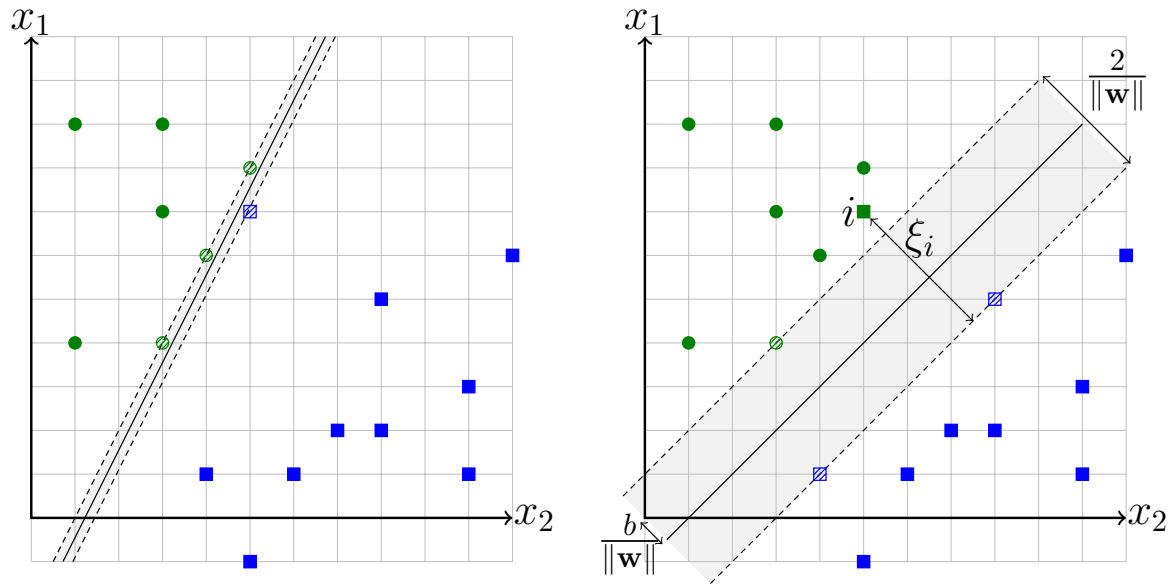
which is a quadratic programming (QP) minimization problem, easier to be solved numerically.

When the data is not linearly separable, it will be difficult to find a solution to the optimization problem shown in Equation 2.2. This problem is usually solved by adding slack variables, so as to still find a maximum-margin hyperplane, even if a few training points have to be misclassified. This concept of relaxing the hard margin constraint is known as soft margin. We can do that by maximizing the margin while softly penalizing points that lie on the wrong side of the decision boundary. Hence, we now have the following minimization problem:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) \geq 1 - \xi_i, \quad \text{for } i = 1, \ldots, n \\ & \xi_i \geq 0, \end{aligned} \tag{2.3}$$

where ξ_i with $i = 1, \ldots, n$, are the slack variables, $C \geq 0$ is a parameter that balances the amount of slack (misclassifications) and the size of the margin, and b is a parameter that indicates the offset of $\mathbf{w}$ with respect to the origin.

Figure 2.2 ilustrates an example where the use of soft margins allowed SVM to define a more generalizable decision boundary.



(a) (b)

Figure 2.2: The undesirable decision boundary (with small margin) in (a) becomes more generalizable in (b) by allowing the point i to be misclassified.

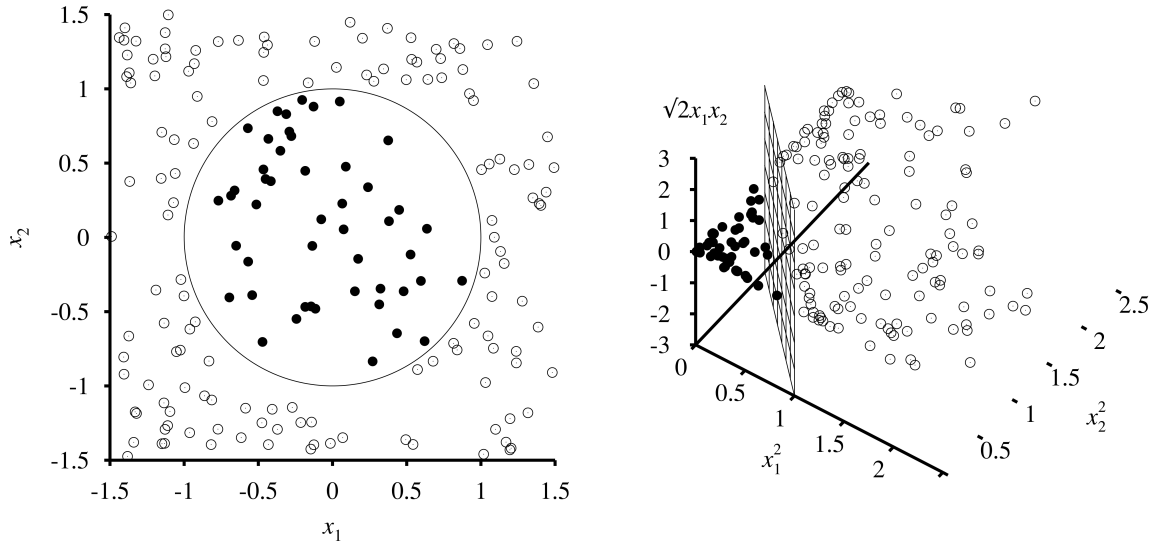
2.2 Non-linear decision boundaries

With the formulation in Equation 2.3, we are now able to find the maximum-margin hyperplane even if the data is not linear separable. However, often the nature of the data is such that it is not desirable to separate them with a linear decision boundary. Figure 2.3 ilustrates an example of a two-dimensional data that was mapped onto three dimensions, and was able to be separated by a linear decision boundary.

2.2.1 The C-SVM primal

Usually, SVMs implicitly maps the input data onto a high-dimensional feature space $\mathcal{H}$, so that it can be better separated by an hyperplane. We then need to rewrite the formulation in Equation 2.3 with such mapping. Now we have the problem:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i(\langle\Psi(\mathbf{x}_i),\mathbf{w}\rangle + b) \geq 1 - \xi_i, \quad \text{for } i = 1,\ldots,n, \\ & \xi_i \geq 0, \qquad\qquad\qquad\qquad\quad \text{for } i = 1,\ldots,n, \end{aligned} \tag{2.4}$$



(a) A two-dimensional data with positive (black) and negative (white) samples.

(b) The same data after mapping into a three-dimensional input space.

Figure 2.3: The circular decision boundary in (a) becomes a linear decision boundary in (b). Figure reprinted from [26].

where $\Psi : \mathbb{R}^d \to \mathcal{H}$ is the transformation that maps $\mathbf{x}$ onto a high-dimensional feature space. The above formulation is known as C-support vector classification (C-SVC) [4, 8], and is the standard formulation of SVM.

2.2.2 The C-SVM dual

In practice, due to the possible high dimensionality of the vector $\mathbf{w}$, it is usually transformed into its dual form and solved:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha} \\ \text{subject to} \quad & \mathbf{y}^T \boldsymbol{\alpha} = 0, \\ & 0 \le \alpha_i \le C, \qquad \text{for } i = 1, \ldots, n, \end{aligned} \tag{2.5}$$

where $\mathbf{e} = [1, \ldots, 1]^T$ is a vector of all ones, Q is an m by m positive semidefinite matrix, with $Q_{ij} = y_i y_j \langle \Psi(\mathbf{x}_i), \Psi(\mathbf{x}_j) \rangle$. Now the transformation Ψ could be implicitly defined by a *kernel* function, such that $\langle \Psi(\mathbf{x}_i), \Psi(\mathbf{x}_j) \rangle = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$. There are specialized algorithms for quickly solving these dual optimization problems that arises on SVM. The most common is the Sequential Minimization Optimization (SMO) algorithm, which was proposed by Platt et al. [24].

The kernel trick. This idea of implicitly mapping the data through a kernel function is known as the *kernel trick*. There are many different kernels available for SVM, being the Gaussian (RBF) kernel the recommended one when we have no prior knowledge on how to represent the data under analysis. The RBF kernel maps the input data onto a Hilbert space of infinite dimensions and is given as follows:

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right) \tag{2.6}$$

The support vector classifier. After solving the problem in Equation 2.5, we recover $\mathbf{w}$ and b by using the primal-dual relationship. Finally, the support vector classifier is given by

$$f_{\mathbf{w},b}(\mathbf{x}) = \operatorname{sgn}(\langle \mathbf{w}, \Psi(\mathbf{x}) \rangle + b). \tag{2.7}$$

Chapter 3

Related Work

There are many practical applications that require the classifier to produce a very low false positive rate. Therefore, several studies have been conducted to develop classifiers in this sense, which include techniques based on Naïve Bayes [29, 2], boosting [6, 38, 22], data compression [5], neural networks [42], ensemble learning [20], and cascade of classifiers [39, 41].

Since the SVM's standard formulation (C-SVM) penalizes errors in both classes equally, it does not offer assurance regarding the false positive rate. This (and other) limitations led to the emergence of many different formulations, each of them generally focused on a particular problem. The most common SVM formulation for the low false positive learning is the $2C$-SVM, proposed by Osuna et al. [23]. It is basically an extension of the C-SVM formulation in which we can define different costs to the positive and negative classes by adjusting the parameters C_+ and C_-. Let $I_+ = \{i \mid y_i = +1\}$ and $I_- = \{i \mid y_i = -1\}$. The $2C$-SVM has its primal formulation

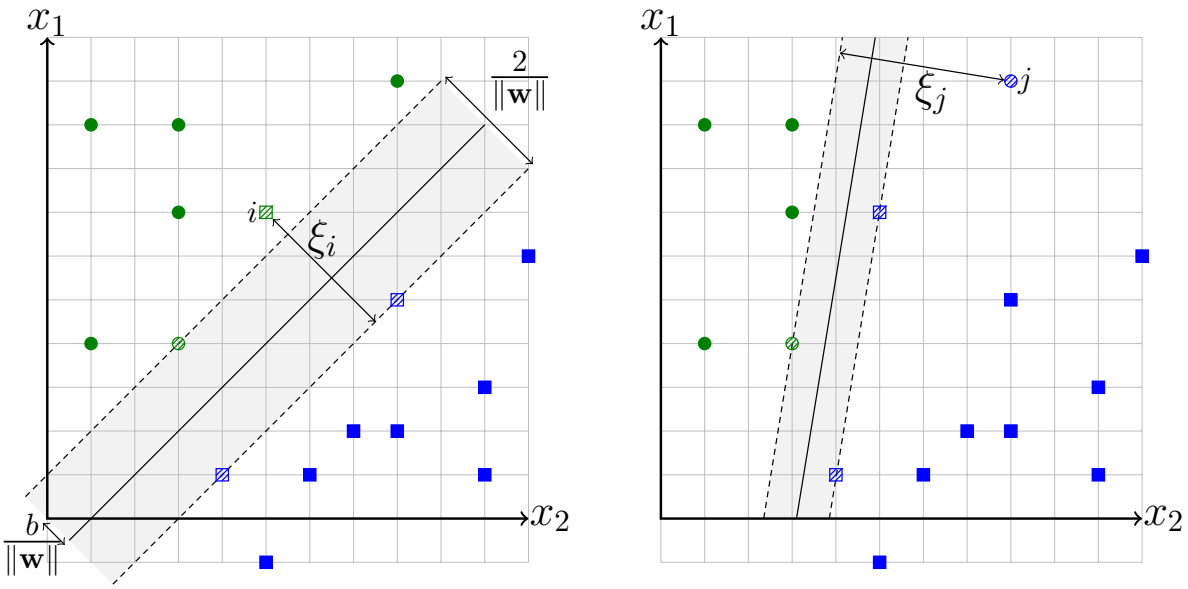
$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C_+ \sum_{i \in I_+} \xi_i + C_- \sum_{i \in I_-} \xi_i \\ \text{subject to} \quad & y_i(\langle \Psi(\mathbf{x}_i), \mathbf{w} \rangle + b) \geq 1 - \xi_i, \qquad \text{for } i = 1, \ldots, n, \\ & \xi_i \geq 0, \qquad \text{for } i = 1, \ldots, n, \end{aligned} \tag{3.1}$$

and the Lagrangian dual,

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha} \\ \text{subject to} \quad & \mathbf{y}^T \boldsymbol{\alpha} = 0, \\ & 0 \leq \alpha_i \leq C_+, \qquad \text{for } i \in I_+, \\ & 0 \leq \alpha_i \leq C_-, \qquad \text{for } i \in I_-, \end{aligned} \tag{3.2}$$

where the cost parameters C_+ and C_- are usually defined through a third parameter $\gamma \in [0, 1]$, so that $C_+ = \gamma$ and $C_- = (1 - \gamma)$ [11].

Those formulations in which it is possible to adjust the costs of the slacks on the SVM formulation are known as Cost-Sensitive SVM (CS-SVM). With proper adjustment in its cost parameters, the CS-SVM is able to consider misclassifications in the positive class more costly than in the negative class, therefore forcing the decision boundary to avoid false positives. This idea is illustrated in Figure 3.1, where the $2C$-SVM on (b) was able to avoid the false positive denoted by i on (a), despite bringing in a new false negative, denoted by j. Filled figures represent training data and striped figures the support vectors. Positive data represent circles while squares denote the negative data points.



(a) C-SVM decision boundary

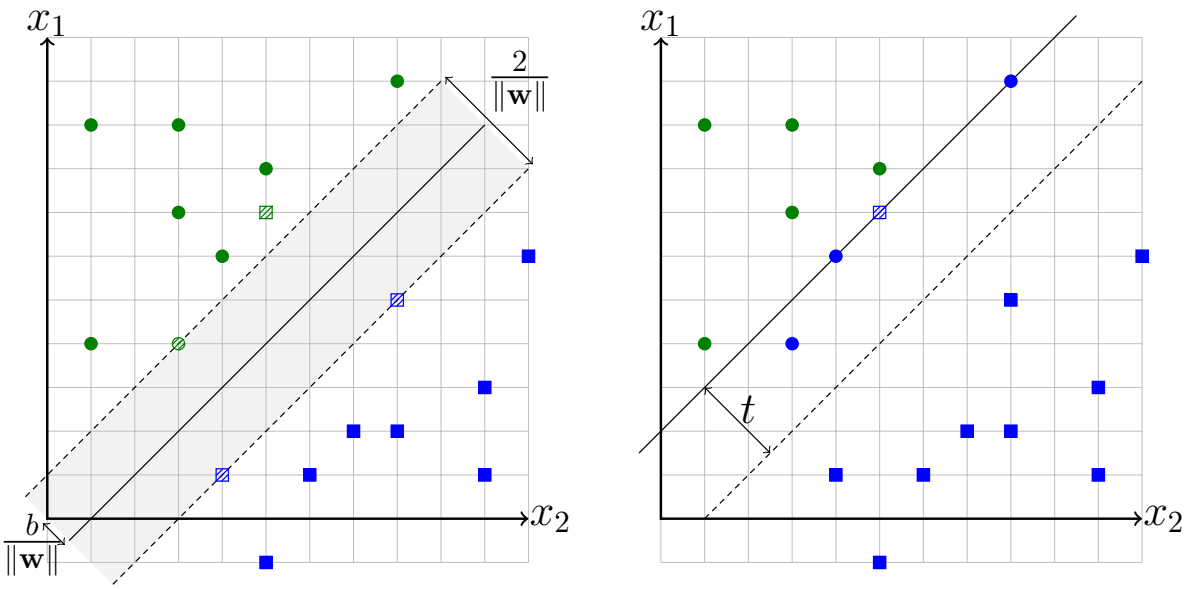
(b) $2C$-SVM decision boundary

Figure 3.1: The effect of the $2C$-SVM formulation on the low false positive classification problem, in comparison to the standard C-SVM.

CS-SVMs have been used in many recent studies to solve problems on low false positive classification for supervised [10, 11] and semi-supervised [18, 25] learning. It usually offers state-of-the-art results on the problem of low false positive classification, and several studies have been conducted in this direction. However, this approach can be very time consuming when dealing with larger datasets so that properly adjusting the parameters C_+ and C_- may be impracticable on problems for which a lot of data is required.

Another common method for controlling the false positive rate on SVMs by is known as Bias-Shifting (BS), and was proposed by Shawe-Taylor and Karakoulas [16]. It is a simple method, which shifts the decision boundary toward the sensitive class by simply adjusting

the threshold parameter b. This idea was motivated after Shawe-Taylor [35] showed that the distance of a data point to the decision boundary is related to its probability of misclassification. The BS technique is usually optimized by selecting the threshold t that minimizes $\mathrm{FN}(f)$ while ensuring that $\mathrm{FP}(f) \leq \alpha$ (on the training data), where α is the user-specified maximum allowed false positive rate parameter. After finding t, a testing instance will be predicted as positive when $\langle \mathbf{w}, \Psi(\mathbf{x}) \rangle + b \geq t$. This method is simple and very efficient. However, it frequently results in classifiers for which the false positive rate significantly exceeds α in the test data. Some researchers have gone even further and successfully applied BS ideas to open-set classification problems [28, 12, 9]. Figure 3.2 ilustrates the BS technique, where the decision boundary is shifted by t on (b) to solve the false positive that existed in (a).



(a) SVM classifier

(b) SVM classifier with BS technique

Figure 3.2: The effect of the BS technique with an SVM classifier on the low false positive classification problem.

A Cost-Sensitive extension was also proposed to the ν-SVM, which is another SVM formulation proposed by Schölkopf et al. [30]. It has been showed that the ν-SVM is equivalent to the traditional C-SVM [7] formulation. This Cost-Sensitive extension is known as 2ν-SVM, and it has been also demonstrated to be equivalent to the $2C$-SVM [11]. Davenport et al. [10, 11] adopted the 2ν-SVM on low false positive classification problems, and provided a careful characterization of the 2ν-SVM parameter space, as well as error estimation approaches based on smoothing that improves the accuracy of cross-validation techniques. In addition, they proposed coordinate descent strategies for parameter selec-

tion that offer significant gains in the 2ν-SVM training time.

Another SVM formulation for the low false positive problem was proposed by Wu et al. [40], which is called Asymmetric Support Vector Machines (ASVM). Their approach tries to provide a better description of the positive class by considering a higher confidence area among the positive training samples. Figure 3.3 shows a logical view of ASVM, where two margins are maximized — the core-margin (i.e., $\rho/||\mathbf{w}||$) and the traditional class-margin (i.e., $\gamma/||\mathbf{w}||$), as in SVM.

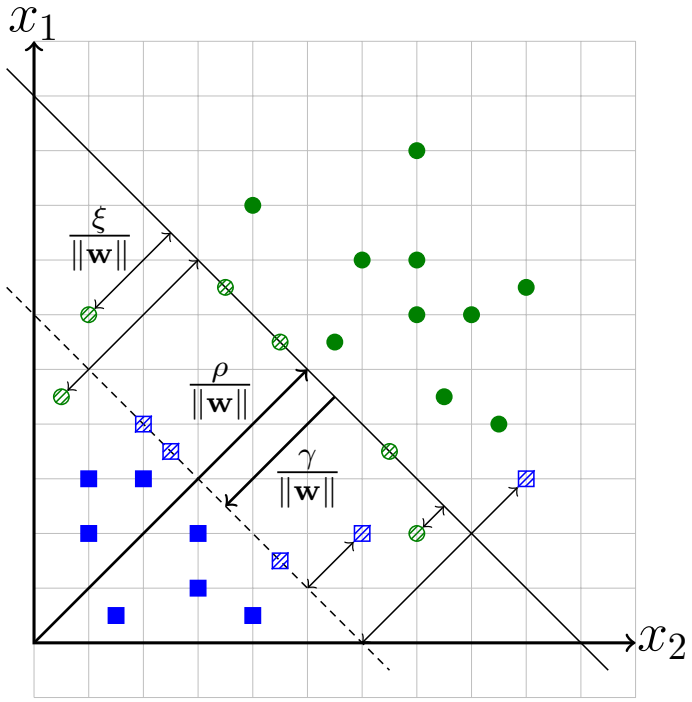


Figure 3.3: A logical view of ASVM. Two margins, the core-margin ($\rho/||\mathbf{w}||$) and class-margin ($\gamma/||\mathbf{w}||$), are maximized simultaneously to allow classifying the negative class and the core of the positive class. Figure adapted from [40].

Chapter 4

Risk Area SVM

In this chapter, we present the Risk Area SVM (RASVM), an extension of the traditional support vector machine classifier that incorporates the ability to control the false positive rate to a user-specified maximum threshold. We start discussing about how misclassifications generally occur on SVMs, and how we can use this notion to develop classifiers that are robust to false positives. After that, we show how the RASVM classifies the points inside the sensitive region of the SVM's feature space (we call it the *risk area*), that is, a region with higher probability of having misclassifications. Next, we describe how the RASVM selects the *risk area*. Finally, we introduce a solution for making the classification inside the *risk area* much faster.

4.1 Motivation

Our approach is mainly motivated by the weaknesses of the state-of-the-art methods for controlling false positives on SVMs. Table 4.1 ranks these methods on important aspects of the low false positive learning problem: (1) false positive control; (2) true positive rate; (3) insensitivity to unbalance; (4) efficiency. We also include on this table our method. As we can see, when compared to the state-of-the-art methods, our solution have superior performance on controlling false positives and good TP rate, insensitivity to unbalance, and efficiency.

To build a classifier that is more robust to false positives, we based on two facts that generally occur on SVMs. Below, we will describe these notions and show how we could explore them.

It is known that, in a support vector classifier, we can only have a high confidence in the classification of a point if it is far from the SVM's decision boundary [21, 36]. In other words, the further away a point is from the hyperplane, more confidence we have in its classification. We can derive this notion to another important observation: *the majority*

	FP control	TP rate	Unbalance	Efficiency
BS	3	4	2	1
CS-SVM	2	3	3	3
ASVM	4	1	1	4
RASVM	1	2	2	2

Table 4.1: Advantages and disadvantages of the state-of-the-art methods for controlling false positives on SVMs.

of misclassifications are usually close to the decision boundary (around the hyperplane). This is a well-known fact from SVM, and is even the primary motivation behind the BS technique — shifting the hyperplane towards the positive class should solve most of the false positives. Such shifting, however, will not only solve false positives. Instead, it will classify as negative *all* the points that are below the shifted decision boundary. This solution therefore often results in a significant reduction of the true positive rate.

As shown in Chapter 2, the SVM finds the maximum-margin hyperplane that better separates the positive from the negative class. So, it is always a linear decision boundary. In order to get non-linear boundaries, SVMs use the notion of kernels to implicitly map the input data onto a high dimensional feature space. This allows SVM to solve the original problem through a linear decision boundary in the augmented space, and when returning to the original space, this solution could be a non-linear decision boundary. This notion is illustrated in Figure 2.3. However, the data could still not be linearly separable in this higher-dimensional space, and thereof comes the notion of soft margin, to allow some training points to be misclassified. This concept of relaxing the hard margin is also useful to define decision boundaries that are more generalizable for new classifications.

For the low false positive classification, these misclassified training points (known as slacks) can give us valuable information regarding the regions of the SVM's feature space where we cannot trust in the classification of a data point as positive. Classifying as positive a testing point that is near to misclassified training points can be risky, and should be avoided when the goal is to achieve a low false positive rate. This is ilustrated in Figure 4.1, in which a training point (highlighted with a red circle) was misclassified in order to define a more generalizable decision boundary.

These ideas are the basis of the Risk Area SVM. First, our approach is focused on the region around the hyperplane, since it should have a higher incidence of false positives; we call this region the *risk area*, and we will show next how we select it. After that, we use a second classifier in order to carefully classify testing points inside the *risk area*. As we will show below, this second classifier will consider the misclassified training points

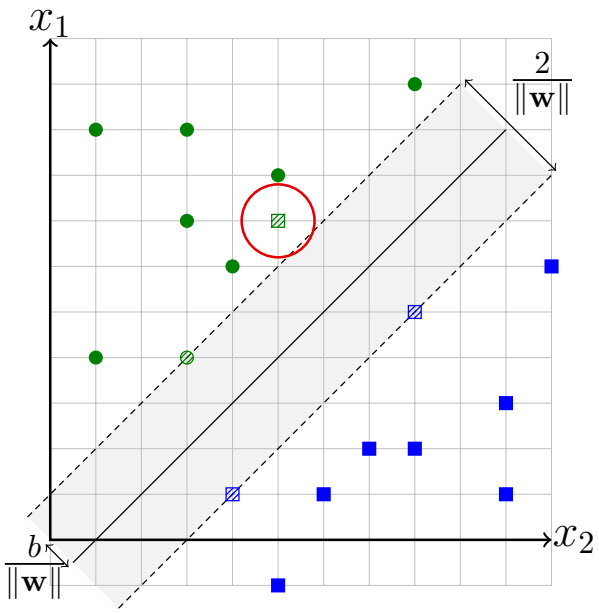


Figure 4.1: Training point misclassified by the SVM (soft margin).

from the SVM classifier in order to avoid false positives.

4.2 The Risk Area SVM classifier

The Risk Area SVM (RASVM) is an extension of the traditional support vector machine classifier that incorporates the ability to control the false positive rate to a user-specified maximum threshold. It is grounded on two presuppositions: (1) most misclassified points on SVMs are close to the decision boundary; (2) these misclassified training points define sensitive areas, so that classifying a testing point as positive in these regions should be avoided. Thus within a region close to the decision boundary of an SVM, and given that our problem is to limit the false positives, one should be very careful in determining a new data point in this area as positive. We call this region around the decision boundary as *risk area*, and the decision to classify a data point in this area as positive will only be made if all of its k-nearest neighbors (for a fixed k) within a training set are also positive. If the data point is outside the *risk area*, then the usual SVM rule for classification applies and the data will be classified according to the side of the decision boundary it lays. Despite making use of the fact described in (1), unlike the BS technique, RASVM does not necessarily classify all the samples that are close to the decision boundary as negative. It also makes use of the fact described on (2) to carefully classify the points inside the *risk area*.

Section 4.3 discusses different ways of defining the risk area. For the moment, assume that it is a symmetrical area around the decision boundary. Figure 4.2 depicts an example of classification in RASVM, assuming that $k = 2$, that is, a test sample will be classified as positive if its two closest training data points are also positive. Filled figures represent training data, unfilled figures the testing data, and striped figures the support vectors. Positive data represent circles while squares denote the negative data points. The *risk area* is the red-colored region.

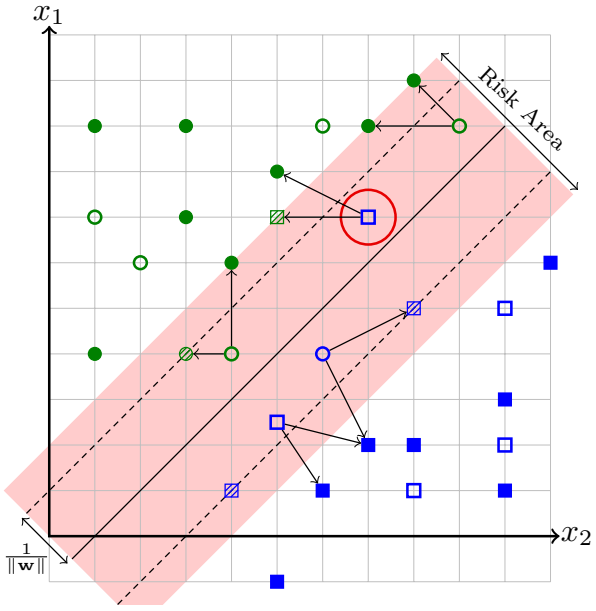


Figure 4.2: An example of the RASVM, with the *risk area*. In this case, we have $k = 2$. The five testing points inside the risk area have their classes defined by the class of its 2-nearest neighbors.

Formally, the classification of an RASVM is performed in two steps. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ denote the training set of data, $\mathbf{z}$ a new data point that must be classified, and $\mathcal{R}$ denote the *risk area*. If $\mathbf{z}$ is outside $\mathcal{R}$, its class is defined by the SVM's decision boundary:

$$f(\mathbf{z}) = \begin{cases} +1 & \text{if } \Psi(\mathbf{z}) + b > 0 \\ -1 & \text{otherwise.} \end{cases}$$

If $\mathbf{z}$ is within the *risk area* $\mathcal{R}$, it will be classified as positive only if all its k-nearest neighbors are all positive:

$$f(\mathbf{z}) = \begin{cases} +1 & \text{if } \forall \mathbf{x}_j \in N_k(\mathbf{z}) \quad f(\mathbf{x}_j) = +1, \\ -1 & \text{otherwise.} \end{cases}$$

where $N_k(\mathbf{z})$ is the neighborhood of $\mathbf{z}$ defined by the k closest points $\mathbf{x}_i$ in the training set [14]. Note that in the SRA and OSSRA forms (the variations from which $\delta > 0$), the

SVM's hyperplane is also shifted by δ together with the *risk area*. So, for every point that is outside $\mathcal{R}$, the ones above the shifted hyperplane are classified as positive, and the remaining as negative.

Closeness implies a metric. So, to get the closest points to $\mathbf{z}$, we consider the Euclidean distance in the feature space $\mathcal{H}$. The Euclidean distance between two points $\mathbf{p}$ and $\mathbf{q}$ is defined as follows:

$$\text{dist}(\mathbf{p}, \mathbf{q}) = \|\mathbf{p}, \mathbf{q}\| = \sqrt{(\mathbf{p} - \mathbf{q}) \cdot (\mathbf{p} - \mathbf{q})}. \tag{4.1}$$

which is equivalent to:

$$\text{dist}(\mathbf{p}, \mathbf{q}) = \sqrt{\|\mathbf{p}\|^2 + \|\mathbf{q}\|^2 - 2\mathbf{p} \cdot \mathbf{q}} = \sqrt{(\mathbf{p} \cdot \mathbf{q}) + (\mathbf{q} \cdot \mathbf{q}) - 2(\mathbf{p} \cdot \mathbf{q})}. \tag{4.2}$$

Thereby, since we are using the RBF as kernel, we could replace the dot products on Euquation 4.2 with that kernel to compute the distances between every pair of points in $\mathcal{H}$.

Definition 4.2.1. Given a testing point $\mathbf{z} \in \mathcal{R}$ and the training points $\mathbf{x}_i, i = 1, \ldots, n$, the Euclidean distance $\text{dist}(\mathbf{z}, \mathbf{x}_i)$ between $\mathbf{z}$ and $\mathbf{x}_i$ in the feature space $\mathcal{H}$ is defined by

$$\text{dist}(\mathbf{z}, \mathbf{x}_i) = \sqrt{\mathcal{K}(\mathbf{z}, \mathbf{z}) + \mathcal{K}(\mathbf{x}_i, \mathbf{x}_i) - 2\mathcal{K}(\mathbf{z}, \mathbf{x}_i)} \tag{4.3}$$

where $\mathcal{K}$ is defined by Equation 2.6.

4.3 Definition of the Risk Area

The *risk area* is a region that is shaped around the SVM's decision boundary in order to outline the samples that will be classified by a second classifier instead of the decision boundary. The general form of the *risk area* is given as follows:

Definition 4.3.1. A testing point $\mathbf{x}_i$ belongs to the *risk area* when

$$\beta_- \leq d(\mathbf{x_i}) - \delta \leq \beta_+, \tag{4.4}$$

where δ is the offset of the *risk area* with respect to the original SVM's hyperplane, β_- and β_+ are the width of the *risk area* above and below the shifted hyperplane, respectively, and $d(\cdot)$ is the oriented (signed) Euclidean distance between $\mathbf{x}_i$ and the hyperplane in the feature space $\mathcal{H}$, which is given by:

$$d(\mathbf{x}_i) = \frac{\mathbf{w}^T \Psi(\mathbf{x}_i) + b}{\|\mathbf{w}\|}. \tag{4.5}$$

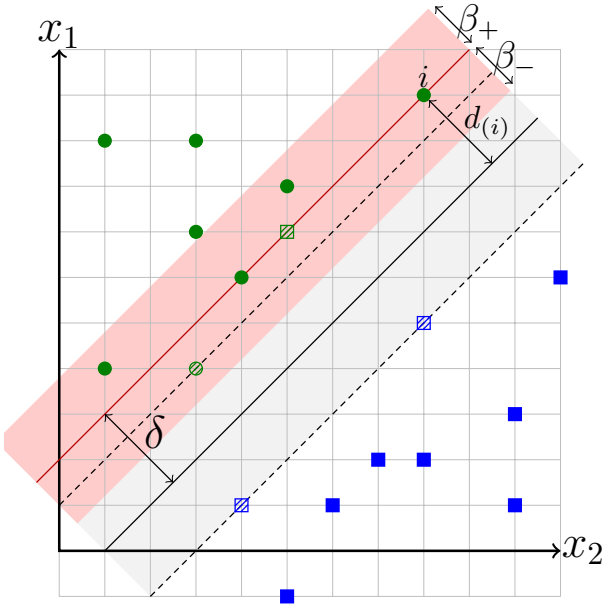


Figure 4.3: The parameters of the *risk area*. The positive samples are represented by circles and the negative samples by squares. The samples that are classified as positive by the RASVM are highlighted in green while those that are classified as negative are in blue.

Figure 4.3 shows the parameters of the *risk area*. In this case, δ is greater than zero, making the *risk area* to be defined around the shifted hyperplane.

Although always located around the hyperplane, we consider different ways of selecting the *risk area*. The simplest case is just called *Risk Area* (RA), in which we select the *risk area* as the region that is distant to the hyperplane at most β. An alternative to this is the *One Sided Risk Area* (OSRA), in which the *risk area* is defined in the same way as in the RA form, but only for the region above the hyperplane.

A more sophisticated way of selecting the *risk area* is the *Shifted Risk Area* (SRA). In this case, we first shift the SVM's decision hyperplane toward the positive class by some threshold δ. Then, we select the *risk area* as the region that is distant to the shifted hyperplane at most β. Finally, an alternative to the SRA form is the *One Sided Shifted Risk Area* (OSSRA), in which we define the *risk area* the same way as in the SRA form, but only for the region above the shifted hyperplane. Table 4.2 shows how the parameters β_+, β_-, and δ are set in each case. The parameters β_+ and β_- are always set to $-\beta$, $+\beta$, or zero. The parameter δ is always set to t or zero.

Note that in the RA form, the second classifier will be applied both above and below the SVM's hyperplane. The intuition behind this is that, besides trying to fix the false positives that could be above the hyperplane, we can also fix some false negatives that are

Table 4.2: How the parameters β_+, β_-, and δ are set in each form of the Risk Area SVM. The parameter t is the same as the obtained in the BS.

		β_-	β_+	δ
RA	Risk Area	$-\beta$	$+\beta$	0
OSRA	One Sided Risk Area	0	$+\beta$	0
SRA	Shifted Risk Area	$-\beta$	$+\beta$	t
OSSRA	One Sided Shifted Risk Area	0	$+\beta$	t

under the hyperplane and increase the true positive rate. In the OSRA form, we restrict the method to just try to solve the false positives that could be above the hyperplane. Figure 4.4 illustrates how the *risk area* is bounded on the RA and OSRA forms.

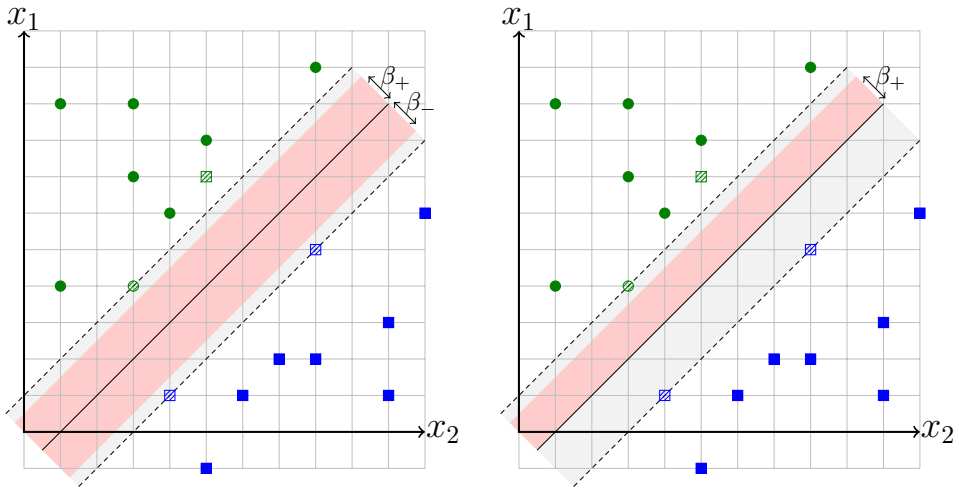


Figure 4.4: Examples of *risk areas* in (a) RA and (b) OSRA forms.

On the SRA and OSSRA forms, besides defining β, we need to set the offset δ. In this work, we use the threshold given by the Bias-Shifting (BS) method, optimized by the Neyman-Pearson score [32]. Figure 4.5 illustrates how the *risk area* is bounded on the SRA and OSSRA forms.

4.4 Optimization of RASVM Parameters

The RASVM has, in its general form, five hyperparameters:

- the C and γ of the standard SVM model (with RBF kernel)

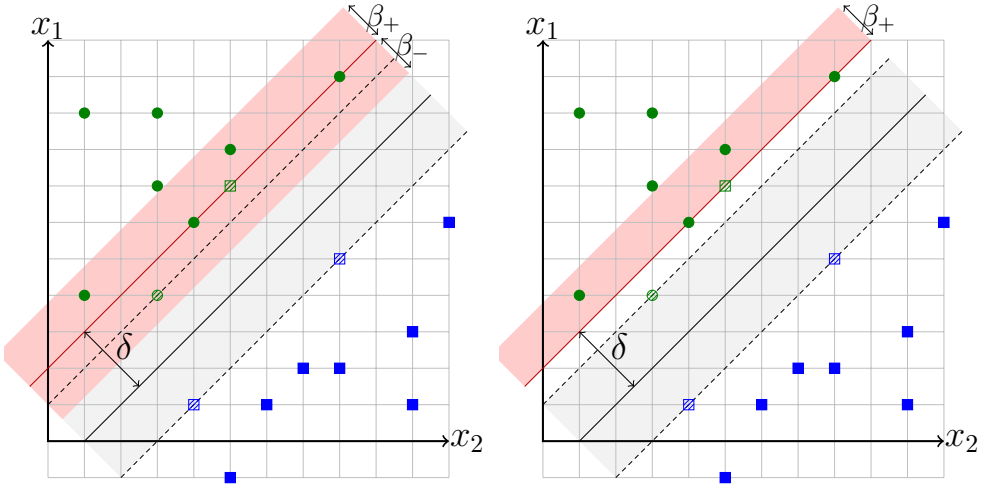


Figure 4.5: Examples of *risk areas* in (a) SRA and (b) OSSRA forms.

- the shift δ
- the width of the *risk area* β
- and the k for the k-NN unanimity voting

These hyperparameters are selected by the following procedures:

- First, C and γ are selected in a grid-search fashion, using a 5-fold validation (on the training set) and selecting for the pair of values with higher mean accuracy over the evaluation sets.
- Second, δ is optimized by moving the hyperplane toward the sensitive class, until finding the position that maximizes the NP-score (see Section 5.1 for the definition and rationale of the NP-score). For the RA and OSRA, $\delta = 0$.
- Finally, β and k are selected also through grid-search using a 5-fold protocol on the training set which also optimizes the NP-score [32].

4.5 Speeding Up the Classification inside the Risk Area

A potential advantage of the k-NN classifier is that it requires no training step [14]. However, because all the work is done at run-time, it can have poor run-time performance

in larger training sets [3]. If a new data falls within the risk area, a naïve implementation would have to scan all of the training set to determine the closest k neighbors. Even if one uses more efficient data structures to organize the data (like metric trees [27]) to speed up the search, one still has to "remember" all the training data.

If the assumption behind the RASVM is correct, that misclassifications happen close to the decision boundary, then possibly only the training data close to the boundary could be used to determine the k-NN. If one is willing to use an SVM classifier, one will have to "remember" all the training set that are support vectors for the decisions boundary, i.e., the training data that holds the margins of the decision boundary or are on the wrong side of the margin and thus contribute to the slack variables ξ_i in Equation 2.4.

Therefore, the faster variant of RASVM, which we call RASVM-SV (after "support vector") only looks to the k neighbors among the support vectors of the SVM classifier. The idea is that the number of support vectors in SVM is usually much smaller than the size of the training set. Besides making the classification on the *risk area* faster (because less training points will be tested to select the k-nearest neighbors), the memory needed by the RASVM-SV would be similar to the one needed by the SVM itself.

Chapter 5

Evaluation Methodology

This chapter discusses the methodology for comparing the proposed methods RASVM and RASVM-SV to other techniques in the literature: BS, ASVM, and cs-SVM.

5.1 Performance Measure

To compare the performance of two different classifiers under the Neyman-Pearson criterium, we will use the Neyman-Pearson score (NP-score) proposed by Scott [32], which is defined as:

$$\frac{1}{\alpha} \max\{\mathrm{FP}(f) - \alpha, 0\} + \mathrm{FN}(f), \tag{5.1}$$

where α is the maximum FP allowed.

The NP-score is a weighted sum of errors, thus the lower the better. It penalizes heavily FP exceeding α, since it is multiplied by $1/\alpha$. However, if the FP is only very slightly over α, it may still define a useful classifier if the FN is sufficiently small. Finally, if FP is below α the NP-score is the FN rate. Scott [32] shows that this measure satisfies some intuitive understanding of how to combine the requirements of a Neyman-Pearson classification problem. For instance, it should not discard altogether a classifier that disrespects the false positive constraint. Instead, the NP-score applies a penalty to classifiers that violates α. This penalty depends on the value of α and becomes more rigorous as α gets closer to zero, i.e., if the classifier violates α by 0.1 its penalty is bigger than when it violates alpha by 0.01.

5.2 Datasets

In order to evaluate our approach in different scenarios under different conditions, we performed experiments on several binary datasets, from different sources and sizes. For

this, we selected some of the datasets published in the LIBSVM [19] website, since it contains many datasets that are commonly used in literature for binary classification. We separated them into two groups, according to their size: small, and large. These datasets are summarized on Tables 5.1, and 5.2.

Table 5.1: Group of small datasets used in our experiments. Size is the amount of data in the dataset, Pos and Neg refer to the proportion of positive and negative examples, respectively, and d is the number of features on the dataset.

Dataset	Size	%Pos	%Neg	d
australian	690	44.5	55.5	14
breast-cancer	683	35.0	65.0	10
colon-cancer	62	35.5	64.5	2,000
diabetes	768	65.1	34.9	8
duke.breast-cancer	44	52.3	47.7	7,129
fourclass	862	35.6	64.4	2
german.numer	1,000	30.0	70.0	24
heart	270	44.4	55.6	13
ionosphere	351	64.1	35.9	34
leukemia	72	65.3	34.7	7,129
liver-disorders	345	42.0	58.0	6
mushrooms	8,124	48.2	51.8	112
sonar	208	46.6	53.4	60
splice	3,175	51.9	48.1	60
svmguide1	7,089	56.4	43.6	4
svmguide3	1,284	26.2	73.8	21

5.3 Experimental Setup

To evaluate the performance of the classifiers in the group of small datasets we used the 5×2 cross-validation protocol [13]. This approach consists in five replications of the standard 2-fold cross-validation protocol. It means that, in each replication, the dataset is randomly partitioned into two subsets S_1 and S_2 roughly of the same size. We then train the classifier on S_1 and test on S_2, followed by training on S_2 and testing on S_1. As discussed in [13], the 5×2 cross-validation provides a more precise estimation of the variance of the error (or in this case the NP-score) for different samples of the data in the dataset, and should be preferred to the more common method of using k-fold cross validation to measure and compare the quality of classifiers. One of the reasons for this is that the 5×2 cross-validation has fewer samples shared between the training subsets than

Table 5.2: Group of large datasets used in our experiments. Train and Test are the amount of data in the dataset, Pos and Neg refer to the proportion of positive and negative examples, respectively, and d is the number of features on the dataset.

Dataset	Train	Test	%Pos	%Neg	d
a1a	1,605	30,956	24.1	75.9	123
a2a	2,265	30,296	24.1	75.9	123
a3a	3,185	29,376	24.1	75.9	123
a4a	4,781	27,780	24.1	75.9	123
a5a	6,414	26,147	24.1	75.9	123
a6a	11,220	21,341	24.1	75.9	123
a7a	16,100	16,461	24.1	75.9	123
a8a	22,696	9,865	24.1	75.9	123
news20.bin	15,996	4,000	50.0	50.0	1,355,191
w1a	2,477	47,272	3.00	97.0	300
w2a	3,470	46,279	3.00	97.0	300
w3a	4,912	44,837	3.00	97.0	300
w4a	7,366	42,383	3.00	97.0	300
w5a	9,888	39,861	3.00	97.0	300

the standard k-fold cross-validation. In the k-fold, the fraction of samples that is shared between two training subsets is given by $(1-2/k)$. If we have $k = 10$, 80% of the training data is shared between each pair of training subsets. As for the 5×2 cross-validation, the expected number of samples shared between two training subsets is only 50%. The 5×2 cross-validation protocol results in 10 measures of classification quality, in this case NP-score for each algorithm. To compare two algorithms, X and Y we compare the set of 10 measures times the number of datasets from one algorithm with the other, in different forms: we show the boxplot of each set of measures and also report the mean and standard deviation of the set of measures. In addition, we use the Wilcoxon signed rank test to verify if the differences between two sets of measures are statistically significant. The test is paired because the same split between train and test subsets is used for all algorithms.

For the group of large datasets, we opted to use the already existing (suggested) training and test splits provided in their documentation [19]. In addition, 5×2 validation protocol in such cases proved to be unfeasible due to the large amount of time required to perform the 10 steps of training. In this case, we have only one measure per algorithm for each dataset. We still perform the Wilcoxon paired test but there are less pairs of values and thus the p-values of the comparison are expected to be higher.

5.4 Comparisons

In our previous experiments (see Appendix A), we discovered that the OSSRA form of RASVM performed better on average than the other three versions, both regarding NP-score and FP rates. For simplicity, in this work, we only list the results for the RASVM and RASVM-SV with the risk area in the OSSRA form, and we will call them as OSSRA and OSSRA-SV, respectively. But the practitioner must be aware that for a particular dataset, one of the other three versions may achieve better results. Below, we summarize the procedures we used to compare OSSRA and OSSRA-SV to other techniques in the literature: BS, ASVM, and CS-SVM.

1. **Comparison with BS**. The BS strategy has three hyperparameters C, γ and t. We selected them by following the first two steps of the procedure described in Section 4.4.

2. **Comparison with ASVM**. ASVM has three hyperparameters μ, τ, and q [40], and we followed the procedure described in [40] to select them: μ and q are selected first through a grid-search, followed by a linear search on τ. We then select the combination (μ, q, and τ) that minimizes the NP-score, since this is the measure we are using to evaluate each method.

3. **Comparison with CS-SVM**. To compare the proposed methods with CS-SVM we used the same experimental procedure described in [10]: 100 permutations of the data, each one with a random split of 70% of the data for training and 30% for test. Then, we follow the three-step procedure described in Section 4.4 for each permutation, and average the results. We compare the obtained results with the ones reported in [10].

Chapter 6

Experiments and Results

In this chapter, we compare OSSRA and OSSRA-SV with other techniques in the literature: BS, ASVM, and CS-SVM. As we mentioned in Section 4.3, the OSSRA methods performed better and we only consider this form in this section to compare RASVM and RASVM-SV with the other strategies. In addition, for a clean presentation, we opted for showing results just in terms of NP-scores, false positives and true positives in this section. For other RASVM results as well as additional metrics to the ones reported in this chapter, please refer to Appendices A and B, respectively.

6.1 Comparison with BS and ASVM

We start comparing OSSRA and OSSRA-SV with BS and ASVM [40] strategies. Table 6.1 shows the NP-scores achieved for the group of small datasets when $\alpha = 0.10$ and $\alpha = 0.01$. We can see that both OSSRA and OSSRA-SV achieved the best (lower) result on most cases — 13 and 12 datasets (out of 16) when $\alpha = 0.1$ and $\alpha = 0.01$, respectively. BS achieved similar results on 10 cases (six with $\alpha = 0.1$ and four with $\alpha = 0.01$), while ASVM was better on seven cases (three with $\alpha = 0.1$ and four with $\alpha = 0.01$).

Table 6.2 shows the p-values of the Wilcoxon signed-rank paired test on the NP-scores of the BS and ASVM strategies when compared with the OSSRA and OSSRA-SV. Thus, for the values of α equal to 0.10 and 0.01 both OSSRA and OSSRA-SV have statistically signiticantly lower scores than BS and ASVM for the group of small datasets. For $\alpha = 0.05$, only the OSSRA has statistically significant lower scores.

Figures 6.1, 6.2, and 6.3 compare the results of ASVM, BS, OSSRA, and OSSRA-SV through a boxplot of NP-scores, TP, and FP, respectively, for the group of small datasets. The red boxes represent the results of $\alpha = 0.10$, the green boxes the results of $\alpha = 0.05$, and the yellow boxes the results of $\alpha = 0.01$. We can see that both OSSRA and OSSRA-SV achieved lower median values of NP-scores and FP than ASVM and BS for all values

Table 6.1: Neyman-Pearson scores of BS, OSSRA, and OSSRA-SV for the group of small datasets.

Dataset	α	BS	ASVM	OSSRA	OSSRA-SV
australian	.10	0.49 ±0.3	0.90 ±0.2	**0.41**±0.2	0.42 ±0.2
	.01	2.59 ±1.9	15.4 ±4.7	1.58 ±1.1	**1.23**±0.9
breast-cancer	.10	0.05 ±0.0	**0.01**±0.0	0.05 ±0.0	0.05 ±0.0
	.01	1.39 ±1.6	1.31 ±1.5	**0.73**±0.7	0.75 ±0.6
colon-cancer	.10	0.51 ±0.3	3.11 ±3.3	**0.45**±0.3	**0.45**±0.3
	.01	**7.13**±6.4	37.8 ±35	**7.13**±6.4	**7.13**±6.4
diabetes	.10	0.64 ±0.2	0.68 ±0.2	0.60 ±0.2	**0.58**±0.1
	.01	2.59 ±1.7	**1.42**±1.0	1.82 ±1.1	1.78 ±1.1
duke	.10	1.81 ±1.9	**1.00**±0.2	1.81 ±1.9	1.81 ±1.9
	.01	23.7 ±21	**7.66**±18	23.7 ±21	23.7 ±21
fourclass	.10	**0.00**±0.0	1.84 ±0.2	**0.00**±0.0	**0.00**±0.0
	.01	**0.00**±0.0	24.1 ±1.7	**0.00**±0.0	**0.00**±0.0
german.numer	.10	0.82 ±0.2	1.33 ±0.4	**0.79**±0.2	0.80 ±0.2
	.01	3.05 ±3.6	13.9 ±7.7	2.37 ±3.4	**2.08**±3.5
heart	.10	0.51 ±0.3	0.93 ±0.4	0.39 ±0.2	**0.34**±0.1
	.01	2.76 ±2.8	11.9 ±7.3	**1.99**±2.0	2.25 ±2.0
ionosphere	.10	0.40 ±0.6	0.60 ±0.3	**0.23**±0.5	**0.23**±0.5
	.01	4.21 ±3.1	**1.21**±0.8	4.21 ±3.1	4.21 ±3.1
leu	.10	1.53 ±1.3	**0.92**±0.2	1.53 ±1.3	1.53 ±1.3
	.01	21.6 ±17	**2.42**±5.0	21.6 ±17	21.6 ±17
liver-disorders	.10	0.91 ±0.4	1.09 ±0.3	**0.84**±0.3	0.85 ±0.2
	.01	4.81 ±5.0	10.3 ±3.5	3.47 ±4.4	**3.28**±4.0
mushrooms	.10	**0.00**±0.0	0.18 ±0.0	**0.00**±0.0	**0.00**±0.0
	.01	**0.00**±0.0	0.67 ±0.4	**0.00**±0.0	**0.00**±0.0
sonar	.10	**0.58**±0.7	2.28 ±2.4	**0.58**±0.7	**0.58**±0.7
	.01	**10.6**±8.6	18.8 ±30	**10.6**±8.6	**10.6**±8.6
splice	.10	**0.59**±0.3	1.32 ±0.4	**0.59**±0.3	**0.59**±0.3
	.01	13.1 ±3.0	17.9 ±7.8	**13.1**±3.0	**13.1**±3.0
svmguide1	.10	**0.03**±0.0	0.25 ±0.1	**0.03**±0.0	**0.03**±0.0
	.01	0.76 ±0.9	0.89 ±0.5	**0.23**±0.2	0.26 ±0.2
svmguide3	.10	**0.57**±0.1	1.26 ±0.5	**0.57**±0.1	**0.57**±0.1
	.01	3.43 ±1.5	12.7 ±6.7	**3.10**±1.5	3.16 ±1.4

Table 6.2: Wilcoxon signed-rank test p-values on the NP-scores of OSSRA and OSSRA-SV with BS and ASVM, on the group of small datasets.

	α	OSSRA	OSSRA-SV
BS	0.10	**0.002**	**0.004**
	0.05	**0.001**	0.140
	0.01	**0.008**	**0.000**
ASVM	0.10	**0.000**	**0.000**
	0.05	**0.000**	**0.000**
	0.01	**0.000**	**0.000**

of α. The TP of ASVM was also much lower than those achieved by OSSRA, and OSSRA-SV. The achieved TP of BS was practically the same of OSSRA, and OSSRA-SV, except for the case of $\alpha = 0.01$ where BS achieved a higher TP.

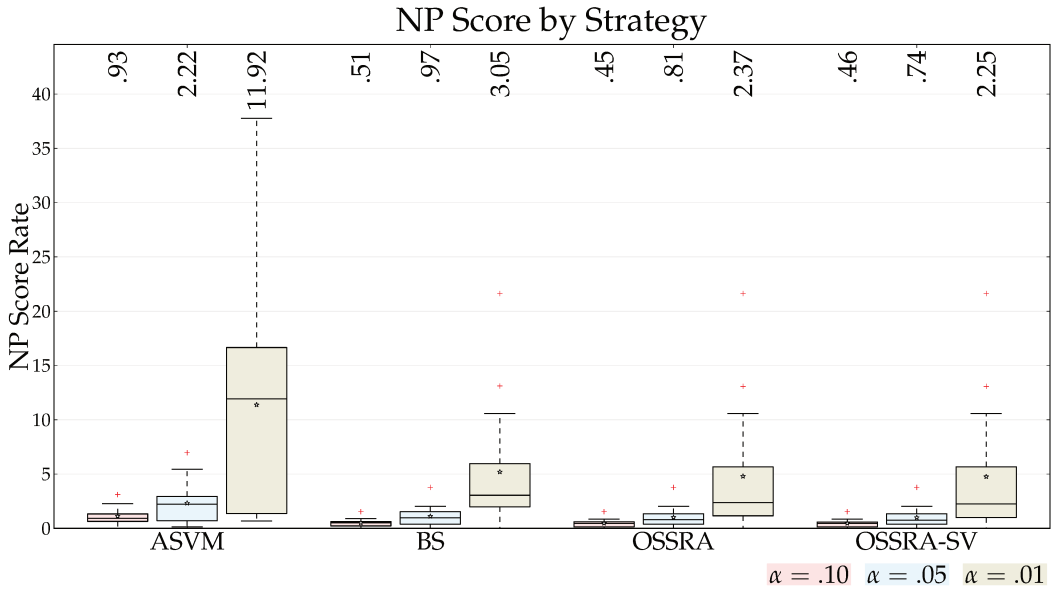


Figure 6.1: Comparison of the NP-scores between ASVM, BS, OSSRA, and OSSRA-SV for the group of small datasets. The median value of the NP-scores of each strategy is shown at the top of the figure and is marked by a *line* inside each box. The average value is marked by a star inside each box. The outliers were labeled with a red plus sign.

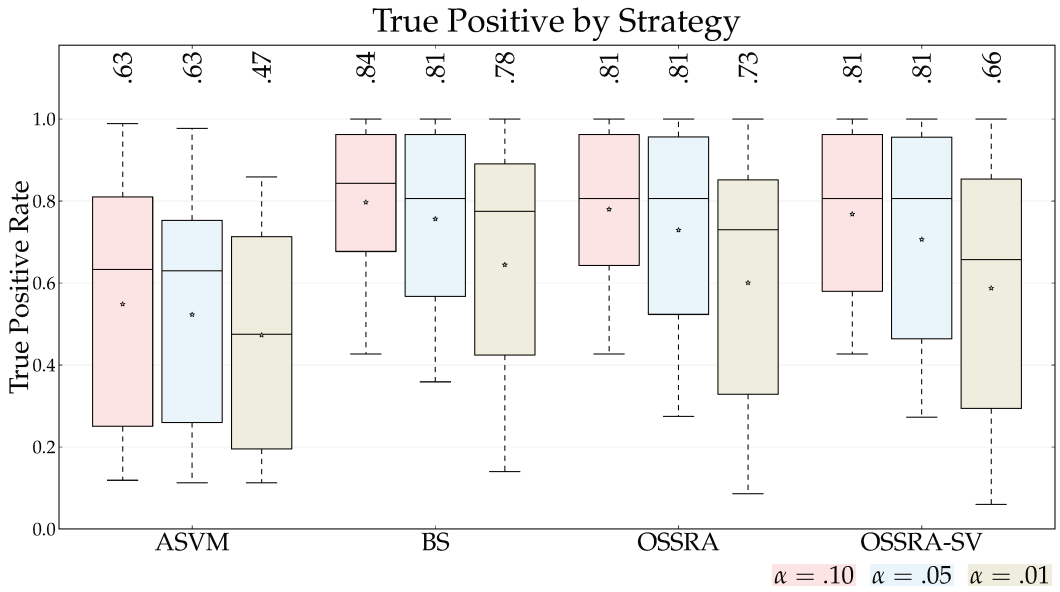


Figure 6.2: Comparison of the true positives between ASVM, BS, OSSRA, and OSSRA-SV for the group of small datasets.

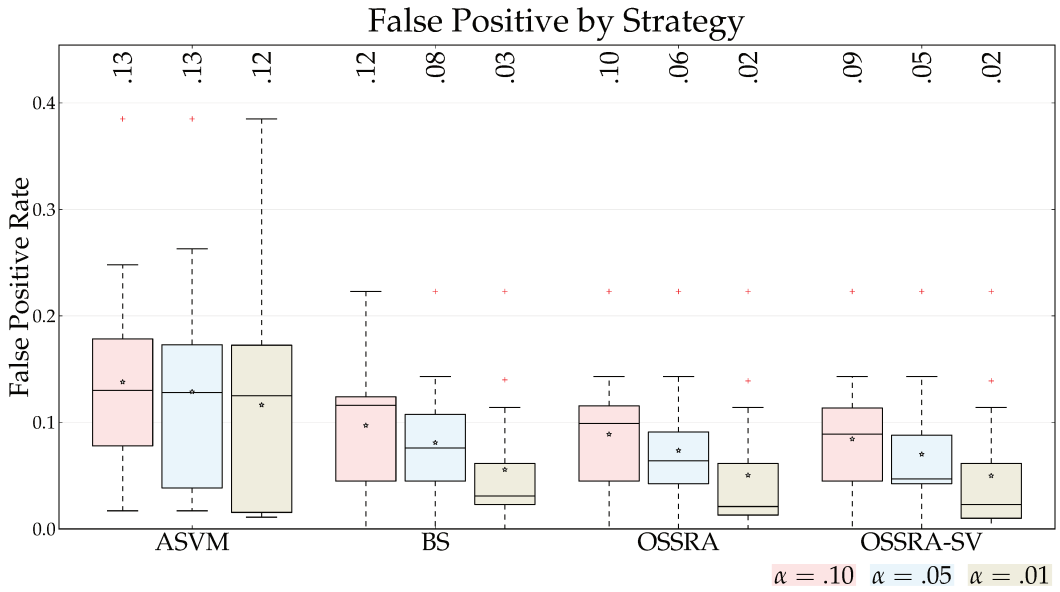


Figure 6.3: Comparison of the false positives between ASVM, BS, OSSRA, and OSSRA-SV for the group of small datasets.

6.2 Comparison with CS-SVM

This section compares the obtained results with the Cost-Sensitive methods proposed by Davenport et al. [10]. They consider the Cost-Sensitive 2ν-SVM formulation, with four different approaches for selecting the parameters ν_+ and ν_-: grid search (GS), windowed grid search (WGS), coordinate descent (CD), and windowed coordinate descent (WCD). Table 6.3 compares the OSSRA and OSSRA-SV with their methods, using the results reported in [10]. We can see that both OSSRA and OSSRA-SV achieved lower FP on all the cases. Comparing the NP-score, OSSRA and OSSRA-SV achieves lower (better) results on three out of the four datasets that we considered in our tests.

6.3 Experiments on Large Datasets

This section shows the experiments considering the larger datasets. In these cases, we trained and tested just following the the training/test partitions proposed in [19].

Table 6.5 shows the average NP-scores achieved by ASVM, BS, OSSRA, and OSSRA-SV on the 14 large datasets that we considered in our experiments[1]. Note how both OSSRA and OSSRA-SV achieves the lowest scores on almost all the cases (all the 14 datasets with $\alpha = 0.1$ and 13 with $\alpha = 0.01$), followed by BS which obtains a similar result in 16 cases (11 with $\alpha = 0.1$ and five with $\alpha = 0.01$). The ASVM obtained worse results in almost all the cases when compared with BS, OSSRA, or OSSRA-SV, except for the news20.bin dataset with $\alpha = 0.01$.

Table 6.4 shows the p-values of the Wilcoxon signed-rank paired test on the NP-scores of the BS and ASVM strategies when compared with the OSSRA and OSSRA-SV. For the values of α equal to 0.05 and 0.01 both OSSRA and OSSRA-SV have statistically significant lower scores than BS and ASVM for the group of large datasets.

6.4 Experiments with Unbalanced Data

An important feature for any classifier is the insensitivity to unbalance data. It is very common to have problems wherein the acquisition of samples from one of the classes is much more costly than in the other class, thereby resulting in unbalanced data. In this section we compare the ability of RASVM for controlling false positives on balanced and unbalanced data with the C-SVM, BS, and ASVM methods.

The experiments below have been made on the small group of datasets. For each dataset in this small group, we randomly generated 9 splits of data, ranging from 10%

[1]We could not evaluate the CS-SVM methods proposed by Davenport et al. [10] on those datasets, since they were not considered in their tests.

Table 6.3: Neyman-Pearson scores of OSSRA, OSSRA-SV, and the CS-SVM methods proposed by Davenport et al. [10].

Dataset	Classifier	FP	FN	NP
ida.banana	OSSRA	.058 ±.01	.142 ±.02	**0.142**
	OSSRA-SV	.058 ±.01	.142 ±.02	**0.142**
	GS	.114 ±.03	.120 ±.02	0.260
	WGS	.104 ±.02	.124 ±.02	0.164
	CD	.104 ±.02	.125 ±.02	0.165
	WCD	.106 ±.03	.124 ±.02	0.184
ida.breast	OSSRA	.056 ±.05	.793 ±.10	**0.793**
	OSSRA-SV	.056 ±.05	.794 ±.10	0.794
	GS	.156 ±.09	.668 ±.10	1.228
	WGS	.112 ±.06	.689 ±.10	0.809
	CD	.114 ±.06	.683 ±.10	0.823
	WCD	.119 ±.06	.678 ±.10	0.868
heart	OSSRA	.090 ±.05	.275 ±.09	0.275
	OSSRA-SV	.091 ±.05	.274 ±.08	**0.274**
	GS	.124 ±.06	.219 ±.07	0.459
	WGS	.113 ±.05	.231 ±.07	0.361
	CD	.106 ±.05	.230 ±.06	0.290
	WCD	.110 ±.05	.231 ±.06	0.331
ida.thyroid	OSSRA	.023 ±.02	.087 ±.08	0.087
	OSSRA-SV	.023 ±.02	.087 ±.08	0.087
	GS	.098 ±.09	.064 ±.09	0.064
	WGS	.087 ±.06	.032 ±.05	**0.032**
	CD	.084 ±.06	.039 ±.05	0.039
	WCD	.093 ±.06	.032 ±.05	**0.032**

Table 6.4: Wilcoxon signed-rank test p-values on the NP-scores of OSSRA and OSSRA-SV with BS and ASVM, on the group of large datasets.

	α	OSSRA	OSSRA-SV
BS	0.10	0.181	0.181
	0.05	**0.014**	**0.014**
	0.01	**0.014**	**0.014**
ASVM	0.10	**0.000**	**0.000**
	0.05	**0.000**	**0.000**
	0.01	**0.000**	**0.000**

to 90% of positives samples. We then followed the procedure described in Section 4.4 for each split and computed their corresponding NP-scores. Figures 6.4, 6.5, and 6.6 shows the NP-score achieved by each method on these splits for the values of α equal to 0.1, 0.05, and 0.01, respectively. The results showed are the average NP-score obtained from the datasets that we considered (group of small datasets). We could see that the NP-score of the A-SVM is the less sensitive to unbalanced data, since its variation between the splits is not large. The standard C-SVM, as expected, showed a very larger increase in the NP-score when the ratio positive samples is greater than 40%. This occurs because, when there are few negative samples, the C-SVM gives more importance to the positive class, thus increasing the ratio of incorrect classifications in the negative class. The BS and RASVM showed good results both with balanced and unbalanced data, but their NP-score significantly increases when the ratio of positive samples is greater than 70%.

6.5 Speed Improvement with RASVM-SV

As we mentioned in Section 4.5, the classification inside the *risk area* can be slow on large datasets. Given this issue, we consider the RASVM-SV, which can provide significant gains in speed.

We selected six datasets that we used in our experiments and measured the time spent by OSSRA and OSSRA-SV methods to optimize the parameters k and β and to classify all the testing data. All the experiments were executed on an Ubuntu machine with an 8-core Intel® Xeon® processor, and 16Gb of RAM. We compare the results on Table 6.6, with the number of training points (Train), the number of support vectors (SVs), and the training time. We can see that the gains in speed with the OSSRA-SV was very significant, up to five times faster than the OSSRA.

Table 6.5: Neyman-Pearson scores of BS, OSSRA, and OSSRA-SV for the group of large datasets.

Dataset	α	BS	ASVM	OSSRA	OSSRA-SV
a1a	.10	0.585	4.763	**0.513**	0.515
	.01	1.754	56.38	**0.983**	1.065
a2a	.10	1.228	3.410	0.965	**0.964**
	.01	1.746	41.82	1.523	**1.462**
a3a	.10	**0.403**	9.000	**0.403**	**0.403**
	.01	1.219	99.00	0.953	**0.831**
a4a	.10	**0.407**	2.343	**0.407**	**0.407**
	.01	1.170	30.76	0.859	**0.843**
a5a	.10	**0.410**	2.014	**0.410**	**0.410**
	.01	0.963	27.19	**0.803**	0.831
a6a	.10	**0.426**	2.015	**0.426**	**0.426**
	.01	1.203	27.08	0.957	**0.811**
a7a	.10	0.353	2.001	0.347	**0.333**
	.01	0.863	27.00	**0.772**	0.779
a8a	.10	**0.391**	1.845	**0.391**	**0.391**
	.01	1.183	25.30	0.937	**0.883**
news20.binary	.10	**0.027**	0.991	**0.027**	**0.027**
	.01	2.778	**0.998**	2.778	2.778
w1a	.10	**0.571**	2.165	**0.571**	**0.571**
	.01	**0.571**	23.10	**0.571**	**0.571**
w2a	.10	**0.449**	4.414	**0.449**	**0.449**
	.01	**0.449**	48.04	**0.449**	**0.449**
w3a	.10	**0.440**	3.781	**0.440**	**0.440**
	.01	**0.440**	41.00	**0.440**	**0.440**
w4a	.10	**0.437**	3.797	**0.437**	**0.437**
	.01	**0.437**	41.22	**0.437**	**0.437**
w5a	.10	**0.371**	3.742	**0.371**	**0.371**
	.01	**0.371**	40.52	**0.371**	**0.371**

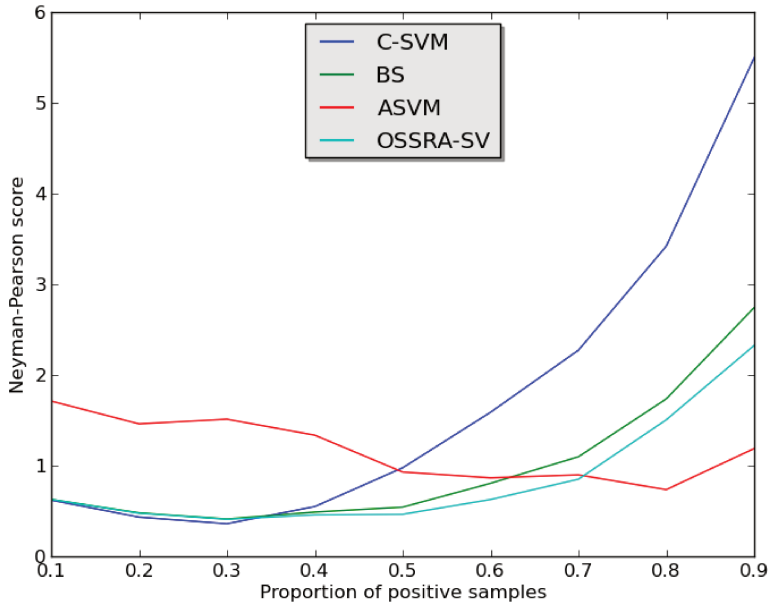


Figure 6.4: Sensitivity to unbalanced data between C-SVM, BS, ASVM, and OSSRA-SV methods with $\alpha = 0.1$.

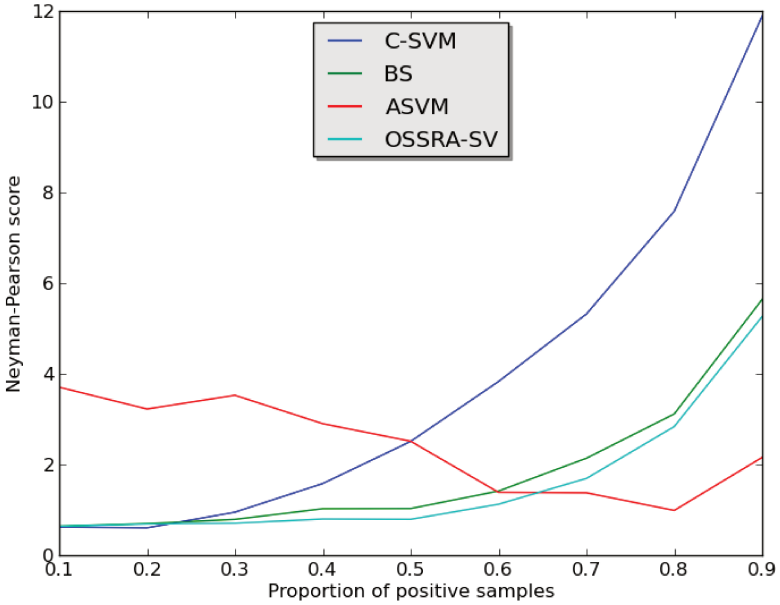


Figure 6.5: Sensitivity to unbalanced data between C-SVM, BS, ASVM, and OSSRA-SV methods with $\alpha = 0.05$.

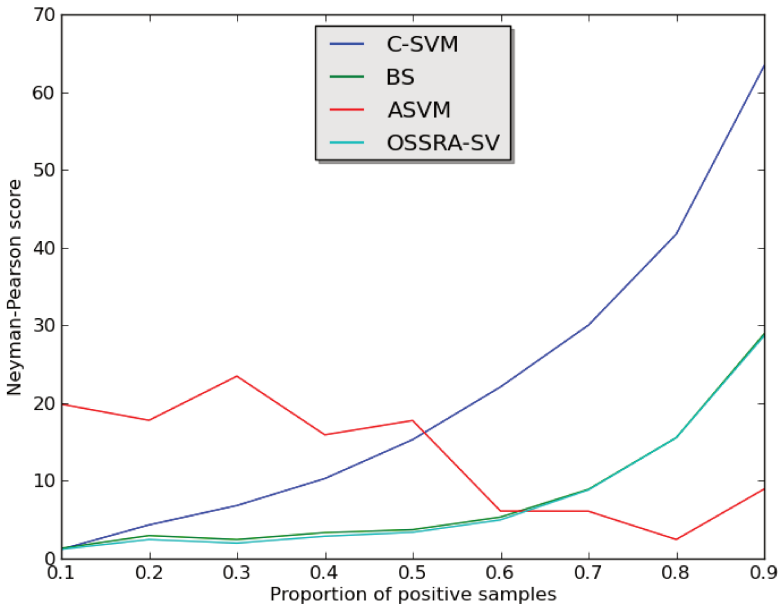


Figure 6.6: Sensitivity to unbalanced data between C-SVM, BS, ASVM, and OSSRA-SV methods with $\alpha = 0.01$.

Table 6.6: Comparison between OSSRA and OSSRA-SV on the time spent to optimize the parameters k and β and to classify all the testing data. Train refers to the number of training points on the dataset, and SVs to the number of support vectors.

Dataset	Train	SVs	OSSRA	OSSRA-SV	% Faster
a1a	1605	579	07h 08m 31s	**02h 28m 39s**	189
australian	552	176	01m 03s	**20s**	275
german.numer	800	434	02m 15s	**01m 23s**	62
heart	216	88	17s	**09s**	88
sonar	167	129	28s	**12s**	133
svmguide1	3089	368	12m 44s	**02m 20s**	446

Chapter 7

Conclusions

Controlling false positives is paramount in several machine learning problems varying from simple spam filtering to more complex computer-aided diagnosis solution. In this work, we have proposed a new method for controlling false alarms for one of the most powerful classifier to date: the Support Vector Machine.

Our approach was mainly based on two pressupositions: (1) most misclassified points on SVMs are close to the decision boundary; (2) these misclassified training points define sensitive areas, so that classifying a testing point as positive in these regions should be avoided in the context of low false positive classification. It is based on selecting a *risk area* around the SVM's decision hyperplane in order to outline the samples that will be classified through a second classifier later on. We discussed four variations for selecting the *risk area*, and evaluated the two best ones against state-of-the-art methods for controlling false positives.

In line with the SVM literature [14] which states that the most important decisions are always around the SVM boundary, the idea of further refining the results by imposing a unanimity decision-making process with a second layer classifier showed to be very effective. Indeed, one can further explore smoother decisions using a majority voting scheme rather than unanimous scheme therefore favoring either false positive or false negative controlling. The additional observation that the support vectors are enough to perform such second layer analysis is also very interesting leading to the proposition of a solution for increasing the RASVM and its OSSRA (One Sided Risk Area) variation performance on large datasets with a speed up to 5 times the one of the original formulation. With these two solutions (*risk area* analysis using only a pre-specific set of support vector points), we offer an effective and efficient methodology for controlling false positives on a variety of problems involving machine learning.

This research also goes in the direction of recent efforts in the machine learning community tackling the problem of open set recognition [12, 28]. Recognition problems,

differently from classification, consider only a fixed set of known classes for training while the testing can face a myriad of unseen examples from either previously trained classes but also from untrained ones. For instance, a biometric system trained with 100 people must reject all other identities not in the gallery of 100 people while in operation. In this new scenario, techniques such as the ones we propose in this work can play a major role since it will protect the classes seen during training while avoiding unknown samples which would be classified as false positives by a traditional classifier. In this context, a whole new research branch opens for exploring RASVM-based methods for openset recognition problems.

Finally, a possible drawback of our approach to be addressed in the future is the need for tuning the hyparameter β through a grid-search along with k. Since β only defines the size of the *risk area*, it may be possible to optimize it in an independent manner and make the RASVM optimization much faster. Further research is also needed in order to evaluate other classification methods for the samples on the *risk area*.

Bibliography

[1] Amaury B. Andre, Eduardo Beltrame, and Jacques Wainer. A combination of support vector machine and k-nearest neighbors for machine fault detection. *Applied Artificial Intelligence*, 27(1):36–49, 2013.

[2] Ion Androutsopoulos, John Koutsias, Konstantinos V Chandrinos, and Constantine D Spyropoulos. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–167. ACM, 2000.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 1 edition, 2006.

[4] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[5] Andrej Bratko, Bogdan Filipič, Gordon V. Cormack, Thomas R. Lynam, and Blaž Zupan. Spam filtering using statistical data compression models. *Machine Learning Research*, 7:2673–2698, 2006.

[6] Xavier Carreras and Lluís Màrquez. Boosting trees for anti-spam email filtering. *CoRR*, cs.CL/0109015, 2001.

[7] Chih-Chung Chang and Chih-Jen Lin. Training v-support vector regression: theory and algorithms. *Neural Computation*, 14(8):1959–1977, 2002.

[8] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[9] Filipe de Oliveira Costa, Michael Eckmann, Walter J. Scheirer, and Anderson Rocha. Open set source camera attribution. In *25th Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 71–78, 2012.

[10] Mark A Davenport, Richard G Baraniuk, and Clayton D Scott. Controlling false alarms with support vector machines. In *Acoustics, Speech and Signal Processing, 2006. ICASSP'2006 Proceedings. IEEE International Conference on*, pages 589–592. IEEE, 2006.

[11] Mark A Davenport, Richard G Baraniuk, and Clayton D Scott. Tuning support vector machines for minimax and Neyman-Pearson classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(10):1888–1898, 2010.

[12] Filipe de O. Costa, Ewerton Silva, Michael Eckmann, Walter Scheirer, and Anderson Rocha. Open set source camera attribution and device linking. *Elsevier Pattern Recognition Letters*, 39:92–101, 2014.

[13] Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923, 1998.

[14] Trevor. Hastie, Robert. Tibshirani, and J Jerome H Friedman. *The elements of statistical learning*. Springer New York, second edition, 2009.

[15] Douglas M Hawkins. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12, 2004.

[16] Grigoris Karakoulas and John Shawe-Taylor. Optimizing classifiers for imbalanced training sets. *Advances in Neural Information Processing Systems 11: Proceedings of the 1998 Conference*, 11:253, 1999.

[17] Aleksander Kołcz and Joshua Alspector. SVM-based filtering of e-mail spam with content-specific misclassification costs. In *Proceedings of the Workshop on Text Mining. TEXTDM'2001.*, 2001.

[18] Yu-Feng Li, James T Kwok, and Zhi-Hua Zhou. Cost-sensitive semi-supervised support vector machine. In *AAAI Conference on Artificial Intelligence*, pages 500–505, 2010.

[19] Chih-Jen Lin. Libsvm – a library for support vector machines. Online at http://www.csie.ntu.edu.tw/ cjlin/libsvm/, May 2013.

[20] Thomas R Lynam, Gordon V Cormack, and David R Cheriton. On-line spam filter fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 123–130. ACM, 2006.

[21] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[22] Hamed Masnadi-Shirazi and Nuno Vasconcelos. Asymmetric boosting. In *Proceedings of the 24th international conference on Machine learning*, pages 609–619. ACM, 2007.

[23] Edgar E. Osuna, Robert Freund, and Federico Girosi. Support vector machines: training and applications. Technical Report A.I. Memo 1602, Massachusetts Institute of Technology (MIT), Cambridge, US, 1997.

[24] John C. Platt. Sequential minimal optimization: a fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research, 1998.

[25] Zhiquan Qi, Yingjie Tian, Yong Shi, and Xiaodan Yu. Cost-sensitive support vector machine for semi-supervised learning. *Procedia Computer Science*, 18:1684–1689, 2013.

[26] S. J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Prentice Hall, third edition, 2010.

[27] Hanan Samet. *Foundations of multidimensional and metric data structures*. Morgan Kaufmann., 2006.

[28] Walter J. Scheirer, Anderson Rocha, Archana Sapkota, and Terrance E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1757–1772, 2013.

[29] Karl-Michael Schneider. A comparison of event models for naive bayes anti-spam e-mail filtering. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-volume 1*, pages 307–314. Association for Computational Linguistics, 2003.

[30] Bernhard Schölkopf, Alex J Smola, Robert C Williamson, and Peter L Bartlett. New support vector algorithms. *Neural computation*, 12(5):1207–1245, 2000.

[31] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization and beyond*. the MIT Press, 2002.

[32] C. Scott. Performance measures for Neyman-Pearson classification. *IEEE Transactions on Information Theory*, 53:2852–2863, 2007.

[33] C. Scott and R. Nowak. A Neyman-Pearson approach to statistical learning. *IEEE Transactions on Information Theory*, 51(8):3806–3819, 2005.

[34] D. Sculley and Gabriel 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*, pages 415–422. ACM, 2007.

[35] J. Shawe-Taylor. Classification accuracy based on observed margin. *Algorithmica*, 22(1-2):157–172, 1998.

[36] John Shawe-Taylor and Nello Cristianini. Further results on the margin distribution. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, COLT '99, pages 278–285. ACM, 1999.

[37] V.N. Vapnik. *Statistical learning theory*. Adaptive and learning systems for signal processing, communications, and control. Wiley, 1998.

[38] Paul Viola and Michael Jones. Fast and robust classification using asymmetric adaboost and a detector cascade. *Advances in Neural Information Processing System*, 14:1311–1318, 2001.

[39] Jianxin Wu, Matthew D Mullin, and James M Rehg. Linear asymmetric classifier for cascade detectors. In *Proceedings of the 22nd international conference on Machine learning*, pages 988–995. ACM, 2005.

[40] Shan-Hung Wu, Keng-Pei Lin, Hao-Heng Chien, Chung-Min Chen, and Ming-Syan Chen. On generalizable low false-positive learning using asymmetric support vector machines. *Knowledge and Data Engineering, IEEE Transactions on*, 25(5):1083–1096, 2013.

[41] Wen-tau Yih, Joshua Goodman, and Geoff Hulten. Learning at low false positive rates. In *Proceedings of the third conference on email and anti-spam*, pages 1–8, 2006.

[42] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *Knowledge and Data Engineering, IEEE Transactions on*, 18(1):63–77, 2006.

Appendix A

Additional Results

In this appendix, we compare all the RASVM and RASVM-SV methods to other techniques in the literature: BS and ASVM.

A.1 Comparison of RASVM with BS and ASVM

Tables A.1, A.2, and A.3 show the NP-scores achieved for the group of small datasets for the values of α equal to 0.10, 0.05, and 0.01, respectively. We can see that the RASVM methods achieved the best scores more often than BS and ASVM (OSSRA for $\alpha = 0.1$, RA for $\alpha = 0.05$, and RA/SRA for $\alpha = 0.01$).

Table A.4 shows the p-values of the Wilcoxon signed-rank paired test on the NP-scores of the BS and ASVM strategies when compared to the RASVM methods. Thus, for the values of α equal to 0.05, and 0.01, all the RASVM methods have statistically significant lower scores than BS and ASVM for the group of small datasets. For $\alpha = 0.10$, only OSSRA has statistically significant lower scores when compared to BS and ASVM.

Figure A.1 summarizes those results through a boxplot of NP-scores. The red boxes represent the results of $\alpha = 0.10$, the green boxes the results of $\alpha = 0.05$, and the yellow boxes the results of $\alpha = 0.01$. We can see that RASVM achieved lower median values of NP-scores than BS and ASVM for all values of α (SRA/OSSRA for $\alpha = 0.1$, OSSRA for $\alpha = 0.05$, and RA for $\alpha = 0.01$).

A.2 Comparison of RASVM-SV with BS and ASVM

Tables A.5, A.6, and A.7 show the NP-scores achieved for the group of small datasets for the values of α equal to 0.10, 0.05, and 0.01, respectively. We can see that the RASVM-SV methods also achieved the best scores more often than BS and ASVM (OSSRA-SV for

Table A.1: NP-scores of BS, ASVM, and RASVM for the small datasets ($\alpha = 0.1$).

Dataset	BS	ASVM	RA	OSRA	SRA	OSSRA
australian	0.49 ±0.3	0.90 ±0.2	0.42 ±0.2	0.43 ±0.2	0.42 ±0.2	**0.41**±0.2
breast-cancer	0.05 ±0.0	**0.01**±0.0	0.04 ±0.0	0.04 ±0.0	0.05 ±0.0	0.05 ±0.0
colon-cancer	0.51 ±0.3	3.11 ±3.3	**0.45**±0.3	**0.45**±0.3	**0.45**±0.3	**0.45**±0.3
diabetes	0.64 ±0.2	0.68 ±0.2	**0.57**±0.1	0.62 ±0.2	0.66 ±0.2	0.60 ±0.2
duke	1.81 ±1.9	**1.00**±0.2	1.81 ±1.9	1.81 ±1.9	1.81 ±1.9	1.81 ±1.9
fourclass	**0.00**±0.0	1.84 ±0.2	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0
german.numer	0.82 ±0.2	1.33 ±0.4	0.83 ±0.3	0.80 ±0.2	**0.79**±0.2	**0.79**±0.2
heart	0.51 ±0.3	0.93 ±0.4	0.53 ±0.3	0.54 ±0.2	0.41 ±0.1	**0.39**±0.2
ionosphere	0.40 ±0.6	0.60 ±0.3	0.24 ±0.5	**0.23**±0.5	0.24 ±0.5	**0.23**±0.5
leu	1.53 ±1.3	**0.92**±0.2	1.53 ±1.3	1.53 ±1.3	1.53 ±1.3	1.53 ±1.3
liver-disorders	0.91 ±0.4	1.09 ±0.3	0.90 ±0.3	0.91 ±0.2	0.88 ±0.3	**0.84**±0.3
mushrooms	**0.00**±0.0	0.18 ±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0
sonar	**0.58**±0.7	2.28 ±2.4	**0.58**±0.7	**0.58**±0.7	**0.58**±0.7	**0.58**±0.7
splice	**0.59**±0.3	1.32 ±0.4	**0.59**±0.3	**0.59**±0.3	**0.59**±0.3	**0.59**±0.3
svmguide1	**0.03**±0.0	0.25 ±0.1	**0.03**±0.0	**0.03**±0.0	**0.03**±0.0	**0.03**±0.0
svmguide3	**0.57**±0.1	1.26 ±0.5	**0.57**±0.1	**0.57**±0.1	**0.57**±0.1	**0.57**±0.1

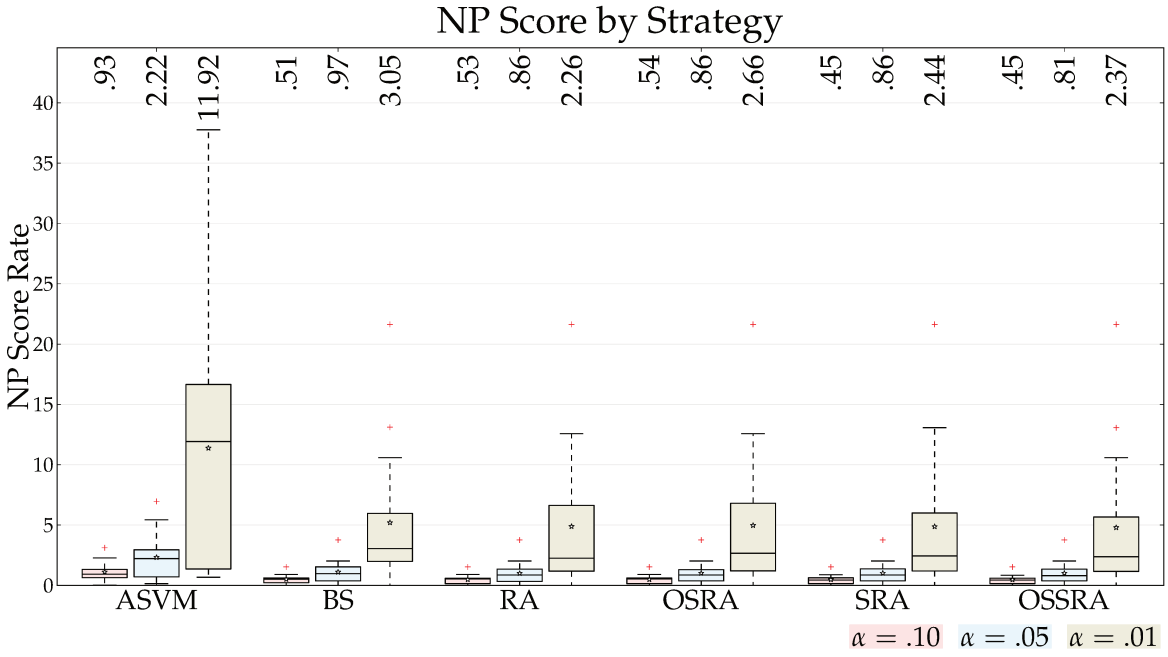


Figure A.1: Comparison of the NP-scores between ASVM, BS, and RASVM for the small datasets.

Table A.2: NP-scores of BS, ASVM, and RASVM for the small datasets ($\alpha = 0.05$).

Dataset	BS	ASVM	RA	OSRA	SRA	OSSRA
australian	0.76 ±0.5	2.44 ±0.8	**0.63**±0.4	**0.63**±0.3	0.66 ±0.5	0.70 ±0.4
breast-cancer	**0.05**±0.0	0.14 ±0.2	**0.05**±0.0	**0.05**±0.0	**0.05**±0.0	**0.05**±0.0
colon-cancer	1.55 ±1.0	6.96 ±6.7	1.28 ±1.0	**1.18**±1.0	**1.18**±1.0	1.20 ±1.0
diabetes	0.97 ±0.4	0.99 ±0.4	0.86 ±0.5	0.86 ±0.5	0.86 ±0.4	**0.81**±0.3
duke	4.24 ±4.0	**2.06**±3.2	4.24 ±4.0	4.24 ±4.0	4.24 ±4.0	4.24 ±4.0
fourclass	**0.00**±0.0	4.33 ±0.3	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0
german.numer	1.53 ±0.6	2.73 ±1.2	1.19 ±0.5	**1.14**±0.5	1.23 ±0.5	**1.14**±0.4
heart	0.96 ±0.6	2.22 ±1.1	**0.54**±0.2	0.72 ±0.5	0.67 ±0.4	0.69 ±0.3
ionosphere	0.99 ±1.1	**0.42**±0.0	0.93 ±1.2	1.02 ±1.1	0.92 ±1.2	0.97 ±1.1
leu	3.77 ±3.1	**1.09**±0.7	3.77 ±3.1	3.77 ±3.1	3.77 ±3.1	3.77 ±3.1
liver-disorders	1.69 ±0.9	1.97 ±0.7	**1.42**±0.9	1.43 ±0.9	1.55 ±0.9	1.51 ±0.9
mushrooms	**0.00**±0.0	0.20 ±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0
sonar	**1.50**±1.7	5.44 ±6.3	**1.50**±1.7	**1.50**±1.7	**1.50**±1.7	**1.50**±1.7
splice	**2.02**±0.6	3.17 ±1.2	**2.02**±0.6	**2.02**±0.6	**2.02**±0.6	**2.02**±0.6
svmguide1	**0.11**±0.1	0.29 ±0.0	**0.11**±0.1	0.12 ±0.1	**0.11**±0.1	0.12 ±0.1
svmguide3	**0.64**±0.1	2.43 ±1.3	0.66 ±0.1	0.66 ±0.1	0.65 ±0.1	**0.64**±0.1

$\alpha = 0.1$, OSSRA-SV for $\alpha = 0.05$, and SRA-SV/OSSRA-SV for $\alpha = 0.01$).

Table A.8 shows the p-values of the Wilcoxon signed-rank paired test on the NP-scores of the BS and ASVM strategies when compared to the RASVM-SV methods. Thus, for the values of α equal to 0.05, and 0.01, all the RASVM-SV methods have statistically significant lower scores than BS and ASVM for the group of small datasets. For $\alpha = 0.10$, only OSSRA-SV has statistically significant lower scores when compared to BS and ASVM.

Figure A.2 summarizes those results through a boxplot of NP-scores. We can see that RASVM-SV achieved lower median values of NP-scores than BS and ASVM for all values of α (OSSRA-SV for $\alpha = 0.1$, OSSRA-SV for $\alpha = 0.05$, and RA-SV for $\alpha = 0.01$).

Table A.3: NP-scores of BS, ASVM, and RASVM for the small datasets ($\alpha = 0.01$).

Dataset	BS	ASVM	RA	OSRA	SRA	OSSRA
australian	2.59 ±1.9	15.4 ±4.7	1.65 ±1.2	1.65 ±1.2	1.70 ±1.0	**1.58**±1.1
breast-cancer	1.39 ±1.6	1.31 ±1.5	0.71 ±0.6	0.75 ±0.7	**0.69**±0.6	0.73 ±0.7
colon-cancer	**7.13**±6.4	37.8 ±35	8.88 ±5.7	8.38 ±5.9	**7.13**±6.4	**7.13**±6.4
diabetes	2.59 ±1.7	**1.42**±1.0	2.13 ±2.2	2.13 ±2.2	2.14 ±1.7	1.82 ±1.1
duke	23.7 ±21	**7.66**±18	23.7 ±21	23.7 ±21	23.7 ±21	23.7 ±21
fourclass	**0.00**±0.0	24.1 ±1.7	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0
german.numer	3.05 ±3.6	13.9 ±7.7	**2.26**±3.3	2.28 ±3.4	2.35 ±3.3	2.37 ±3.4
heart	2.76 ±2.8	11.9 ±7.3	2.14 ±1.9	2.66 ±2.6	2.44 ±1.9	**1.99**±2.0
ionosphere	4.21 ±3.1	**1.21**±0.8	4.38 ±2.9	5.24 ±2.5	4.87 ±2.9	4.21 ±3.1
leu	21.6 ±17	**2.42**±5.0	21.6 ±17	21.6 ±17	21.6 ±17	21.6 ±17
liver-disorders	4.81 ±5.0	10.3 ±3.5	**2.80**±3.1	3.18 ±3.1	3.26 ±3.2	3.47 ±4.4
mushrooms	**0.00**±0.0	0.67 ±0.4	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0
sonar	**10.6**±8.6	18.8 ±30	**10.6**±8.6	**10.6**±8.6	**10.6**±8.6	**10.6**±8.6
splice	13.1 ±3.0	17.9 ±7.8	**12.6**±3.8	**12.6**±3.8	13.1 ±3.0	13.1 ±3.0
svmguide1	0.76 ±0.9	0.89 ±0.5	**0.20**±0.1	0.21 ±0.1	**0.20**±0.1	0.23 ±0.2
svmguide3	3.43 ±1.5	12.7 ±6.7	3.36 ±1.7	3.32 ±1.5	**3.06**±1.4	3.10 ±1.5

Table A.4: Wilcoxon signed-rank test p-values on the NP-scores of RASVM methods with BS and ASVM, on the group of small datasets.

	α	RA	OSRA	SRA	OSSRA
BS	0.10	0.165	0.309	0.188	**0.001**
	0.05	**0.000**	**0.001**	**0.000**	**0.000**
	0.01	**0.002**	**0.005**	**0.001**	**0.000**
ASVM	0.10	**0.000**	**0.000**	**0.000**	**0.000**
	0.05	**0.000**	**0.000**	**0.000**	**0.000**
	0.01	**0.000**	**0.000**	**0.000**	**0.000**

Table A.5: NP-scores of BS, ASVM, and RASVM-SV for the small datasets ($\alpha = 0.1$).

Dataset	BS	ASVM	RA-SV	OSRA-SV	SRA-SV	OSSRA-SV
australian	0.49 ±0.3	0.90 ±0.2	0.46 ±0.3	0.46 ±0.3	**0.42**±0.2	**0.42**±0.2
breast-cancer	0.05 ±0.0	**0.01**±0.0	0.04 ±0.0	0.04 ±0.0	0.05 ±0.0	0.05 ±0.0
colon-cancer	0.51 ±0.3	3.11 ±3.3	0.50 ±0.4	**0.45**±0.3	0.50 ±0.4	**0.45**±0.3
diabetes	0.64 ±0.2	0.68 ±0.2	0.68 ±0.1	0.71 ±0.1	0.72 ±0.1	**0.58**±0.1
duke	1.81 ±1.9	**1.00**±0.2	1.81 ±1.9	1.81 ±1.9	1.81 ±1.9	1.81 ±1.9
fourclass	**0.00**±0.0	1.84 ±0.2	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0
german.numer	0.82 ±0.2	1.33 ±0.4	0.84 ±0.2	**0.80**±0.2	0.81 ±0.2	**0.80**±0.2
heart	0.51 ±0.3	0.93 ±0.4	0.44 ±0.2	0.49 ±0.2	0.36 ±0.1	**0.34**±0.1
ionosphere	0.40 ±0.6	0.60 ±0.3	**0.23**±0.5	**0.23**±0.5	**0.23**±0.5	**0.23**±0.5
leu	1.53 ±1.3	**0.92**±0.2	1.53 ±1.3	1.53 ±1.3	1.53 ±1.3	1.53 ±1.3
liver-disorders	0.91 ±0.4	1.09 ±0.3	0.94 ±0.3	0.94 ±0.3	0.87 ±0.3	**0.85**±0.2
mushrooms	**0.00**±0.0	0.18 ±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0
sonar	**0.58**±0.7	2.28 ±2.4	**0.58**±0.7	**0.58**±0.7	**0.58**±0.7	**0.58**±0.7
splice	**0.59**±0.3	1.32 ±0.4	**0.59**±0.3	**0.59**±0.3	**0.59**±0.3	**0.59**±0.3
svmguide1	**0.03**±0.0	0.25 ±0.1	**0.03**±0.0	**0.03**±0.0	**0.03**±0.0	**0.03**±0.0
svmguide3	0.57 ±0.1	1.26 ±0.5	**0.56**±0.1	0.57 ±0.1	0.57 ±0.1	0.57 ±0.1

Table A.6: NP-scores of BS, ASVM, and RASVM-SV for the small datasets ($\alpha = 0.05$).

Dataset	BS	ASVM	RA-SV	OSRA-SV	SRA-SV	OSSRA-SV
australian	0.76 ±0.5	2.44 ±0.8	**0.71**±0.3	**0.71**±0.3	0.76 ±0.3	0.74 ±0.3
breast-cancer	**0.05**±0.0	0.14 ±0.2	0.06 ±0.0	**0.05**±0.0	0.06 ±0.0	**0.05**±0.0
colon-cancer	1.55 ±1.0	6.96 ±6.7	1.69 ±1.2	1.19 ±1.0	**1.18**±1.0	1.20 ±1.0
diabetes	0.97 ±0.4	0.99 ±0.4	0.82 ±0.2	0.90 ±0.2	0.88 ±0.2	**0.74**±0.2
duke	4.24 ±4.0	**2.06**±3.2	4.24 ±4.0	4.24 ±4.0	4.24 ±4.0	4.24 ±4.0
fourclass	**0.00**±0.0	4.33 ±0.3	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0
german.numer	1.53 ±0.6	2.73 ±1.2	1.24 ±0.5	1.14 ±0.4	1.27 ±0.5	**1.12**±0.4
heart	0.96 ±0.6	2.22 ±1.1	**0.53**±0.2	0.63 ±0.4	0.79 ±0.4	0.67 ±0.2
ionosphere	0.99 ±1.1	**0.42**±0.0	0.89 ±1.2	0.89 ±1.2	0.89 ±1.2	0.90 ±1.2
leu	3.77 ±3.1	**1.09**±0.7	3.77 ±3.1	3.77 ±3.1	3.77 ±3.1	3.77 ±3.1
liver-disorders	1.69 ±0.9	1.97 ±0.7	1.47 ±0.9	**1.42**±0.8	1.51 ±0.9	1.50 ±0.9
mushrooms	**0.00**±0.0	0.20 ±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0
sonar	**1.50**±1.7	5.44 ±6.3	**1.50**±1.7	**1.50**±1.7	**1.50**±1.7	**1.50**±1.7
splice	**2.02**±0.6	3.17 ±1.2	**2.02**±0.6	**2.02**±0.6	**2.02**±0.6	**2.02**±0.6
svmguide1	**0.11**±0.1	0.29 ±0.0	**0.11**±0.1	0.12 ±0.1	**0.11**±0.1	0.12 ±0.1
svmguide3	**0.64**±0.1	2.43 ±1.3	0.69 ±0.1	0.66 ±0.1	0.66 ±0.1	**0.64**±0.1

Table A.7: NP-scores of BS, ASVM, and RASVM-SV for the small datasets ($\alpha = 0.01$).

Dataset	BS	ASVM	RA-SV	OSRA-SV	SRA-SV	OSSRA-SV
australian	2.59 ±1.9	15.4 ±4.7	1.24 ±1.1	1.24 ±1.1	1.27 ±0.9	**1.23**±0.9
breast-cancer	1.39 ±1.6	1.31 ±1.5	0.78 ±0.6	0.77 ±0.6	**0.72**±0.6	0.75 ±0.6
colon-cancer	**7.13**±6.4	37.8 ±35	10.9 ±6.5	8.39 ±5.9	**7.13**±6.4	**7.13**±6.4
diabetes	2.59 ±1.7	**1.42**±1.0	1.74 ±0.8	1.86 ±1.5	2.03 ±1.4	1.78 ±1.1
duke	23.7 ±21	**7.66**±18	23.7 ±21	23.7 ±21	23.7 ±21	23.7 ±21
fourclass	**0.00**±0.0	24.1 ±1.7	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0
german.numer	3.05 ±3.6	13.9 ±7.7	2.16 ±3.5	2.15 ±3.5	2.09 ±3.5	**2.08**±3.5
heart	2.76 ±2.8	11.9 ±7.3	**2.19**±1.9	2.32 ±1.9	2.27 ±1.8	2.25 ±2.0
ionosphere	4.21 ±3.1	**1.21**±0.8	4.26 ±3.1	4.45 ±2.9	4.44 ±2.9	4.21 ±3.1
leu	21.6 ±17	**2.42**±5.0	21.6 ±17	21.6 ±17	21.6 ±17	21.6 ±17
liver-disorders	4.81 ±5.0	10.3 ±3.5	2.70 ±2.6	**2.38**±2.7	2.85 ±2.5	3.28 ±4.0
mushrooms	**0.00**±0.0	0.67 ±0.4	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0	**0.00**±0.0
sonar	**10.6**±8.6	18.8 ±30	**10.6**±8.6	**10.6**±8.6	**10.6**±8.6	**10.6**±8.6
splice	13.1 ±3.0	17.9 ±7.8	12.9 ±3.3	**12.7**±3.5	13.1 ±3.0	13.1 ±3.0
svmguide1	0.76 ±0.9	0.89 ±0.5	0.28 ±0.2	0.29 ±0.2	**0.26**±0.1	**0.26**±0.2
svmguide3	3.43 ±1.5	12.7 ±6.7	3.29 ±1.5	3.27 ±1.5	**3.14**±1.4	3.16 ±1.4

Table A.8: Wilcoxon signed-rank test p-values on the NP-scores of RASVM-SV methods with BS and ASVM, on the group of small datasets.

	α	RA-SV	OSRA-SV	SRA-SV	OSSRA-SV
BS	0.10	0.774	0.880	0.079	**0.003**
	0.05	**0.002**	**0.000**	**0.007**	**0.000**
	0.01	**0.004**	**0.000**	**0.000**	**0.000**
ASVM	0.10	**0.000**	**0.000**	**0.000**	**0.000**
	0.05	**0.000**	**0.000**	**0.000**	**0.000**
	0.01	**0.000**	**0.000**	**0.000**	**0.000**

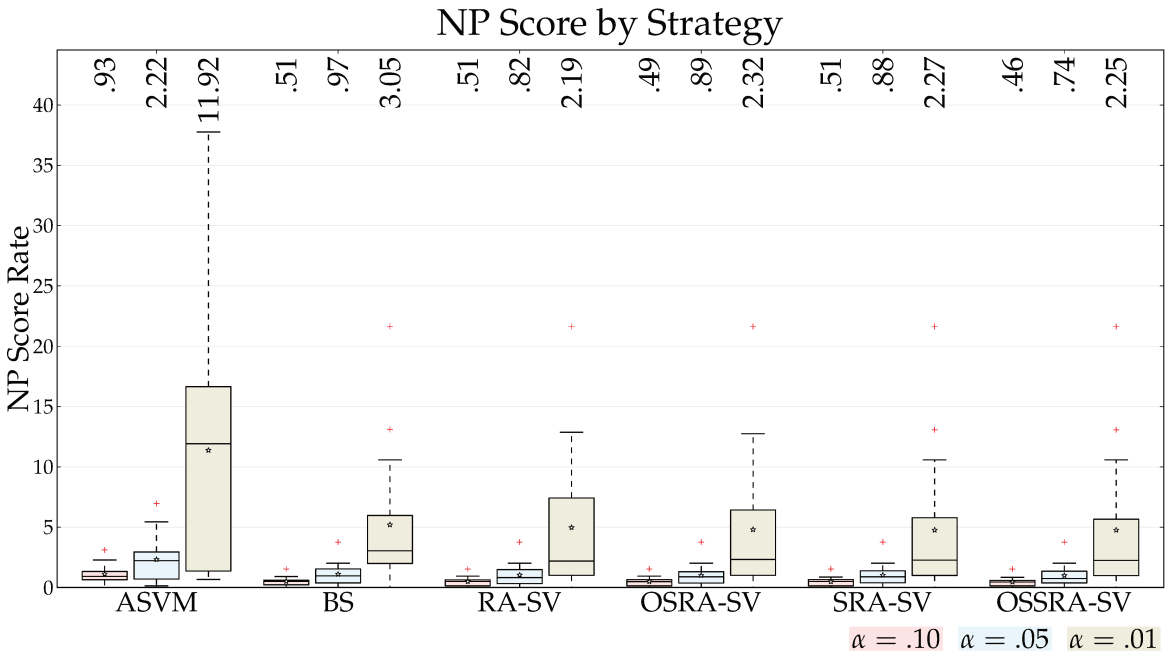


Figure A.2: Comparison of the NP-scores between ASVM, BS, and RASVM-SV for the small datasets.

Appendix B

Experiments with F1 score

In this appendix, we compare RASVM and RASVM-SV with BS and ASVM through the F1 score (also known as F-Measure)[1]. In this case, the training optimization function considers the F1 score instead of NP score when accounting for the max allowed α. The F1 score is defined as:

$$\frac{2 \times \text{tp}}{2 \times \text{tp} + \text{fn} + \text{fp}}, \tag{B.1}$$

where tp, fn, and fp are the number of true positives, false negatives, and false positives, respectively. To compare the strategies through the F1 score, as we mention above, we also adopted this measure to select the hyperparameters of the strategies considered in this appendix. The RASVM hyparameters are selected by the following procedures:

1. First, C and γ are selected in a grid-search fashion using a 5-fold validation and selecting for the pair of values with higher mean accuracy over the evaluation sets.

2. Second, δ is optimized by moving the hyperplane toward the sensitive class, until finding the position that maximizes the F1 score, subject to the FP $\leq \alpha$ constraint[2]. For the RA and OSRA, $\delta = 0$.

3. Finally, β and k are selected also through grid-search using a 5-fold protocol on the training set which also optimizes the F1 score, subject to the FP $\leq \alpha$ constraint[2].

The BS hyperparameters are selected by following the first two steps of the aforementioned procedure. To select the ASVM hyperparameters we also followed the procedure described in [40]. We then select the combination (μ, q, and τ) that maximizes the F1 score, subject to the FP $\leq \alpha$ constraint[2].

[1]It would not be fair to compare the results obtained by Davenport et al. [10] in this section, since they were not optimized for the F1 score.

[2]When any combination of the parameters respects α, the combination with the lowest FP is used.

B.1 Comparison of RASVM with BS and ASVM

Tables B.1, B.2, and B.3 show the F1 scores achieved for the group of small datasets for the values of α equal to 0.10, 0.05, and 0.01, respectively. The results that violates α by a factor of two (somewhat unacceptable) are highlighted in red. For the three values of α, RA and OSRA achieve the best F1 scores more often. However, the technique that violates α by a factor of 2 less frequently is the OSSRA (only 4 cases — out of 48 —, both of them with $\alpha = 0.01$).

Table B.1: F1 scores of BS, ASVM, and RASVM for the small datasets ($\alpha = 0.1$).

Dataset	BS	ASVM	RA	OSRA	SRA	OSSRA
australian	**0.83**±0.0	0.75 ±0.1	**0.83**±0.0	**0.83**±0.0	**0.83**±0.0	**0.83**±0.0
breast-cancer	0.93 ±0.1	**0.95**±0.0	**0.95**±0.0	**0.95**±0.0	0.93 ±0.1	0.93 ±0.1
colon-cancer	0.16 ±0.2	0.68 ±0.1	**0.72**±0.2	**0.72**±0.2	0.16 ±0.2	0.16 ±0.2
diabetes	**0.77**±0.0	0.56 ±0.0	0.70 ±0.0	0.70 ±0.0	0.70 ±0.0	0.72 ±0.0
duke	0.16 ±0.1	0.18 ±0.2	**0.82**±0.1	**0.82**±0.1	0.16 ±0.1	0.16 ±0.1
fourclass	0.97 ±0.0	0.62 ±0.0	**1.00**±0.0	**1.00**±0.0	0.97 ±0.0	0.97 ±0.0
german.numer	0.43 ±0.1	0.34 ±0.1	**0.51**±0.0	**0.51**±0.0	0.46 ±0.0	0.43 ±0.1
heart	0.78 ±0.0	0.72 ±0.1	**0.79**±0.0	0.78 ±0.0	0.78 ±0.0	0.78 ±0.0
ionosphere	0.88 ±0.1	0.74 ±0.0	**0.95**±0.0	**0.95**±0.0	0.88 ±0.1	0.88 ±0.1
leu	0.20 ±0.2	0.27 ±0.3	**0.93**±0.0	**0.93**±0.0	0.20 ±0.2	0.20 ±0.2
liver-disorders	0.55 ±0.1	0.20 ±0.1	0.57 ±0.1	**0.58**±0.1	0.55 ±0.1	0.55 ±0.1
mushrooms	0.96 ±0.0	0.88 ±0.0	**1.00**±0.0	**1.00**±0.0	0.96 ±0.0	0.96 ±0.0
sonar	0.42 ±0.3	0.20 ±0.3	**0.81**±0.0	**0.81**±0.0	0.42 ±0.3	0.42 ±0.3
splice	0.58 ±0.2	0.44 ±0.3	**0.86**±0.0	**0.86**±0.0	0.58 ±0.2	0.58 ±0.2
svmguide1	**0.97**±0.0	0.87 ±0.0	**0.97**±0.0	**0.97**±0.0	**0.97**±0.0	**0.97**±0.0
svmguide3	**0.52**±0.0	0.22 ±0.1	**0.52**±0.0	**0.52**±0.0	**0.52**±0.0	**0.52**±0.0

Table B.4 shows the p-values of the Wilcoxon signed-rank paired test on the F1 scores of the BS and ASVM strategies when compared to the RASVM methods. Thus, for all the values of α, only the RA and OSRA methods have statistically significant higher F1 scores than BS and ASVM for the group of small datasets.

Figure B.1 summarizes those results through a boxplot of F1 scores. We can see that RASVM achieved higher median values of F1 scores than BS and ASVM for all values of α (RA/OSRA for $\alpha = 0.1$, RA/OSRA for $\alpha = 0.05$, and RA for $\alpha = 0.01$).

Table B.2: F1 scores of BS, ASVM, and RASVM for the small datasets ($\alpha = 0.05$).

Dataset	BS	ASVM	RA	OSRA	SRA	OSSRA
	0.76 ±0.1	0.75 ±0.1	**0.78**±0.0	**0.78**±0.0	**0.78**±0.0	0.76 ±0.1
breast-cancer	0.93 ±0.1	**0.95**±0.0	**0.95**±0.0	**0.95**±0.0	0.93 ±0.1	0.93 ±0.1
colon-cancer	0.16 ±0.2	0.68 ±0.1	**0.72**±0.2	**0.72**±0.2	0.16 ±0.2	0.16 ±0.2
diabetes	**0.70**±0.0	0.36 ±0.1	0.60 ±0.0	0.60 ±0.1	0.60 ±0.0	0.63 ±0.0
duke	0.16 ±0.1	0.17 ±0.2	**0.82**±0.1	**0.82**±0.1	0.16 ±0.1	0.16 ±0.1
fourclass	0.97 ±0.0	0.62 ±0.0	**1.00**±0.0	**1.00**±0.0	0.97 ±0.0	0.97 ±0.0
german.numer	0.31 ±0.1	0.34 ±0.1	0.42 ±0.1	**0.44**±0.1	0.38 ±0.0	0.31 ±0.1
heart	0.73 ±0.1	0.71 ±0.1	0.73 ±0.1	**0.75**±0.0	0.74 ±0.0	0.73 ±0.1
ionosphere	0.88 ±0.1	0.74 ±0.0	**0.95**±0.0	**0.95**±0.0	0.88 ±0.1	0.88 ±0.1
leu	0.20 ±0.2	0.27 ±0.3	**0.93**±0.0	**0.93**±0.0	0.20 ±0.2	0.20 ±0.2
liver-disorders	0.44 ±0.1	0.16 ±0.1	0.47 ±0.1	**0.49**±0.1	0.45 ±0.1	0.44 ±0.1
mushrooms	0.96 ±0.0	0.88 ±0.0	**1.00**±0.0	**1.00**±0.0	0.96 ±0.0	0.96 ±0.0
sonar	0.42 ±0.3	0.20 ±0.3	**0.81**±0.0	**0.81**±0.0	0.41 ±0.3	0.42 ±0.3
splice	0.58 ±0.2	0.44 ±0.3	**0.86**±0.0	**0.86**±0.0	0.58 ±0.2	0.58 ±0.2
svmguide1	**0.97**±0.0	0.82 ±0.0	**0.97**±0.0	**0.97**±0.0	**0.97**±0.0	**0.97**±0.0
svmguide3	**0.52**±0.0	0.18 ±0.1	**0.52**±0.0	**0.52**±0.0	**0.52**±0.0	**0.52**±0.0

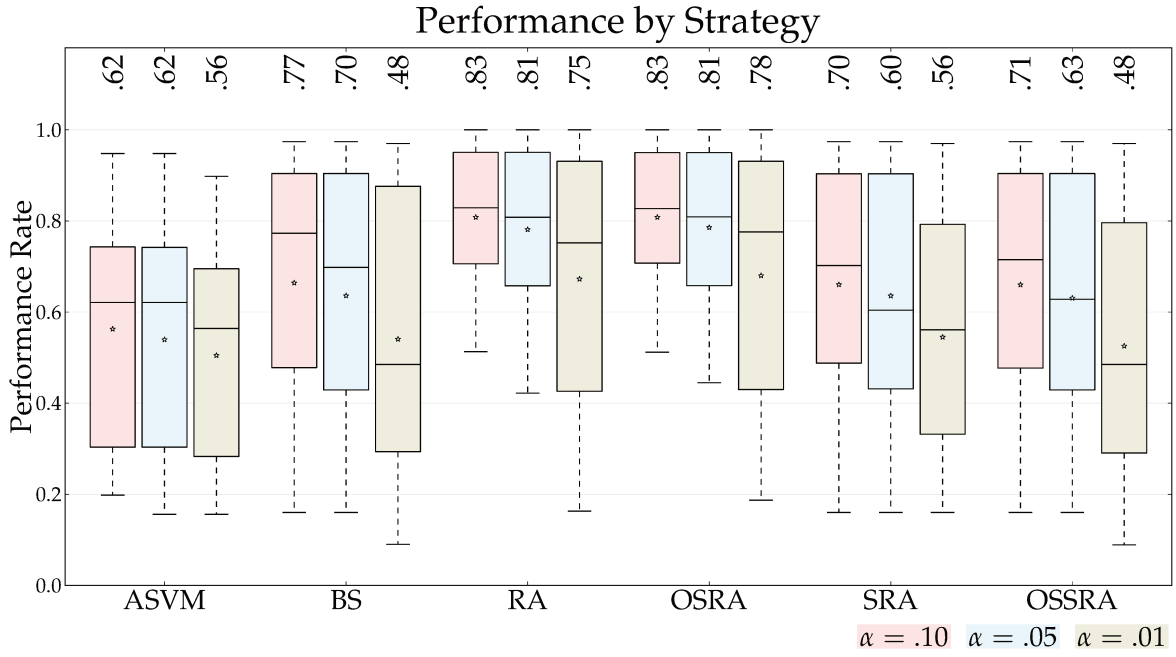


Figure B.1: Comparison of the F1 scores between ASVM, BS, and RASVM for the small datasets.

Table B.3: F1 scores of BS, ASVM, and RASVM for the small datasets ($\alpha = 0.01$).

Dataset	BS	ASVM	RA	OSRA	SRA	OSSRA
australian	0.56 ±0.1	**0.75**±0.1	0.36 ±0.2	0.36 ±0.2	0.56±0.1	0.56±0.1
breast-cancer	0.87 ±0.1	0.90 ±0.0	**0.93**±0.0	**0.93**±0.0	0.89±0.1	0.87±0.1
colon-cancer	0.16 ±0.2	0.68 ±0.1	**0.72**±0.2	**0.72**±0.2	0.16±0.2	0.16±0.2
diabetes	**0.43**±0.1	0.30 ±0.1	0.38 ±0.1	0.38 ±0.1	0.40±0.1	0.38±0.1
duke	0.16 ±0.1	0.17 ±0.2	**0.82**±0.1	**0.82**±0.1	0.16±0.1	0.16±0.1
fourclass	0.97 ±0.0	0.62 ±0.0	**1.00**±0.0	**1.00**±0.0	0.97±0.0	0.97±0.0
german.numer	0.09 ±0.0	**0.34**±0.1	0.16 ±0.1	0.19 ±0.0	0.17±0.1	0.09±0.0
heart	0.49 ±0.1	**0.71**±0.1	0.52 ±0.1	0.53 ±0.1	0.56±0.1	0.49±0.1
ionosphere	**0.88**±0.1	0.56 ±0.1	0.75 ±0.3	0.79 ±0.2	0.70±0.2	0.72±0.2
leu	0.20 ±0.2	0.27 ±0.3	**0.93**±0.0	**0.93**±0.0	0.20±0.2	0.20±0.2
liver-disorders	0.18 ±0.1	0.16 ±0.1	0.27 ±0.1	**0.31**±0.1	0.28±0.1	0.18±0.1
mushrooms	0.96 ±0.0	0.87 ±0.1	**1.00**±0.0	**1.00**±0.0	0.96±0.0	0.96±0.0
sonar	0.39 ±0.2	0.20 ±0.3	**0.78**±0.1	**0.78**±0.1	0.40±0.3	0.39±0.2
splice	0.58 ±0.2	0.44 ±0.3	**0.86**±0.0	**0.86**±0.0	0.58±0.2	0.58±0.2
svmguide1	**0.97**±0.0	0.61 ±0.1	0.95 ±0.0	0.95 ±0.0	0.95±0.0	0.95±0.0
svmguide3	0.38 ±0.1	0.18 ±0.1	0.47 ±0.1	**0.48**±0.1	0.39±0.1	0.38±0.1

Table B.4: Wilcoxon signed-rank test p-values on the F1 scores of RASVM methods with BS and ASVM, on the group of small datasets.

	α	RA	OSRA	SRA	OSSRA
BS	0.10	**0.000**	**0.000**	**0.049**	**0.001**
	0.05	**0.000**	**0.000**	0.883	**0.001**
	0.01	**0.000**	**0.000**	0.057	**0.000**
ASVM	0.10	**0.000**	**0.000**	**0.000**	**0.000**
	0.05	**0.000**	**0.000**	**0.000**	**0.000**
	0.01	**0.000**	**0.000**	**0.015**	0.133

B.2 Comparison of RASVM-SV with BS and ASVM

Tables B.5, B.6, and B.7 show the F1 scores achieved for the group of small datasets for the values of α equal to 0.10, 0.05, and 0.01, respectively. The results that violates α by a factor of two (somewhat unacceptable) are highlighted in red. For the three values of α, RA-SV and OSRA-SV achieve the best F1 scores more often. However, the technique that violate α by a factor of two less frequently is the OSSRA-SV (only two cases — out of 48 —, both of them with $\alpha = 0.01$).

Table B.5: F1 scores of BS, ASVM, and RASVM-SV for the small datasets ($\alpha = 0.1$).

Dataset	BS	ASVM	RA-SV	OSRA-SV	SRA-SV	OSSRA-SV
australian	**0.83**±0.0	0.75 ±0.1	0.75 ±0.2	0.75 ±0.2	**0.83**±0.0	**0.83**±0.0
breast-cancer	0.93 ±0.1	**0.95**±0.0	**0.95**±0.0	**0.95**±0.0	0.93 ±0.1	0.93 ±0.1
colon-cancer	0.16 ±0.2	0.68 ±0.1	**0.72**±0.2	**0.72**±0.2	0.16 ±0.2	0.16 ±0.2
diabetes	**0.77**±0.0	0.56 ±0.0	0.35 ±0.3	0.45 ±0.2	0.53 ±0.2	0.62 ±0.1
duke	0.16 ±0.1	0.18 ±0.2	**0.82**±0.1	**0.82**±0.1	0.16 ±0.1	0.16 ±0.1
fourclass	0.97 ±0.0	0.62 ±0.0	**1.00**±0.0	**1.00**±0.0	0.97 ±0.0	0.97 ±0.0
german.numer	0.43 ±0.1	0.34 ±0.1	**0.51**±0.0	**0.51**±0.1	0.45 ±0.1	0.43 ±0.1
heart	**0.78**±0.0	0.72 ±0.1	**0.78**±0.0	**0.78**±0.0	**0.78**±0.0	**0.78**±0.0
ionosphere	0.88 ±0.1	0.74 ±0.0	**0.95**±0.0	**0.95**±0.0	0.88 ±0.1	0.88 ±0.1
leu	0.20 ±0.2	0.27 ±0.3	**0.93**±0.0	**0.93**±0.0	0.20 ±0.2	0.20 ±0.2
liver-disorders	0.55 ±0.1	0.20 ±0.1	**0.58**±0.1	0.57 ±0.0	0.55 ±0.0	0.54 ±0.0
mushrooms	0.96 ±0.0	0.88 ±0.0	**1.00**±0.0	**1.00**±0.0	0.96 ±0.0	0.96 ±0.0
sonar	0.42 ±0.3	0.20 ±0.3	**0.81**±0.0	**0.81**±0.0	0.42 ±0.3	0.42 ±0.3
splice	0.58 ±0.2	0.44 ±0.3	**0.86**±0.0	**0.86**±0.0	0.58 ±0.2	0.58 ±0.2
svmguide1	**0.97**±0.0	0.87 ±0.0	**0.97**±0.0	**0.97**±0.0	**0.97**±0.0	**0.97**±0.0
svmguide3	0.52 ±0.0	0.22 ±0.1	**0.53**±0.0	0.52 ±0.0	0.52 ±0.0	0.52 ±0.0

Table B.8 shows the p-values of the Wilcoxon signed-rank paired test on the F1 scores of the BS and ASVM strategies when compared to the RASVM-SV methods. Thus, for all the values of α, only the RA-SV and OSRA-SV methods have statistically significant higher F1 scores than BS and ASVM for the group of small datasets.

Figure B.2 summarizes those results through a boxplot of F1 scores. We can see that RASVM-SV achieved higher median values of F1 scores than BS and ASVM for all values of α (RA-SV/OSRA-SV for $\alpha = 0.1$, OSRA-SV for $\alpha = 0.05$, and OSRA-SV for $\alpha = 0.01$).

Table B.6: F1 scores of BS, ASVM, and RASVM-SV for the small datasets ($\alpha = 0.05$).

Dataset	BS	ASVM	RA-SV	OSRA-SV	SRA-SV	OSSRA-SV
australian	**0.76**±0.1	0.75 ±0.1	0.55 ±0.3	0.57 ±0.3	0.74 ±0.1	0.66 ±0.2
breast-cancer	0.93 ±0.1	**0.95**±0.0	**0.95**±0.0	**0.95**±0.0	0.93 ±0.1	0.93 ±0.1
colon-cancer	0.16 ±0.2	0.68 ±0.1	**0.72**±0.2	**0.72**±0.2	0.16 ±0.2	0.16 ±0.2
diabetes	**0.70**±0.0	0.36 ±0.1	0.20 ±0.2	0.27 ±0.2	0.38 ±0.2	0.36 ±0.3
duke	0.16 ±0.1	0.17 ±0.2	**0.82**±0.1	**0.82**±0.1	0.16 ±0.1	0.16 ±0.1
fourclass	0.97 ±0.0	0.62 ±0.0	**1.00**±0.0	**1.00**±0.0	0.97 ±0.0	0.97 ±0.0
german.numer	0.31 ±0.1	0.34 ±0.1	0.42 ±0.1	**0.44**±0.1	0.37 ±0.0	0.31 ±0.1
heart	0.73 ±0.1	0.71 ±0.1	0.73 ±0.1	0.73 ±0.1	**0.74**±0.0	0.73 ±0.1
ionosphere	0.88 ±0.1	0.74 ±0.0	**0.95**±0.0	**0.95**±0.0	0.88 ±0.1	0.88 ±0.1
leu	0.20 ±0.2	0.27 ±0.3	**0.93**±0.0	**0.93**±0.0	0.20 ±0.2	0.20 ±0.2
liver-disorders	0.44 ±0.1	0.16 ±0.1	0.41 ±0.1	**0.46**±0.1	0.44 ±0.1	0.44 ±0.1
mushrooms	0.96 ±0.0	0.88 ±0.0	**1.00**±0.0	**1.00**±0.0	0.96 ±0.0	0.96 ±0.0
sonar	0.42 ±0.3	0.20 ±0.3	0.80 ±0.1	**0.81**±0.0	0.41 ±0.3	0.42 ±0.3
splice	0.58 ±0.2	0.44 ±0.3	**0.86**±0.0	**0.86**±0.0	0.58 ±0.2	0.58 ±0.2
svmguide1	**0.97**±0.0	0.82 ±0.0	**0.97**±0.0	**0.97**±0.0	**0.97**±0.0	**0.97**±0.0
svmguide3	0.52 ±0.0	0.18 ±0.1	**0.53**±0.0	0.52 ±0.0	0.51 ±0.0	0.52 ±0.0

Table B.7: F1 scores of BS, ASVM, and RASVM-SV for the small datasets ($\alpha = 0.01$).

Dataset	BS	ASVM	RA-SV	OSRA-SV	SRA-SV	OSSRA-SV
australian	0.56 ±0.1	**0.75**±0.1	0.19 ±0.1	0.18 ±0.1	0.35±0.2	0.34±0.2
breast-cancer	0.87 ±0.1	**0.90**±0.0	0.88 ±0.1	0.88 ±0.1	0.73±0.1	0.83±0.1
colon-cancer	0.16 ±0.2	0.68 ±0.1	**0.72**±0.2	**0.72**±0.2	0.16±0.2	0.16±0.2
diabetes	**0.43**±0.1	0.30 ±0.1	0.14 ±0.2	0.16 ±0.1	0.17±0.1	0.16±0.2
duke	0.16 ±0.1	0.17 ±0.2	**0.82**±0.1	**0.82**±0.1	0.16±0.1	0.16±0.1
fourclass	0.97 ±0.0	0.62 ±0.0	**1.00**±0.0	**1.00**±0.0	0.97±0.0	0.97±0.0
german.numer	0.09 ±0.0	**0.34**±0.1	0.07 ±0.0	0.11 ±0.1	0.14±0.1	0.08±0.0
heart	0.49 ±0.1	**0.71**±0.1	0.47 ±0.1	0.48 ±0.1	0.50±0.1	0.49±0.1
ionosphere	**0.88**±0.1	0.56 ±0.1	0.76 ±0.2	0.81 ±0.1	0.70±0.2	0.74±0.1
leu	0.20 ±0.2	0.27 ±0.3	**0.93**±0.0	**0.93**±0.0	0.20±0.2	0.20±0.2
liver-disorders	0.18 ±0.1	0.16 ±0.1	0.25 ±0.1	**0.29**±0.1	0.26±0.1	0.18±0.1
mushrooms	0.96 ±0.0	0.87 ±0.1	**1.00**±0.0	**1.00**±0.0	0.96±0.0	0.96±0.0
sonar	0.39 ±0.2	0.20 ±0.3	**0.78**±0.1	**0.78**±0.1	0.38±0.2	0.39±0.2
splice	0.58 ±0.2	0.44 ±0.3	**0.86**±0.0	**0.86**±0.0	0.58±0.2	0.58±0.2
svmguide1	**0.97**±0.0	0.61 ±0.1	0.85 ±0.1	0.88 ±0.1	0.90±0.0	0.92±0.0
svmguide3	0.38 ±0.1	0.18 ±0.1	**0.48**±0.1	**0.48**±0.1	0.41±0.1	0.35±0.1

Table B.8: Wilcoxon signed-rank test p-values on the F1 scores of RASVM-SV methods with BS and ASVM, on the group of small datasets.

	α	RA-SV	OSRA-SV	SRA-SV	OSSRA-SV
BS	0.10	**0.000**	**0.000**	**0.018**	**0.001**
	0.05	**0.000**	**0.000**	0.114	**0.000**
	0.01	**0.000**	**0.000**	**0.000**	**0.000**
ASVM	0.10	**0.000**	**0.000**	**0.000**	**0.000**
	0.05	**0.000**	**0.000**	**0.000**	**0.000**
	0.01	**0.000**	**0.000**	0.834	0.953

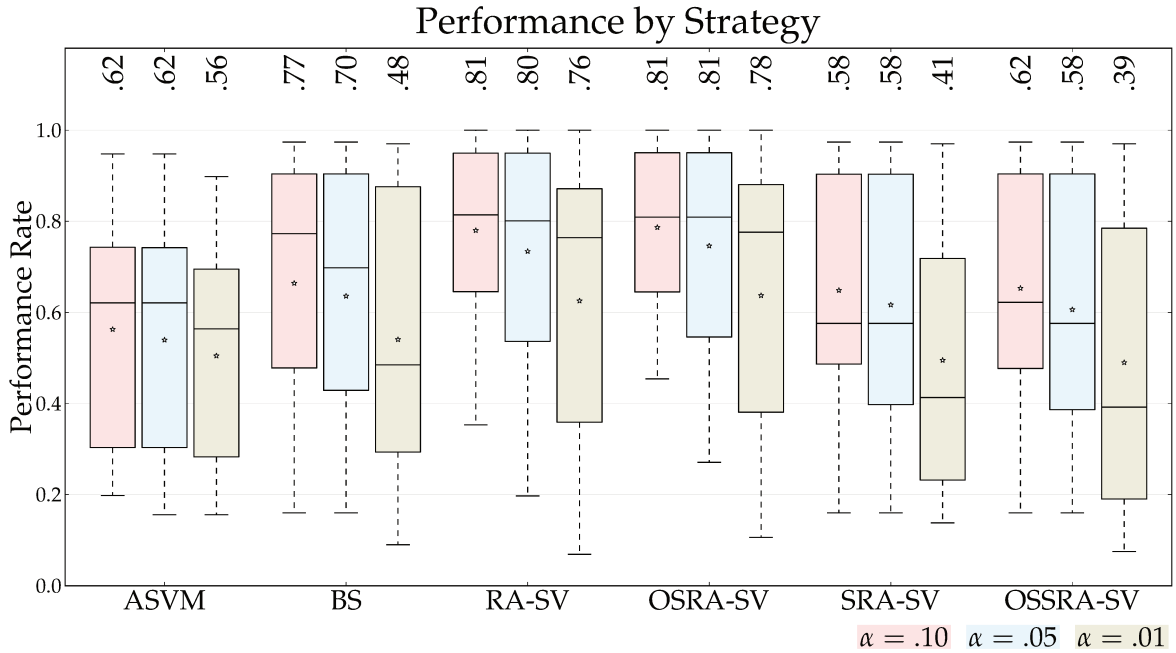


Figure B.2: Comparison of the F1 scores between ASVM, BS, and RASVM-SV for the small datasets.