A Note on Probability and Conditional Probability SVM models

Miguel Carrasco[a,*], Julio López[b], Benjamin Ivorra[c], Angel M. Ramos[c]

[a] *Universidad de los Andes, Chile, Facultad de Ingeniería y Ciencias Aplicadas. Monseñor Álvaro del Portillo 12455, Las Condes, Santiago, Chile. E-mail: micarrasco@uandes.cl*
[b] *Facultad de Ingeniería y Ciencias, Universidad Diego Portales, Av. Ejército 441, Santiago, Chile. E-mail: julio.lopez@udp.cl*
[c] *Interdisciplinary Mathematics Institute & Department of Applied Mathematics and Mathematical Analysis, Universidad Complutense de Madrid, Madrid, Espaa. E-mail: ivorra@ucm.es; angel@mat.ucm.es*

Abstract

Support Vector Machines (SVMs) are powerful tools in machine learning, widely used for classification and regression tasks. Over time, various extensions, such as Probability Estimation SVM (PSVM) and Conditional Probability SVM (CPSVM), have been proposed to enhance SVM performance across different conditions and datasets. This article offers a comprehensive review and analysis of SVMs, with a particular focus on the PSVM and CPSVM models. We delve into the intricacies of these models, addressing computational nuances and presenting corrected formulations where necessary. Our empirical evaluation, conducted on diverse benchmark datasets, implements both linear and nonlinear versions of these SVMs. Performance is benchmarked using the Balanced Accuracy metric. The results highlight the comparative strengths of these models in handling varied datasets and their potential advantages over traditional SVM formulations. To rigorously assess and compare the performance of these SVM variants, we employ statistical tests, including the Friedman test and post hoc analysis.

Keywords: Support Vector Machines, Probability Support Vector Machine, Conditional Probability Support Vector Machine

1. Introduction

Support Vector Machines (**SVM**) have established themselves as powerful tools in the realm of machine learning, offering versatility and efficiency in solving complex classification and regression tasks [7]. However, traditional **SVMs** only provide classification outputs and do not directly estimate class probabilities, which can be of significant scientific interest as they indicate the strength or confidence of the classification outcome. To obtain probability estimates, additional techniques such as Platt scaling [10, 13], Isotonic Regression [19], or monotonic approximation [16] are needed. These techniques enhance the basic **SVM** framework by enabling it to produce probabilistic outputs, thereby making the models more versatile and applicable to a wider range of real-world problems, such as utility models [12] or credit card fraud detection [14].

Over time, various extensions and modifications have been proposed to enhance the performance of **SVMs** under different conditions and datasets. Among these, the **Probability Estimation SVM** (**PSVM**) [17] and **Conditional Probability SVM** (**CPSVM**) [15] models have garnered attention for their ability to handle complex data distributions and improve classification accuracy, particularly in terms of probabilistic output and conditional probability estimations. The main idea of these approaches is to incorporate the probability estimation into the **SVM** optimization model. Additionally, in [15],

*Corresponding author.

the authors estimate the conditional probability function by presenting a new twin SVM model in the context of binary and multiclass classification problems.

In this article, we provide a comprehensive review and analysis of **SVMs**, focusing particularly on the **PSVM** and **CPSVM** models. We delve into the intricacies of these models, addressing computational nuances and presenting corrected formulations where necessary. Our analysis includes a thorough examination of the **Karush-Kuhn-Tucker (KKT)** conditions, shedding light on the theoretical foundations that underpin these **SVM** variants.

To empirically evaluate their effectiveness, we conduct extensive experiments on a diverse set of benchmark datasets sourced from the UCI [2] and KEEL [1] repositories. We implement both linear and nonlinear (specifically Gaussian kernel) versions of **SVMs**, including **PSVM** and **CPSVM**, and benchmark their performance using the **Balanced Accuracy (BAC)** metric. The results from our experiments highlight the comparative strengths of these models in handling varied dataset characteristics and underline their potential advantages over traditional **SVM** formulations.

Furthermore, we apply statistical tests such as the Friedman test and post hoc analysis to rigorously assess and compare the performance of these **SVM** variants across multiple datasets. This statistical validation not only corroborates our experimental results but also provides insights into the relative strengths and weaknesses of each model.

The paper is organized as follows. In Section 2, we present a review of the **SVM**, **PSVM**, and **CPSVM** models. Section 3 is dedicated to analyzing the **PSVM** and **CPSVM** models. In Section 4, we compare the performance of these models on benchmark datasets. Finally, in Section 5, we provide the main conclusions and suggest directions for future research.

2. Prior Works

Let us consider a set of m samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, where $\mathbf{x}_i \in \mathbb{R}^n$ denotes the feature vector representing the i-th sample, and $y_i \in \{-1, +1\}$ its respective class label, for $i = 1, \ldots, m$.

The **classical Support Vector Machine (SVM)** builds a hyperplane of the form $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b = 0$ by solving the following **Quadratic Programming (QP)** problem:

$$\begin{aligned} &\min_{\mathbf{w},b,\boldsymbol{\xi}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \xi_i \\ &\text{s.t.} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, m, \end{aligned} \tag{1}$$

where $C > 0$ is a penalty parameter. In some applications, it is very useful to know the posterior class probability $P(y = 1 \mid \mathbf{x})$ of a new sample $\mathbf{x} \in \mathbb{R}^n$, since it allows us to make decisions, for example, in utility models [12].

To calculate this probability, standard implementations of SVM commonly employ Platt's model-driven approximation [10, 13] since SVMs provide binary classification outputs and do not directly estimate class probabilities. The idea of this approach is to approximate the posterior probability by a sigmoid function, that is,

$$P(y = 1 \mid \mathbf{x}) \approx \frac{1}{1 + \exp(Af(\mathbf{x}) + B)}, \tag{2}$$

where A and B are parameters, which are determined by solving a regularized maximum likelihood problem (see [10, 13] for more details).

Recently, to find the conditional probability function $P(y = 1 \mid \mathbf{x})$ directly, Vapnik and Izmailov [17] consider $P(y = 1 \mid \mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ from a statistical point of view, and compute $(\mathbf{w}, b)$ by solving the

following QP problem, called **Probability estimation SVM (PSVM)**:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \ & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{\varepsilon}\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \ & y_i(\mathbf{w}^\top \mathbf{x}_i + b - 0.5) \geq 0.5\varepsilon - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, m, \\ & 0 \leq \mathbf{w}^\top \mathbf{x}_i + b \leq 1, \quad i = 1, \ldots, m, \end{aligned} \tag{3}$$

where C is a positive parameter and $\varepsilon > 0$ is a small value.

By introducing the respective Lagrangian multipliers, it follows that the dual formulation of problem (3) is given by:

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}} \ & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_i y_i + \beta_i - \gamma_i)(\alpha_j y_j + \beta_j - \gamma_j)\mathbf{x}_i^\top \mathbf{x}_j + \sum_{i=1}^{m}(\gamma_i - 0.5\alpha_i(y_i + \varepsilon)) \\ \text{s.t.} \ & \sum_{i=1}^{m}(\alpha_i y_i + \beta_i - \gamma_i) = 0, \\ & 0 \leq \alpha_i \leq \frac{C}{\varepsilon}, \quad \beta_i, \gamma_i \geq 0, \quad i = 1, \ldots, m, \end{aligned} \tag{4}$$

and the primal solution $\mathbf{w}$ (in terms of the dual variables) can be computed by:

$$\mathbf{w} = \sum_{i=1}^{m}(\alpha_i y_i + \beta_i - \gamma_i)\mathbf{x}_i. \tag{5}$$

Then, a new sample $\mathbf{x} \in \mathbb{R}^n$ is assigned to class +1 or -1 according to the following decision function:

$$f_1(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b - 0.5) = \text{sign}\left(\sum_{i=1}^{m}(\alpha_i y_i + \beta_i - \gamma_i)\mathbf{x}_i^\top \mathbf{x} + b - 0.5\right). \tag{6}$$

Motivated by this novel approach and with the aim of maximizing the conditional probability of the positive class while simultaneously minimizing the conditional probability of the negative class, Shao et al. [15, eq. (37)] propose a new model called **Conditional Probability SVM (CPSVM)**, which is given by:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \ & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C_1}{\varepsilon}\sum_{i=1}^{m}\xi_i - C_2\sum_{i=1}^{m} y_i(\mathbf{w}^\top \mathbf{x}_i + b) \\ \text{s.t.} \ & y_i(\mathbf{w}^\top \mathbf{x}_i + b - 0.5\varepsilon) \geq -\xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, m, \\ & 0 \leq \mathbf{w}^\top \mathbf{x}_i + b \leq 1, \quad i = 1, \ldots, m, \end{aligned} \tag{7}$$

where $C_1, C_2 > 0$, and $\varepsilon > 0$ is a small value. In this case, a new sample $\mathbf{x} \in \mathbb{R}^n$ is assigned to class +1 or -1 according to the following decision function:

$$f_2(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b - 0.5\varepsilon). \tag{8}$$

In Proposition 3.2 below, we present the correct dual formulation of Problem (7). Indeed, the expression given in [15, eq. (38)] contains calculation errors and corresponds to the dual of a different primal formulation, which we will introduce in the next section.

3. Analysis of PSVM and CPSVM Models

In this section, we present results derived from the **Karush-Kuhn-Tucker (KKT)** conditions of Problems (3) and (7), following the derivation presented in [3] (see Propositions 3.1 and 3.3 below). Additionally, we present the correct dual formulation of Problem (7) and propose another model related to the **CPSVM** approach.

Proposition 3.1. *Suppose that $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ is a nonzero solution of Problem (4), with $\varepsilon \in (0, 1]$. Then:*

(a) If there exists a component of $\boldsymbol{\alpha}$ such that $\alpha_i \in (0, \frac{C}{\varepsilon})$, the variable b can be computed by

$$b = 0.5\varepsilon y_i + 0.5 - \mathbf{w}^\top \mathbf{x}_i. \tag{9}$$

Moreover, for such i, one has $\xi_i = 0$ and when $\varepsilon \in (0, 1)$, we have $\beta_i = \gamma_i = 0$.

(b) If there exists a component of $\boldsymbol{\beta}$ such that $\beta_i > 0$, then

$$b = -\mathbf{w}^\top \mathbf{x}_i \quad \textit{and} \quad \gamma_i = 0. \tag{10}$$

Moreover, if $y_i = -1$, one has $\xi_i = 0$.

(c) If there exists a component of $\boldsymbol{\gamma}$ such that $\gamma_i > 0$, then

$$b = 1 - \mathbf{w}^\top \mathbf{x}_i \quad \textit{and} \quad \beta_i = 0. \tag{11}$$

Moreover, if $y_i = 1$, one has $\xi_i = 0$.

The proof of Proposition 3.1 can be found in Appendix A.

Remark 1. *We note that in the case of $\alpha_i \in (0, \frac{C}{\varepsilon})$ and $\varepsilon = 1$, we also have $\beta_i = 0$ and $\gamma_i = 0$ for $y_i = 1$ and $y_i = -1$, respectively.*

Remark 2. *Similar to **SVM**, in practice, it is more robust to average over all support vectors (with $\xi_i = 0$) and calculate the bias b as*

$$b = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \left(0.5\varepsilon y_i + 0.5 - \mathbf{w}^\top \mathbf{x}_i\right), \tag{12}$$

where $\mathcal{S}$ is defined by $\mathcal{S} = \{i \in \{1, \ldots, m\} : \alpha_i \in (0, \frac{C}{\varepsilon})\}$ and $|\mathcal{S}|$ denotes the cardinality of the set $\mathcal{S}$.

We now present the correct dual formulation of Problem (7), whose derivation is presented in Appendix B.

Proposition 3.2. *The dual formulation of Problem (7) is given by*

$$\begin{aligned} \min_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}} \; & \sum_{i=1}^{m} \sum_{j=1}^{m} (C_2 y_i + \alpha_i y_i + \beta_i - \gamma_i)(C_2 y_j + \alpha_j y_j + \beta_j - \gamma_j) \mathbf{x}_i^\top \mathbf{x}_j + \sum_{i=1}^{m} (\gamma_i - 0.5\alpha_i y_i \varepsilon) \\ s.t. \; & \sum_{i=1}^{m} (C_2 y_i + \alpha_i y_i + \beta_i - \gamma_i) = 0, \\ & 0 \le \alpha_i \le \frac{C_1}{\varepsilon}, \; \beta_i, \gamma_i \ge 0, \quad i = 1, \ldots, m. \end{aligned} \tag{13}$$

Considering this dual formulation, we have the following properties related to Problems (7) and (13).

Proposition 3.3. *Suppose that $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ is a nonzero solution of Problem (13), with $\varepsilon \in (0, 1]$. Then:*

(a) The solution $\mathbf{w}$ *related to Problem* (7) *is given by*

$$\mathbf{w} = \sum_{i=1}^{m} (C_2 y_i + \alpha_i y_i + \beta_i - \gamma_i)\mathbf{x}_i. \tag{14}$$

(b) If there exists a component of $\boldsymbol{\alpha}$ *such that* $\alpha_i \in (0, \frac{C_1}{\varepsilon})$*, the variable* b *can be computed by*

$$b = 0.5\varepsilon - \mathbf{w}^\top \mathbf{x}_i. \tag{15}$$

Moreover, for such i*, one has* $\xi_i = \beta_i = \gamma_i = 0$*.*

(c) If there exists a component of $\boldsymbol{\beta}$ *such that* $\beta_i > 0$*, then*

$$b = -\mathbf{w}^\top \mathbf{x}_i \quad \textit{and} \quad \gamma_i = 0.$$

Moreover, if $y_i = -1$*, one has* $\xi_i = 0$*.*

(d) If there exists a component of $\boldsymbol{\gamma}$ *such that* $\gamma_i > 0$*, then*

$$b = 1 - \mathbf{w}^\top \mathbf{x}_i \quad \textit{and} \quad \beta_i = 0.$$

Moreover, if $y_i = 1$*, one has* $\xi_i = 0$*.*

Its proof can be found in Appendix C.

Remark 3. *Similar to Remark 2, the bias* b *can be computed as*

$$b = \frac{1}{|\mathcal{S}_1|} \sum_{i \in \mathcal{S}_1} \left(0.5\varepsilon - \mathbf{w}^\top \mathbf{x}_i\right), \tag{16}$$

with $\mathcal{S}_1 = \{i \in \{1, \ldots, m\} : \alpha_i \in (0, \frac{C_1}{\varepsilon})\}$.

Next, we propose an alternative model corresponding to the primal formulation of the dual formulation provided in [15, eq. (38)]. It's worth noting that this formulation does not correspond to (7) as stated in that reference. Here, we propose

$$\begin{aligned} \min_{\mathbf{w}, b, \boldsymbol{\xi}} \ & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C_1}{\varepsilon}\sum_{i=1}^{m} \xi_i - C_2 \sum_{i=1}^{m} y_i(\mathbf{w}^\top \mathbf{x}_i + b) \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b - 0.5) \geq 0.5\varepsilon - \xi_i, \quad \xi_i \geq 0, \ i = 1, \ldots, m, \\ & 0 \leq \mathbf{w}^\top \mathbf{x}_i + b \leq 1, \quad i = 1, \ldots, m. \end{aligned} \tag{17}$$

This formulation differs from model (7) in the first linear constraints and in the decision function (see eq. (19) below). However, the following result shows that both formulations are equivalent under certain conditions of the parameter ε.

Proposition 3.4. *Formulations* (7) *and* (17) *are equivalent when* $\varepsilon \geq 0.5$*.*

Proof. We note that the first linear constraints of (7) can be written as

$$\begin{aligned} y_i(\mathbf{w}^\top \mathbf{x}_i + b - \frac{\varepsilon}{2}) \geq -\xi_i &\Leftrightarrow y_i(\mathbf{w}^\top \mathbf{x}_i + b - 0.5) \geq -0.5y_i(1-\varepsilon) - \xi_i \\ &\Leftrightarrow y_i(\mathbf{w}^\top \mathbf{x}_i + b - 0.5) \geq 0.5\varepsilon - \left(0.5y_i(1-\varepsilon) + 0.5\varepsilon + \xi_i\right). \end{aligned}$$

Let us define

$$\zeta_i = 0.5(y_i(1-\varepsilon) + \varepsilon) + \xi_i.$$

If $y_i = 1$, we have $\zeta_i = 0.5 + \xi_i \geq 0$, and if $y_i = -1$, we have $\zeta_i = 0.5(2\varepsilon - 1) + \xi_i \geq 0$, for $\varepsilon \geq 0.5$. Thus, we have the following linear constraints: $y_i(\mathbf{w}^\top \mathbf{x}_i + b - 0.5) \geq 0.5\varepsilon - \zeta_i$, with $\zeta_i \geq 0$ for $i = 1, \ldots, m$ and $\varepsilon \geq 0.5$. Moreover, given that y_i and ε are parameters, the second term of the objective function of Problem (7) becomes $\frac{C_1}{\varepsilon}\sum_{i=1}^{m} \zeta_i$. With this, the equivalence between both problems is established. □

By introducing the respective Lagrangian multipliers and proceeding in a similar way to the proof of Proposition 3.2, it follows that the dual formulation of Problem (17) is given by

$$\begin{aligned}\min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}} \; & \sum_{i=1}^{m}\sum_{j=1}^{m}(C_2 y_i + \alpha_i y_i + \beta_i - \gamma_i)(C_2 y_j + \alpha_j y_j + \beta_j - \gamma_j)\mathbf{x}_i^\top \mathbf{x}_j \\ & + \sum_{i=1}^{m}(\gamma_i - 0.5\alpha_i(y_i + \varepsilon)) \\ \text{s.t.} \; & \sum_{i=1}^{m}(C_2 y_i + \alpha_i y_i + \beta_i - \gamma_i) = 0, \\ & 0 \le \alpha_i \le \frac{C_1}{\varepsilon},\ \beta_i, \gamma_i \ge 0, \quad i = 1, \ldots, m,\end{aligned} \tag{18}$$

and the primal solution $\mathbf{w}$ can be computed by equation (14). Then, a new sample $\mathbf{x} \in \mathbb{R}^n$ is assigned to class +1 or -1 according to the following decision function:

$$f_3(\mathbf{x}) = \operatorname{sign}(\mathbf{w}^\top \mathbf{x} + b - 0.5) = \operatorname{sign}\left(\sum_{i=1}^{m}(C_2 y_i + \alpha_i y_i + \beta_i - \gamma_i)\mathbf{x}_i^\top \mathbf{x} + b - 0.5\right). \tag{19}$$

Similar to the results presented previously, we present some properties related to Problems (17) and (18). The proof follows in a similar way to that of Proposition 3.1.

Proposition 3.5. *Suppose that $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ is a nonzero solution of Problem (18), with $\varepsilon \in (0, 1]$. Then:*

(a) If there exists a component of $\boldsymbol{\alpha}$ such that $\alpha_i \in (0, \frac{C_1}{\varepsilon})$, the variable b can be computed by

$$b = 0.5(1 + \varepsilon y_i) - \mathbf{w}^\top \mathbf{x}_i. \tag{20}$$

Moreover, for such i, one has $\xi_i = 0$ and when $\varepsilon \in (0, 1)$, we have $\beta_i = \gamma_i = 0$.

(b) If there exists a component of $\boldsymbol{\beta}$ such that $\beta_i > 0$, then

$$b = -\mathbf{w}^\top \mathbf{x}_i \quad \textit{and} \quad \gamma_i = 0. \tag{21}$$

Moreover, if $y_i = -1$, one has $\xi_i = 0$.

(c) If there exists a component of $\boldsymbol{\gamma}$ such that $\gamma_i > 0$, then

$$b = 1 - \mathbf{w}^\top \mathbf{x}_i \quad \textit{and} \quad \beta_i = 0. \tag{22}$$

Moreover, if $y_i = 1$, one has $\xi_i = 0$.

Remark 4. *The above formulations can be directly extended to the nonlinear kernel context. The idea is to replace the inner products $\mathbf{x}_i^\top \mathbf{x}_j$ with $K(\mathbf{x}_i, \mathbf{x}_j)$, where $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a kernel function that satisfies the Mercer condition [4]. Some examples are the Gaussian and polynomial kernels, which are defined by $K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/2\sigma^2)$, with $\sigma > 0$, and $K(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x}^\top \mathbf{y})^d$, with $d \in \mathbb{Z}_{++}$, respectively. The conditional probability and decision functions (for instance, in the **PSVM** model) would then take the form:*

$$P(y = 1 \mid \mathbf{x}) = \sum_{i=1}^{m}(\alpha_i y_i + \beta_i - \gamma_i)K(\mathbf{x}, \mathbf{x}_i) + b$$

and

$$f_1(\mathbf{x}) = \operatorname{sign}\left(\sum_{i=1}^{m}(\alpha_i y_i + \beta_i - \gamma_i)K(\mathbf{x}, \mathbf{x}_i) + b - 0.5\right).$$

To illustrate the shape of the classifiers generated by each approach, Figures 1 and 2 present a 2D dataset along with the respective optimal classifiers obtained from the four described formulations, each employing linear and nonlinear kernel versions, respectively. These classifiers were constructed using the best parameters for each approach (see Section 4.1 for details about the techniques used).

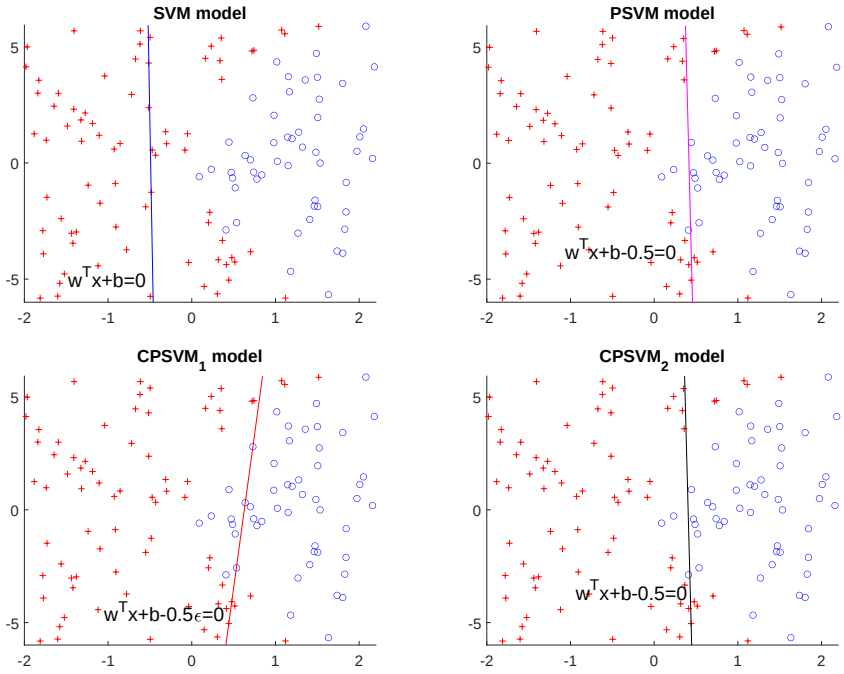


Figure 1: Optimal hyperplane for **SVM**, **PSVM**, and **CPSVM** (cf. eqs. (7) and (17)) approaches (with linear kernel).

4. Numerical Experiments

In this section, we conducted experiments on 17 benchmark datasets from the UCI [2] and KEEL [1] data repositories to assess the effectiveness of the **SVM**, **PSVM**, **CPSVM**$_1$ (cf. eq. (7)), and **CPSVM**$_2$ (cf. eq. (17)) approaches. First, a description of the benchmark datasets is provided in Section 4.1, including the experimental setting. Then, the main results are presented and discussed in Sections 4.2 and 4.3.

4.1. Benchmark Datasets and Experimental Setting

In Table 1, we present relevant information about the seventeen datasets used in this experimental section, which were scaled to the range $[-1, 1]$. This information includes the dataset name, number of elements $N_e \in \mathbb{N}$, number of features $N_f \in \mathbb{N}$, and the shorthand notation used for reference.

Table 1: Characteristics of Benchmark Datasets

Datasets (shorthand notation)	N_e	N_f	Datasets	N_e	N_f
Sonar (ds1)	208	60	M-class Solar (ds10)	1389	10
Heart (ds2)	270	13	Yeast3 (ds11)	1484	8
BUPA Liver (ds3)	345	6	Yeast4 (ds12)	1484	8
Ionosphere (ds4)	351	34	Titanic (ds13)	2201	3
Wisconsin Breast Cancer (ds5)	569	30	Segment0 (ds14)	2308	19
Australian Credit (ds6)	690	14	Image (ds15)	2310	19
Pima Indians Diabetes (ds7)	768	8	Waveform (ds16)	5000	21
German Credit (ds8)	1000	24	Phoneme (ds17)	5404	5
Splice (ds9)	1000	24			

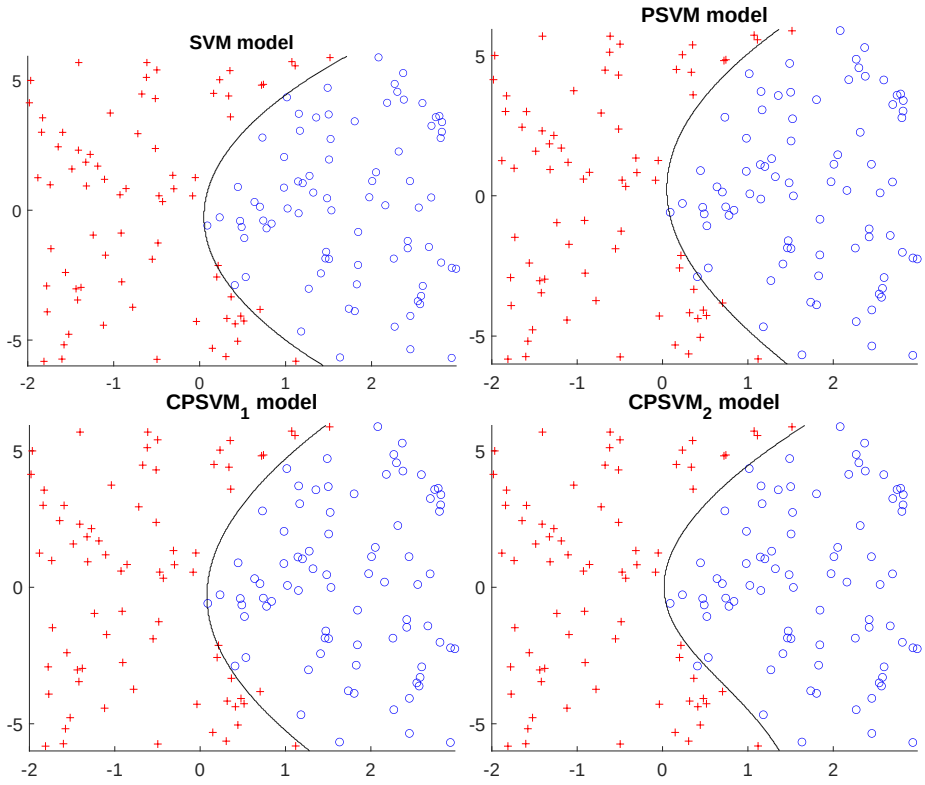


Figure 2: Optimal hyperplane for **SVM**, **PSVM**, and **CPSVM** (cf. eqs. (7) and (17)) approaches (with RBF kernel).

To analyze the performance of the **SVM**, **PSVM**, **CPSVM**$_1$, and **CPSVM**$_2$ approaches, we implemented both their linear and nonlinear kernel versions. In the latter case, we used the Gaussian kernel.

The experimental setting is as follows: the hyperparameters for each approach were tuned using ten-fold cross-validation, where the entire dataset was divided into ten subsets. The **Balance accuracy (BAC)** was used as the performance metric. We explored the following sets of hyperparameters: $C, C_1, C_2, \sigma \in \{2^{-7}, 2^{-6}, \ldots, 2^{6}, 2^{7}\}$ and $\varepsilon \in \{2^{-7}, 2^{-6}, \ldots, 2^{-1}, 2^{0}\}$. For the **CPSVM**$_1$ and **CPSVM**$_2$ (nonlinear kernel versions) approaches, we imposed the relation $C_1 = C_2$ to reduce the number of combinations in the grid search.

All experiments were implemented and run in Matlab R2022a on a MacBook Pro with a 2GHz Intel Core i5 processor (4 cores) and 16GB RAM. We used QUADSOLVE solver (available in Matlabs Spider toolbox [18]) for **SVM** and the Matlab's `quadprog` function for the **PSVM**, **CPSVM**$_1$, and **CPSVM**$_2$ approaches.

4.2. Results on the Benchmark Datasets

Tables 2 and 3 summarize the results of each classification model applied to all datasets concerning the evaluation criterion BAC. The highest BAC value for each model is presented, and the best model is highlighted in bold for each dataset. The optimal hyperparameters and CPU time for all datasets and models are reported in Tables D.1 and D.2 in Appendix D.

From Table 2, we observe that the best predictive results are achieved by **CPSVM**$_2$, which has the highest BAC in eight out of seventeen datasets. The second-best results are achieved with the **SVM** and

Table 2: **Linear Kernel**. Predictive performance summary in terms of BAC×100. We report the best configuration for each classification method.

Datasets	**SVM**	**PSVM**	**CPSVM$_1$**	**CPSVM$_2$**
ds1	76.61	77.14	77.14	**78.69**
ds2	84.08	84.08	**85.08**	84.33
ds3	68.01	68.01	68.17	**69.33**
ds4	85.23	85.67	**88.35**	85.39
ds5	97.34	**97.58**	96.28	96.88
ds6	86.22	86.19	86.78	**87.21**
ds7	72.14	**73.25**	73.12	**73.25**
ds8	69.43	69.81	68.95	**70.07**
ds9	**81.60**	80.41	80.82	80.56
ds10	50.00	50.00	**55.78**	50.53
ds11	84.22	85.10	83.18	**85.14**
ds12	50.00	50.00	63.98	**64.01**
ds13	**71.09**	69.83	70.46	69.96
ds14	99.37	99.57	99.36	**99.60**
ds15	54.91	50.72	**57.41**	53.82
ds16	**87.46**	85.79	83.45	85.29
ds17	**75.17**	71.71	65.62	71.83

Table 3: **Gaussian Kernel**. Predictive performance summary in terms of BAC×100. We report the best configuration for each classification method.

Datasets	**SVM**	**PSVM**	**CPSVM$_1$**	**CPSVM$_2$**
ds1	87.59	87.94	88.85	**89.15**
ds2	84.42	84.42	83.67	**84.50**
ds3	71.95	71.98	72.31	**72.50**
ds4	95.19	**95.40**	95.19	95.13
ds5	97.11	**98.16**	97.42	97.58
ds6	86.64	86.64	**87.34**	86.92
ds7	72.23	72.59	73.23	**74.53**
ds8	68.48	68.79	**70.83**	**70.83**
ds9	88.57	88.57	88.44	**88.59**
ds10	53.79	54.21	**64.04**	56.85
ds11	86.96	87.44	**88.07**	86.92
ds12	64.38	67.00	**74.46**	68.67
ds13	**71.09**	**71.09**	**71.09**	**71.09**
ds14	99.67	99.79	99.69	**99.84**
ds15	67.94	68.48	**69.58**	68.76
ds16	88.25	88.08	**88.71**	88.05
ds17	88.01	88.23	**89.27**	88.42

CPSVM$_1$ models, each performing best in four datasets.

Additionally, Table 3 shows that the best predictive results are achieved by **CPSVM$_1$** and **CPSVM$_2$**, with the highest BAC in nine and eight out of seventeen datasets, respectively. This indicates that when the Gaussian kernel is used, the **CPSVM$_1$** and **CPSVM$_2$** methods perform better than **SVM** and **PSVM** ones.

4.3. Statistical Analysis

To statistically assess the performance of the studied models across multiple datasets, we employ the Friedman test along with the corresponding post hoc test [5]. This combination is a simple, non-parametric, and robust test.

First, we compute the average rank of the **SVM**, **PSVM**, **CPSVM**$_1$, and **CPSVM**$_2$ approaches across all datasets. A rank of 1 corresponds to the model with the highest BAC for a given dataset. The average ranks of the four models, derived from the results presented in Tables 2 and 3, are reported in Table 4.

Second, we calculate the Friedman test with Iman-Davenport correction (F_F) to verify whether all models are statistically similar (null hypothesis). Let $\mathcal{N}_c$ denote the number of models, $\mathcal{N}_d$ the number of datasets, r_i^j the rank of the j-th model on the i-th dataset, and $R_j = \frac{1}{\mathcal{N}_d}\sum_{i=1}^{\mathcal{N}_d} r_i^j$ the average rank of the j-th model (reported in Table 4). The F_F statistic (distributed according to the F-distribution with $\mathcal{N}_c - 1$ and $(\mathcal{N}_c - 1)(\mathcal{N}_d - 1)$ degrees of freedom) is given by:

$$F_F = \frac{(\mathcal{N}_d - 1)\chi_F^2}{\mathcal{N}_d(\mathcal{N}_c - 1) - \chi_F^2}, \tag{23}$$

where

$$\chi_F^2 = \frac{12\mathcal{N}_d}{\mathcal{N}_c(\mathcal{N}_c + 1)}\left(\sum_{j=1}^{\mathcal{N}_c} R_j^2 - \frac{\mathcal{N}_c(\mathcal{N}_c + 1)^2}{4}\right), \tag{24}$$

which is distributed according to the chi-square distribution with $\mathcal{N}_c - 1$ degrees of freedom and is conservative [5].

Table 4: Average ranks (R_j) of **SVM**, **PSVM**, **CPSVM**$_1$, and **CPSVM**$_2$.

Type	**SVM**	**PSVM**	**CPSVM**$_1$	**CPSVM**$_2$
linear	2.6765	2.8824	2.5294	1.9118
nonlinear (rbf)	3.3235	2.5882	2.0294	2.0588

aneca

Here, we obtain $\chi_F^2 = 5.347$ and $F_F = 1.87$ for the linear kernel case, and $\chi_F^2 = 11.241$ and $F_F = 4.524$ for the nonlinear kernel case. For a significance level of $\alpha = 0.05$, the critical value is $F_{0.05}(3, 48) = 2.798$.

In the linear kernel case, $F_F < 2.798$; therefore, there is no statistically significant difference among the models in terms of BAC. However, the **CPSVM**$_2$ model has a much better average rank (1.9118) than the others.

In the nonlinear kernel case, $F_F > 2.798$; thus, the null hypothesis is rejected, indicating a significant difference between the four models studied. Consequently, we proceed with a post-hoc test. We employ the Nemenyi test to pairwise analyze the performance differences of all models to identify specific differences. In this test, we use the so-called Critical Difference (CD) defined by:

$$CD = q_\alpha\sqrt{\frac{\mathcal{N}_c(\mathcal{N}_c + 1)}{6\mathcal{N}_d}}, \tag{25}$$

where critical values q_α are based on the Studentized range statistic divided by $\sqrt{2}$. At the significance level of $\alpha = 0.05$, the value of q_α is 2.569, and thus $CD = 1.138$. If the absolute value of the difference between the average ranks of the two models is greater than CD, there is a statistically significant difference between their performance. Figure 3 illustrates the comparison of **SVM**, **PSVM**, **CPSVM**$_1$, and **CPSVM**$_2$ methods based on their average ranks. Groups of methods that do not have a significant difference are connected by a solid red line.

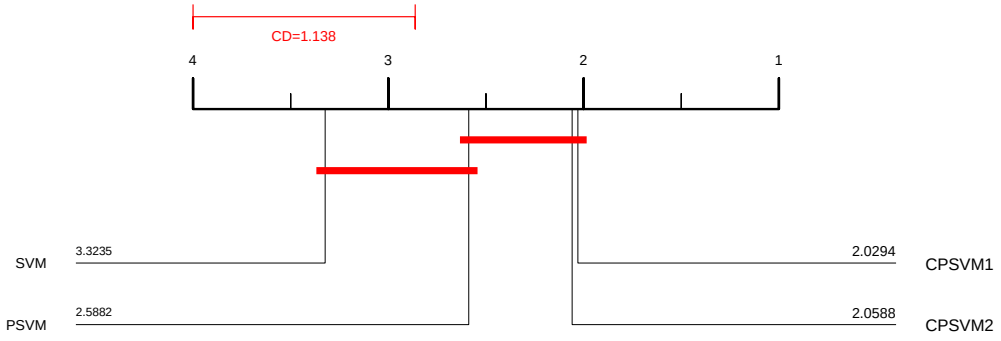


Figure 3: **Nemenyi test**. Comparison of four models based on their average ranks. Groups of methods without significant differences are connected by a red line.

According to Figure 3, it is observed that the **CPSVM**$_1$ and **CPSVM**$_2$ models are statistically significantly better than the **SVM** model. However, these models are not statistically significantly better than the **PSVM** approach, and the **PSVM** model is also not statistically significantly better than the **SVM** model. Additionally, **CPSVM**$_1$ and **CPSVM**$_2$ models are relatively close together, with average ranks of 2.0294 and 2.0588, respectively.

5. Conclusions and Future Research

In this article, we presented a comprehensive review and analysis of the **SVM**, **PSVM**, and **CPSVM** models previously proposed in the literature. Our focus was specifically on the **PSVM** and **CPSVM** models, addressing calculation errors found in previous studies. We provided the correct dual formulation of the **CPSVM** model (denoted as **CPSVM**$_1$), correcting the discrepancies noted in [15, eq. (38)]. Our analysis included results derived from the Karush-Kuhn-Tucker (KKT) conditions. Additionally, we proposed a new model related to the **CPSVM** approach (denoted as **CPSVM**$_2$).

To evaluate the performance of the **SVM**, **PSVM**, **CPSVM**$_1$, and **CPSVM**$_2$ approaches, we performed numerical experiments on seventeen benchmark datasets. Both linear and nonlinear (Gaussian kernel) versions of these models were implemented. The results highlight the performance of each model with respect to the BAC evaluation criterion.

In the linear case, we observed that **CPSVM**$_2$ achieved the highest BAC in eight out of seventeen datasets, while **SVM** and **CPSVM**$_1$ each performed best in four datasets. In the nonlinear case, **CPSVM**$_1$ and **CPSVM**$_2$ achieved the highest BAC in nine and eight datasets, respectively. This indicates that **CPSVM**$_1$ and **CPSVM**$_2$ outperform **SVM** and **PSVM** in the nonlinear setting.

To statistically assess the performance of the models across multiple datasets, we employed the Friedman test along with the corresponding post hoc test. The results indicated that **CPSVM**$_1$ and **CPSVM**$_2$ models are statistically significantly better than the **SVM** model. However, there is no statistically significant difference between the **CPSVM**$_1$, **CPSVM**$_2$, and **PSVM** models. Additionally, the **CPSVM**$_1$ and **CPSVM**$_2$ models are relatively close in performance.

This study demonstrates that the **CPSVM** models, namely **CPSVM**$_1$ and **CPSVM**$_2$, show superior performance over traditional **SVM** and **PSVM** models, especially when using a Gaussian kernel. These findings underscore the promising potential of **CPSVM** approaches for classification tasks across various datasets, suggesting the need for further in-depth studies.

Regarding future work, it would be interesting to increase the robustness of **PSVM** and **CPSVM** models to handle uncertain data, noisy data, and outliers. To deal with uncertain data, **Second-Order Cone Programming (SOCP)** appears to be a useful tool for formulating robust models [11]. **SOCP** problems allow us to tackle a wide variety of real-world applications, particularly classification problems. On the other hand, to increase the robustness of these models in dealing with noisy data and outliers, the convex pinball [9] and non-convex ramp loss [8] functions appear to be useful and appropriate tools.

Acknowledgements

This work was carried out thanks to financial support from ANID Chile: FONDECYT through grants 1201403 and 1230694, UMMISCO UGB, and the Spanish "Ministry of Science and Innovation" under Project PID2019-106337GB-I00.

Appendix A. Proof of Proposition 3.1

Proof. (a) The Lagrangian function associated with problem (3) is given by:

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\gamma}) &= \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{\varepsilon}\sum_{i=1}^{m}\xi_i \\ &\quad - \sum_{i=1}^{m}\alpha_i\left(y_i(\mathbf{w}^\top\mathbf{x}_i + b - 0.5) - 0.5\varepsilon + \xi_i\right) \\ &\quad - \sum_{i=1}^{m}\theta_i\xi_i - \sum_{i=1}^{m}\beta_i(\mathbf{w}^\top\mathbf{x}_i + b) + \sum_{i=1}^{m}\gamma_i(\mathbf{w}^\top\mathbf{x}_i + b - 1), \end{aligned}$$

where $\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{R}^m_+$ are the Lagrangian multipliers.

Since Problem (3) is convex and satisfies the Slater condition, the **Karush-Kuhn-Tucker (KKT)** conditions are necessary and sufficient for optimality [6]. These conditions, particularly the stationarity and complementary slackness conditions, are formulated as follows:

$$\nabla_{\mathbf{w}} L = \mathbf{w} - \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i - \sum_{i=1}^{m}\beta_i \mathbf{x}_i + \sum_{i=1}^{m}\gamma_i \mathbf{x}_i = \mathbf{0}, \tag{A.1}$$

$$\nabla_b L = -\sum_{i=1}^{m}\alpha_i y_i - \sum_{i=1}^{m}\beta_i + \sum_{i=1}^{m}\gamma_i = 0, \tag{A.2}$$

$$\nabla_{\xi_i} L = \frac{C}{\varepsilon} - \alpha_i - \theta_i = 0, \quad i = 1, \ldots, m, \tag{A.3}$$

$$\alpha_i\left(y_i(\mathbf{w}^\top\mathbf{x}_i + b - 0.5) - 0.5\varepsilon + \xi_i\right) = 0, \quad i = 1, \ldots, m, \tag{A.4}$$

$$\theta_i\xi_i = 0, \quad i = 1, \ldots, m, \tag{A.5}$$

$$\beta_i(\mathbf{w}^\top\mathbf{x}_i + b) = 0, \quad i = 1, \ldots, m, \tag{A.6}$$

$$\gamma_i(\mathbf{w}^\top\mathbf{x}_i + b - 1) = 0, \quad i = 1, \ldots, m. \tag{A.7}$$

We assume that there exists $i \in \{1, \ldots, m\}$ such that $\alpha_i \in (0, \frac{C}{\varepsilon})$. Then, from equations (A.3) and (A.4), we have that $\theta_i > 0$ and:

$$y_i(\mathbf{w}^\top\mathbf{x}_i + b - 0.5) - 0.5\varepsilon + \xi_i = 0. \tag{A.8}$$

Moreover, from equation (A.5), it follows that $\xi_i = 0$. Therefore, from equality (A.8), we deduce that:

$$b = 0.5\varepsilon y_i + 0.5 - \mathbf{w}^\top \mathbf{x}_i. \tag{A.9}$$

Substituting (A.9) into equation (A.6), we have:

$$\beta_i(0.5\varepsilon y_i + 0.5) = 0.$$

Since $\varepsilon \in (0, 1)$, the above equality implies that $\beta_i = 0$.

Similarly, substituting (A.9) into (A.7) gives us:

$$\gamma_i(\varepsilon y_i - 1) = 0.$$

This implies that $\gamma_i = 0$.

(b) We assume that there exists $i \in \{1, \ldots, m\}$ such that $\beta_i > 0$. Then, from equation (A.6), it follows that $\mathbf{w}^\top \mathbf{x}_i + b = 0$. This equality and equation (A.7) imply that $\gamma_i = 0$. Substituting the above equality into equation (A.4), we have:

$$\alpha_i \left(-0.5(y_i + \varepsilon) + \xi_i\right) = 0.$$

If $y_i = -1$ and $\varepsilon \in (0, 1)$, then $\alpha_i = 0$ and hence, $\theta_i = \frac{C}{\varepsilon}$ (cf. eq. (A.3)). This implies that $\xi_i = 0$ (cf. eq. (A.5)). Now, assuming $\varepsilon = 1$, we have $\alpha_i \xi_i = 0$, and hence $(C - \theta_i)\xi_i = 0$ (cf. eq. (A.3)). By using (A.5), we deduce that $\xi_i = 0$.

(c) Finally, we suppose that there exists $i \in \{1, \ldots, m\}$ such that $\gamma_i > 0$. Then, from equation (A.7), it follows that $\mathbf{w}^\top \mathbf{x}_i + b - 1 = 0$. This equality and equation (A.6) imply that $\beta_i = 0$. Substituting the above equality into equation (A.4), we have:

$$\alpha_i \left(0.5(y_i - \varepsilon) + \xi_i\right) = 0.$$

If $y_i = 1$ and $\varepsilon \in (0, 1)$, then $\alpha_i = 0$ and hence, $\theta_i = \frac{C}{\varepsilon}$ (cf. eq. (A.3)). This implies that $\xi_i = 0$ (cf. eq. (A.5)). Now, assuming $\varepsilon = 1$, by using the same argument as in part (b), we deduce that $\xi_i = 0$.

□

Appendix B. Proof of Proposition 3.2

The Lagrangian function associated with problem (7) is given by:

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \;=\; & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C_1}{\varepsilon}\sum_{i=1}^{m}\xi_i - C_2\sum_{i=1}^{m} y_i(\mathbf{w}^\top \mathbf{x}_i + b) - \sum_{i=1}^{m}\theta_i\xi_i \\ & - \sum_{i=1}^{m}\beta_i(\mathbf{w}^\top \mathbf{x}_i + b) - \sum_{i=1}^{m}\alpha_i\left(y_i(\mathbf{w}^\top \mathbf{x}_i + b - 0.5\varepsilon) + \xi_i\right) \\ & + \sum_{i=1}^{m}\gamma_i(\mathbf{w}^\top \mathbf{x}_i + b - 1), \end{aligned} \tag{B.1}$$

where $\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{R}^m_+$ are the Lagrangian multipliers.

Since Problem (7) is convex with a Slater point, the KKT conditions are necessary and sufficient for deriving its dual formulation. The KKT conditions for Problem (7) are given by:

$$\mathbf{w} - C_2\sum_{i=1}^{m} y_i\mathbf{x}_i - \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i - \sum_{i=1}^{m}\beta_i\mathbf{x}_i + \sum_{i=1}^{m}\gamma_i\mathbf{x}_i = \mathbf{0}, \tag{B.2}$$

$$-C_2\sum_{i=1}^{m} y_i - \sum_{i=1}^{m}\alpha_i y_i - \sum_{i=1}^{m}\beta_i + \sum_{i=1}^{m}\gamma_i = 0, \tag{B.3}$$

$$\frac{C_1}{\varepsilon} - \alpha_i - \theta_i = 0, \quad i = 1, \ldots, m, \tag{B.4}$$

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b - 0.5\varepsilon) + \xi_i \geq 0, \quad i = 1, \ldots, m, \tag{B.5}$$

$$\mathbf{w}^\top \mathbf{x}_i + b \geq 0, \quad i = 1, \ldots, m, \tag{B.6}$$

$$1 - \mathbf{w}^\top \mathbf{x}_i - b \geq 0, \quad i = 1, \ldots, m, \tag{B.7}$$

$$\xi_i,\, \theta_i,\, \alpha_i,\, \beta_i,\, \gamma_i \geq 0, \quad i = 1, \ldots, m, \tag{B.8}$$

$$\alpha_i \left(y_i(\mathbf{w}^\top \mathbf{x}_i + b - 0.5\varepsilon) + \xi_i\right) = 0, \quad i = 1, \ldots, m, \tag{B.9}$$

$$\beta_i(\mathbf{w}^\top \mathbf{x}_i + b) = 0, \quad i = 1, \ldots, m, \tag{B.10}$$

$$\gamma_i(\mathbf{w}^\top \mathbf{x}_i + b - 1) = 0, \quad i = 1, \ldots, m, \tag{B.11}$$

$$\theta_i \xi_i = 0, \quad i = 1, \ldots, m. \tag{B.12}$$

From (B.4) and (B.8), we have $0 \leq \alpha_i \leq \frac{C_1}{\varepsilon}$, for $i = 1, \ldots, m$. Using the expressions (B.2) to (B.4) in the Lagrangian function (B.1), we obtain the dual formulation given in (13).

Appendix C. Proof of Proposition 3.3

Proof. (a) From equation (B.2), we derive the expression (14).

(b) We assume that there exists $i \in \{1, \ldots, m\}$ such that $\alpha_i \in (0, \frac{C_1}{\varepsilon})$. Then, from equations (B.4) and (B.9), we have that $\theta_i > 0$ and:

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b - 0.5\varepsilon) + \xi_i = 0.$$

Moreover, from equation (B.12), it follows that $\xi_i = 0$. Therefore, from the above equality, we deduce that:

$$b = 0.5\varepsilon - \mathbf{w}^\top \mathbf{x}_i. \tag{C.1}$$

Substituting (C.1) into equations (B.10) and (B.11), we have that $\beta_i = \gamma_i = 0$.

(c) We assume that there exists $i \in \{1, \ldots, m\}$ such that $\beta_i > 0$. Then, from equation (B.10), it follows that $\mathbf{w}^\top \mathbf{x}_i + b = 0$. This equality and equation (B.11) imply that $\gamma_i = 0$. Substituting the above equality into equation (B.9), we have:

$$\alpha_i(-0.5\varepsilon y_i + \xi_i) = 0.$$

If $y_i = -1$, then $\alpha_i = 0$ and hence, $\theta_i = \frac{C_1}{\varepsilon}$ (cf. eq. (B.4)). This implies that $\xi_i = 0$ (cf. eq. (B.12)).

(d) Finally, we suppose that there exists $i \in \{1, \ldots, m\}$ such that $\gamma_i > 0$. Then, from equation (B.11), it follows that $\mathbf{w}^\top \mathbf{x}_i + b - 1 = 0$. This equality and equation (B.10) imply that $\beta_i = 0$. Substituting the above equality into equation (B.9), we have:

$$\alpha_i \left((1 - 0.5\varepsilon) y_i + \xi_i\right) = 0.$$

If $y_i = 1$, then $\alpha_i = 0$ and hence, $\theta_i = \frac{C_1}{\varepsilon}$ (cf. eq. (B.4)). Thus, $\xi_i = 0$ (cf. eq. (B.12)). □

Table D.1: **Linear Kernel**. Best combination of hyper-parameters and CPU time for all models.

Datasets	**SVM** C Time (s)	**PSVM** $\{\varepsilon, C\}$ Time (s)	**CPSVM$_1$** $\{\varepsilon, C_1, C_2\}$ Time (s)	**CPSVM$_2$** $\{\varepsilon, C_1, C_2\}$ Time (s)
ds1	2^{0}	$\{2^{-7}, 2^{0}\}$	$\{2^{-4}, 2^{-6}, 2^{-6}\}$	$\{2^{-7}, 2^{-4}, 2^{-4}\}$
	0.19	2.61	2.86	2.07
ds2	2^{-3}	$\{2^{-2}, 2^{-7}\}$	$\{2^{-7}, 2^{2}, 2^{2}\}$	$\{2^{-3}, 2^{-4}, 2^{-4}\}$
	0.55	1.94	1.95	2.44
ds3	2^{4}	$\{2^{-4}, 2^{-5}\}$	$\{2^{-2}, 2^{-2}, 2^{-2}\}$	$\{2^{-4}, 2^{-2}, 2^{0}\}$
	0.76	1.78	1.87	1.92
ds4	2^{6}	$\{2^{-4}, 2^{-5}\}$	$\{2^{-2}, 2^{-1}, 2^{0}\}$	$\{2^{-5}, 2^{-7}, 2^{-7}\}$
	0.98	1.95	2.00	2.07
ds5	2^{2}	$\{2^{-4}, 2^{-6}\}$	$\{2^{-5}, 2^{-7}, 2^{-7}\}$	$\{2^{-5}, 2^{-5}, 2^{-7}\}$
	2.26	2.08	2.40	2.68
ds6	2^{6}	$\{2^{-2}, 2^{-3}\}$	$\{2^{-3}, 2^{-7}, 2^{-7}\}$	$\{2^{-2}, 2^{-2}, 2^{-4}\}$
	2.52	2.26	2.11	1.99
ds7	2^{2}	$\{2^{-3}, 2^{-3}\}$	$\{2^{-3}, 2^{-6}, 2^{-6}\}$	$\{2^{-7}, 2^{-1}, 2^{-7}\}$
	1.18	2.09	2.17	2.81
ds8	2^{3}	$\{2^{-3}, 2^{-7}\}$	$\{2^{-5}, 2^{4}, 2^{1}\}$	$\{2^{-3}, 2^{-3}, 2^{-3}\}$
	1.67	2.15	2.78	1.97
ds9	2^{-6}	$\{2^{-2}, 2^{-2}\}$	$\{2^{-7}, 2^{-2}, 2^{-7}\}$	$\{2^{-2}, 2^{-2}, 2^{-2}\}$
	1.37	3.01	4.46	2.84
ds10	2^{6}	$\{2^{-2}, 2^{-2}\}$	$\{2^{-6}, 2^{7}, 2^{2}\}$	$\{2^{-3}, 2^{-5}, 2^{-5}\}$
	2.11	2.00	2.00	1.96
ds11	2^{7}	$\{2^{-4}, 2^{0}\}$	$\{2^{-7}, 2^{7}, 2^{5}\}$	$\{2^{-7}, 2^{-3}, 2^{-4}\}$
	2.16	2.42	1.66	2.71
ds12	2^{6}	$\{2^{-2}, 2^{-2}\}$	$\{2^{-6}, 2^{-6}, 2^{-6}\}$	$\{2^{-6}, 2^{-6}, 2^{-6}\}$
	1.66	2.05	2.23	2.38
ds13	2^{-2}	$\{2^{0}, 2^{0}\}$	$\{2^{-5}, 2^{-2}, 2^{2}\}$	$\{2^{-2}, 2^{-2}, 2^{-2}\}$
	1.58	1.62	2.01	1.63
ds14	2^{6}	$\{2^{-7}, 2^{-2}\}$	$\{2^{-7}, 2^{0}, 2^{-4}\}$	$\{2^{-7}, 2^{1}, 2^{-5}\}$
	3.65	4.08	4.40	4.15
ds15	2^{4}	$\{2^{-1}, 2^{1}\}$	$\{2^{-7}, 2^{7}, 2^{-5}\}$	$\{2^{-2}, 2^{0}, 2^{2}\}$
	3.14	2.89	3.06	3.82
ds16	2^{0}	$\{2^{-2}, 2^{-6}\}$	$\{2^{-7}, 2^{7}, 2^{6}\}$	$\{2^{-7}, 2^{0}, 2^{-2}\}$
	3.97	5.07	5.90	7.97
ds17	2^{-1}	$\{2^{-2}, 2^{-2}\}$	$\{2^{-4}, 2^{3}, 2^{0}\}$	$\{2^{-6}, 2^{0}, 2^{0}\}$
	3.15	2.90	2.45	3.55

Appendix D. Best hyper-parameters of linear and nonlinear models

In Tables D.1 and D.2, we present the optimal hyper-parameter values and the corresponding CPU time required to solve the model across all datasets for the four studied models, in both their linear and nonlinear (Gaussian kernel) versions, respectively.

[1] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17(2-3):255–287, 2011.

[2] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[3] C. Carrasco, J. López, and S. Maldonado. Epsilon-nonparallel support vector regression. *Applied Intelligence*, 49:4223–4236, 2019.

Table D.2: **Gaussian Kernel**. Best combination of hyper-parameters and CPU time for all models.

Datasets	**SVM** $\{C, \sigma\}$ Time (s)	**PSVM** $\{\varepsilon, C, \sigma\}$ Time (s)	**CPSVM$_1$** $\{\varepsilon, C_1 = C_2, \sigma\}$ Time (s)	**CPSVM$_2$** $\{\varepsilon, C_1 = C_2, \sigma\}$ Time (s)
ds1	$\{2^2, 2^0\}$ 0.21	$\{2^{-3}, 2^{-4}, 2^1\}$ 0.35	$\{2^0, 2^6, 2^3\}$ 0.39	$\{2^{-3}, 2^4, 2^2\}$ 0.26
ds2	$\{2^3, 2^7\}$ 0.44	$\{2^{-3}, 2^{-4}, 2^7\}$ 0.71	$\{2^{-2}, 2^{-6}, 2^7\}$ 0.79	$\{2^{-5}, 2^{-7}, 2^7\}$ 0.71
ds3	$\{2^6, 2^3\}$ 0.24	$\{2^{-5}, 2^3, 2^7\}$ 0.80	$\{2^{-2}, 2^{-1}, 2^3\}$ 0.81	$\{2^{-4}, 2^{-1}, 2^4\}$ 0.57
ds4	$\{2^2, 2^0\}$ 0.34	$\{2^{-1}, 2^2, 2^0\}$ 1.09	$\{2^{-\vert}, 2^0, 2^3\}$ 0.66	$\{2^{-4}, 2^{-1}, 2^1\}$ 0.67
ds5	$\{2^3, 2^{-1}\}$ 1.14	$\{2^{-1}, 2^2, 2^{-1}\}$ 1.96	$\{2^{-5}, 2^{-6}, 2^4\}$ 2.59	$\{2^{-4}, 2^2, 2^1\}$ 2.48
ds6	$\{2^{-4}, 2^1\}$ 0.94	$\{2^{-1}, 2^{-7}, 2^1\}$ 4.52	$\{2^{-2}, 2^1, 2^5\}$ 4.09	$\{2^{-3}, 2^{-6}, 2^4\}$ 3.48
ds7	$\{2^5, 2^5\}$ 3.59	$\{2^{-3}, 2^2, 2^7\}$ 7.35	$\{2^{-4}, 2^{-6}, 2^6\}$ 5.82	$\{2^{-5}, 2^3, 2^7\}$ 6.06
ds8	$\{2^5, 2^5\}$ 6.15	$\{2^{-3}, 2^2, 2^7\}$ 10.14	$\{2^{-3}, 2^{-7}, 2^6\}$ 13.36	$\{2^{-2}, 2^5, 2^7\}$ 11.98
ds9	$\{2^2, 2^5\}$ 1.33	$\{2^{-2}, 2^{-4}, 2^5\}$ 8.84	$\{2^{-4}, 2^{-4}, 2^7\}$ 6.21	$\{2^{-1}, 2^1, 2^6\}$ 7.66
ds10	$\{2^6, 2^3\}$ 12.12	$\{2^{-2}, 2^6, 2^2\}$ 21.33	$\{2^{-4}, 2^1, 2^{-1}\}$ 13.71	$\{2^{-3}, 2^6, 2^3\}$ 21.06
ds11	$\{2^5, 2^{-1}\}$ 16.35	$\{2^{-5}, 2^0, 2^1\}$ 35.41	$\{2^{-2}, 2^{-2}, 2^{-5}\}$ 22.70	$\{2^{-7}, 2^{-6}, 2^1\}$ 40.33
ds12	$\{2^5, 2^{-4}\}$ 15.84	$\{2^{-3}, 2^6, 2^{-2}\}$ 31.42	$\{2^{-3}, 2^2, 2^{-7}\}$ 19.49	$\{2^{-4}, 2^3, 2^{-3}\}$ 28.94
ds13	$\{2^{-2}, 2^1\}$ 6.25	$\{2^{-1}, 2^{-6}, 2^0\}$ 45.87	$\{2^0, 2^{-4}, 2^0\}$ 44.18	$\{2^0, 2^{-4}, 2^1\}$ 47.41
ds14	$\{2^4, 2^{-1}\}$ 46.60	$\{2^{-2}, 2^3, 2^0\}$ 79.41	$\{2^0, 2^6, 2^{-2}\}$ 83.92	$\{2^{-4}, 2^6, 2^1\}$ 108.17
ds15	$\{2^2, 2^4\}$ 12.18	$\{2^{-3}, 2^0, 2^1\}$ 89.31	$\{2^{-1}, 2^{-1}, 2^{-2}\}$ 95.30	$\{2^{-2}, 2^{-1}, 2^0\}$ 87.51
ds16	$\{2^6, 2^5\}$ 370.61	$\{2^{-3}, 2^{-6}, 2^7\}$ 804.69	$\{2^{-2}, 2^{-5}, 2^7\}$ 1,012.69	$\{2^{-4}, 2^{-7}, 2^5\}$ 765.97
ds17	$\{2^2, 2^{-2}\}$ 426.92	$\{2^{-2}, 2^{-2}, 2^{-3}\}$ 1,252.51	$\{2^{-5}, 2^{-6}, 2^{-5}\}$ 945,83	$\{2^{-4}, 2^2, 2^{-3}\}$ 1,216.71

[4] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408:189–215, 2020.

[5] J. Demšar. Statistical comparisons of classifiers over multiple data set. *Journal of Machine Learning Research*, pages 1–30, 2006.

[6] N.Y. Deng, Y.J. Tian, and C.H. Zhang. *Support vector machines: optimization based theory, algorithms, and extensions*. Chapman and Hall/CRC, 2012.

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, Stanford, California, USA, second edition, 2009.

[8] X. Huang, L. Shi, and Johan A.K. Suykens. Ramp loss linear programming support vector machine. *The Journal of Machine Learning Research*, 15(1):21852211, 2014.

[9] X. Huang, L. Shi, and Johan A.K. Suykens. Support vector machine classifier with pinball loss. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):984–997, 2014.

[10] H.-T. Lin, C.-J. Lin, and R.C. Weng. A note on platt's probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276, 2007.

[11] J. López, S. Maldonado, and M. Carrasco. Robust nonparallel support vector machines via second-order cone programming. *Neurocomputing*, 364:227–238, 2019.

[12] S. Maldonado, J. López, and C. Vairetti. Profit-based churn prediction based on minimax probability machines. *European Journal of Operational Research*, 284(1):273–284, 2020.

[13] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers, MIT Press*, pages 61–74, 2000.

[14] N. Rtayli and N. Enneya. Enhanced credit card fraud detection based on svm-recursive feature elimination and hyper-parameters optimization. *Journal of Information Security and Applications*, 55:102596, 2020.

[15] Y.-H. Shao, X.-J. Lv, L.-W. Huang, and L. Bai. Twin svm for conditional probability estimation in binary and multiclass classification. *Pattern Recognition*, 136:109253, 2023.

[16] V. Vapnik and R. Izmailov. Synergy of monotonic rules. *Journal of Machine Learning Research*, 17:1–33, 2016.

[17] V. Vapnik and R. Izmailov. Reinforced svm method and memorization mechanisms. *Pattern Recognition*, 119:108018, 2021.

[18] J. Weston, A. Elisseeff, G. BakIr, and F. Sinz. The spider machine learning toolbox, 2005. Software available at `https://people.kyb.tuebingen.mpg.de/spider/`.

[19] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 694699, 2002.