A NOVEL TWIN BOUNDED SUPPORT VECTOR MACHINE WITH SMOOTH GENERALIZED PINBALL LOSS

PATCHARAPA SRICHOK

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Applied Mathematics

Suranaree University of Technology

Academic Year 2024

ซัพพอร์ตเวกเตอร์แมชชีนมีขอบเขตแบบคู่
ด้วยฟังก์ชันการสูญเสียพินบอลวางนัยทั่วไปแบบปรับเรียบ

นางสาวพัชราภา ศรีโชค

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาคณิตศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2567

A NOVEL TWIN BOUNDED SUPPORT VECTOR MACHINE WITH SMOOTH GENERALIZED PINBALL LOSS

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for a Master's Degree.

Thesis Examining Committee

Benjawan Rodjanadid
(Asst. Prof. Dr. Benjawan Rodjanadid)
Chairperson

Panu Yimmuang
(Asst. Prof. Dr. Panu Yimmuang)
Member (Thesis Advisor)

(Assoc. Prof. Dr. Eckart Robert Schulz)
Member (Thesis Co-advisor)

T. Areerak
(Asst. Prof. Dr. Tidarut Areerak)
Member

Tirawut Worrakitpoonpon
(Assoc. Prof. Dr. Tirawut Worrakitpoonpon)
Member

T. Grace
(Asst. Prof. Dr. Thanatporn Grace)
Member

Yupaporn
(Assoc. Prof. Dr. Yupaporn Ruksakulpiwat)
Vice Rector for Academic Affairs
and Quality Assurance

(Prof. Dr. Santi Maensiri)
Dean of Institute of Science

พัชราภา ศรีโชค : ซัพพอร์ตเวกเตอร์แมชชีนมีขอบเขตแบบคู่ด้วยฟังก์ชันการสูญเสียพินบอลวางนัยทั่วไปแบบปรับเรียบ (A NOVEL TWIN BOUNDED SUPPORT VECTOR MACHINE WITH SMOOTH GENERALIZED PINBALL LOSS) อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.ภาณุ ยิ้มเมือง, 68 หน้า.

คำสำคัญ: ฟังก์ชันการสูญเสียพินบอลวางนัยทั่วไปแบบปรับเรียบ/ซัพพอร์ตเวกเตอร์แมชชีน

เราได้นำเสนอชุดของฟังก์ชันการสูญเสียพินบอลวางนัยทั่วไปแบบปรับเรียบเพื่อแก้ไขปัญหาความไม่สามารถหาอนุพันธ์ได้ ความไวต่อสัญญาณรบกวน และความไม่เสถียรในการสุ่มตัวอย่างใหม่ ซึ่งเป็นข้อจำกัดของฟังก์ชันการสูญเสียแบบดั้งเดิม เช่น ฟังก์ชันการสูญเสียแบบบานพับ และฟังก์ชันการสูญเสียพินบอล โดยอ้างอิงจากแนวคิดนี้เราได้นำเสนอฟังก์ชันการสูญเสียพินบอลวางนัยทั่วไปแบบปรับเรียบตัวใหม่ซึ่งทำให้ฟังก์ชันวัตถุประสงค์ของแบบจำลองซัพพอร์ตเวกเตอร์แมชชีนสามารถหาอนุพันธ์ได้ ช่วยลดความไวต่อสัญญาณรบกวน และยังคงคุณสมบัติของความบางในคำตอบของแบบจำลอง ดังนั้นจึงเกิดแบบจำลองซัพพอร์ตเวกเตอร์แมชชีนมีขอบเขตแบบคู่ด้วยฟังก์ชันการสูญเสียพินบอลวางนัยทั่วไปแบบปรับเรียบขึ้น นอกจากนี้เราได้ทำการเปรียบเทียบประสิทธิภาพของซัพพอร์ตเวกเตอร์แมชชีนมีขอบเขตแบบคู่ด้วยฟังก์ชันการสูญเสียพินบอลวางนัยทั่วไปแบบปรับเรียบใหม่นี้กับวิธีการอื่น ๆ ที่มีอยู่โดยให้การประเมินอย่างครอบคลุมถึงจุดแข็งและข้อจำกัดของแบบจำลองผ่านการทดสอบด้วยชุดข้อมูล UCI ผลการทดลองแสดงให้เห็นว่าแบบจำลองที่นำเสนอให้ประสิทธิภาพที่ดีที่สุดในซัพพอร์ตเวกเตอร์แมชชีนมีขอบเขตแบบคู่เมื่อใช้ร่วมกันกับ RBFSampler นอกจากนี้เรายังแสดงให้เห็นว่าฟังก์ชันการสูญเสียพินบอลวางนัยทั่วไปสามารถประมาณค่าได้โดยใช้ฟังก์ชันการสูญเสียพินบอลวางนัยทั่วไปแบบปรับเรียบตัวใหม่นี้ภายใต้นอร์มแบบเอกรูปด้วยความแม่นยำตามต้องการ และคำตอบของแบบจำลองของเรายังมีเพียงคำตอบเดียวและลู่เข้าสู่คำตอบที่ถูกต้อง

สาขาวิชาคณิตศาสตร์และภูมิสารสนเทศ ลายมือชื่อนักศึกษา พัชราภา ศรีโชค

ปีการศึกษา 2567 ลายมือชื่ออาจารย์ที่ปรึกษา ภาณุ ยิ้มเมือง

PATCHARAPA SRICHOK : A NOVEL TWIN BOUNDED SUPPORT VECTOR MACHINE WITH SMOOTH GENERALIZED PINBALL LOSS. THESIS ADVISOR : ASST. PROF. PANU YIMMUANG, Ph.D. 68 PP.

Keyword: SMOOTH GENERALIZED PINBALL LOSS FUNCTION, SUPPORT VECTOR MACHINE

We present a one-parameter family of smooth generalized pinball loss functions to overcome the challenges of non-differentiability, noise sensitivity, and resampling instability inherent in traditional loss functions such as hinge loss and pinball loss. These functions make the objective function in the formulation of the support vector machine (SVM) model twice continuously differentiable and improve model performance by reducing noise sensitivity and preserving the sparsity of the solution. In a similar way, a novel twin bounded support vector machine (TBSVM) model with smooth generalized pinball loss function is obtained. Furthermore, we compare the performance of the TBSVM with the novel type of smooth loss function against other contemporary approaches, offering a comprehensive assessment of its strengths and limitations by conducting an evaluation with UCI datasets. The experimental results show that the proposed model has the best performance in the TBSVM with RBFSampler. Additionally, we prove that the generalized pinball loss function can be approximated by a novel smooth generalized pinball loss function in the uniform norm with arbitrary precision. We further show that the solutions of the proposed SVM and TBSVM models are unique and that they converge to the solutions of the models with non-smooth generalized pinball loss as the parameter approaches zero.

School of Mathematical Sciences
and Geoinformatics
Academic Year 2024

Student's Signature พัชราภา ศรีโชค
Advisor's Signature ภาณุ ยิ้มเมือง

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to my project advisor, Assistant Professor Dr. Panu Yimmuang, for his initial idea, support and encouragement throughout this research. Thank you for all your guidance, comments, suggestion and support, you have given to me during this research.

Furthermore, I would like to give my special thanks for the professional support received from Associate Professor Dr. Eckart Robert Schulz and all professors in the School of Mathematical Sciences and Geoinformatics, Institute of Science, Suranaree University of Technology (SUT).

I would also like to thank my family who have always been there for me. I can not forget my senior friends and my friends for their helps and great relationships.

Lastly, I gratefully acknowledge the funding received towards my undergraduate from Development and Promotion of Science and Technology Talents Project (DPST) Scholarship.

Patcharapa Srichok

CONTENTS

CONTENTS (Continued)

CONTENTS (Continued)

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The Support Vector Machine (SVM), as introduced by Vapnik and Cortes (1995), is a machine learning technique founded on the principles of the Vapnik-Chervonenkis (VC) dimension and structural risk minimization theory within the realm of statistical learning. It is a powerful and widely used supervised machine learning algorithm that is primarily employed for classification and regression tasks, such as text categorization (Joachims, 1998), or scene classification (Yin, Jiao, and Chai, 2015), to name just a few examples. When applied to classification, this approach employs a strategy of maximizing the distances between two distinct data classes from a separating hyperplane, ensuring the correct classification of the two training datasets with a high level of confidence. Through the introduction of "slack variables" and the "kernel trick", SVMs are particularly suited for their ability to effectively handle high-dimensional data and complex, non-linear decision boundaries.

The Twin Support Vector Machine (TWSVM), an innovative extension of the traditional SVM, has garnered considerable attention for its potential to address complex data distributions. Khemchandani and Chandra (2007) proposed the TWSVM, which consists of two non-parallel hyperplanes, each positioned in close proximity to one of the two classes while maintaining a minimum separation distance from the other class. The TWSVM reduces the algorithmic complexity to just a quarter of that of the standard SVM, resulting in a significant reduction of computational time. Shao and his research team (2011) modified the TWSVM to the Twin Bounded Support Vector Machine (TBSVM), aiming to minimize structural risk and leading to improved general performance.

Creating the hyperplanes in any of the support vector machine models involves solving a constrained convex minimization problem. The most common solution technique is to formulate the dual optimization problem and solve it using the method of Lagrange multipliers. More recently, techniques for solving the primal problem directly

have become popular. These methods use a "loss function" to reformulate the problem as an unconstrained optimization problem. Common loss functions include the hinge loss, the pinball loss, and the generalized pinball loss functions, listed in order of complexity. Because these loss functions are not differentiable, efficient numerical methods such as gradient descent or Newton-methods cannot be applied, however.

In 2023, Makmuang, Ratiphaphongthon, and Wangkeeree introduced a smooth approximation to the generalized pinball loss function within the standard SVM framework. The results demonstrate that, on average, the proposed method exhibits superior performance compared to the baseline models. Similarly, Kai and Zhen (2021) introduced a smooth approximation to the pinball loss function in the twin bounded support vector machine model to mitigate the noise sensitivity and resampling instability associated with the hinge loss function.

This thesis primarily concentrates on conducting an extensive comparative analysis to investigate the impact of various loss functions and their generalization on model performance. It also introduces a new one parameter family of smooth approximations of the generalized pinball loss into the TBSVM model. By comparing the TBSVM equipped with this novel loss function against other contemporary approaches, it aims to provide a comprehensive perspective on the strengths and limitations of this methodology. Additionally, it proves that the generalized pinball loss function can be arbitrarity approximated by a one of the smooth loss functions in the uniform norm.

1.1 Research Objectives

1. To introduce and study a new C^2-smooth loss function and study its properties.

2. To modify the twin bounded support vector machine model to include the new loss function.

3. To evaluate the performance of the proposed TBSVM model with other representative models on selected UCI datasets.

1.2 Scope and Limitations

The Scopes of the research work is as follows:

1. We select UCI datasets to study the proposed model, and select other representative loss functions for comparisons to validate and test the effectiveness of the proposed model.
2. All experiments use the same numerical algorithm which is based on the Newton-Armijo approach.
3. The Python language is used.

1.3 Research Procedure

The research work proceeded as follows:

1. Study the theory of support vector machine (SVM), twin support vector machine (TWSVM) and twin bounded support vector machine (TBSVM).
2. Study loss functions for SVMs.
3. Study the Newton-Armijo iteration method.
4. Apply a novel smooth approximation to the generalized pinball loss function into the twin bounded support vector machine.
5. Prove a theorem on convergence of the proposed algorithm.
6. Implement the algorithm to the selected UCI datasets and analyze the experimental results.

1.4 Results

In this thesis, we

1. Present a TBSVM that incorporates our novel 1-parameter family of C^2-smooth generalized pinball loss functions .

2. Show that as the model parameter approaches zero, our proposed TBSVM model converges to the TBSVM model with generalized pinball loss function.

3. Show by experiments that our proposed TBSVM model is competitive with two TBSVM models incorporating different loss functions in term of accuracy and insensitivity to noise, and in some instances is superior.

CHAPTER II

LITERATURE REVIEW

This chapter introduces the fundamental mathematical concepts in machine learning as they pertain to the support vector machine models to be used in the thesis work. We only present the linear models for simplicity of exposition; all models presented can be adapted to include the kernel-trick which allows for non-linear separating boundaries.

The statistical significance of the performance values attained by the various models is also evaluated using the Friedman test.

2.1 Linear Separability

In its simplest form when used for classification, the support vector machine deals with two classes of data, denoted as class $+1$ and class -1. Mathematically speaking, a data sample is a vector in Euclidean space $\mathbb{R}^n$, and the data is considered linearly separable if there exists a hyperplane that separates the data into its two classes: data samples from class $+1$ lie on one side of the hyperplane, and samples from class -1 lie on the other side. By a hyperplane, we mean a shifted $(n-1)-$dimensional subspace of $\mathbb{R}^n$. Its equation is:

$$f(\overrightarrow{x}) = \overrightarrow{w}^T \overrightarrow{x} + b = 0,$$

where $\overrightarrow{x}$ is any vector on the hyperplane and $\overrightarrow{w}$ is a vector perpendicular to the hyperplane. Note that $\overrightarrow{w}$ is unique up to a scalar multiple and essentially determines the direction of the hyperplane, while b fixes the position of the hyperplane in $\mathbb{R}^n$ space.

In one-dimensional space, a hyperplane is simply a point. In two-dimensional space, a hyperplane is a line, and in three-dimensional space, it is a plane. Hence the name “hyperplane”.

2.2 Support Vector Machine

Consider a dataset $X = \{(\vec{x_i}, y_i)|i = 1, 2, ..., m\}, \vec{x_i} \in \mathbb{R}^n, y_i \in \{+1, -1\}$. It comprises m_1 positive samples represented in form of an $m_1 \times n$ matrix A, and m_2 negative samples represented by an $m_2 \times n$ matrix B, with $m_1 + m_2 = m$. A support vector machine (SVM) finds the optimal hyperplane $f(x) = \vec{w}^T\vec{x} + b = 0$ which separates the data into its two classes and has larges distance (maximal margin) from two classes of samples. Vectors $x_i^{(1)}$ and $x_i^{(2)}$ in the respective class closest to the hyperplane are called support vectors as shown below

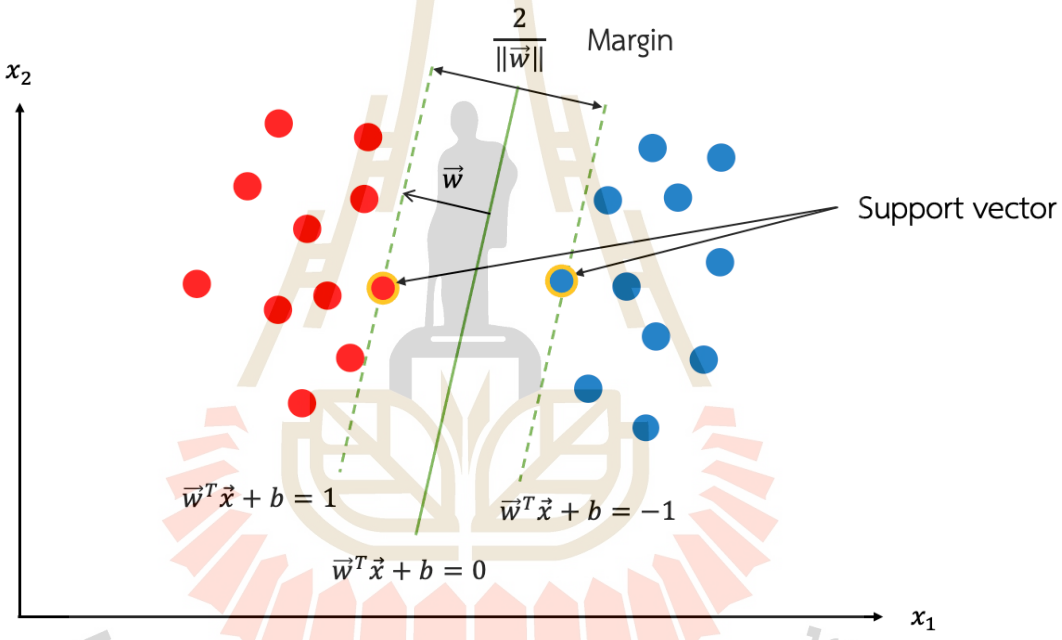


Figure 2.1 Support vector machine.

The SVM model takes the following form:

$$\min_{\vec{w},b} \ \frac{1}{2}\|\vec{w}\|^2 \qquad (2.1)$$

$$\text{s.t.} \ \ y_i(\vec{w}^T\vec{x_i} + b) \geq 1, \quad (i = 1, ..., m).$$

This formulation implies normalization of the vector $\vec{w}$ so that the margin is $\frac{2}{\|\vec{w}\|}$.

If the training data is nearly linearly separable, except for some outliers, then slack variables $\xi_1, \ldots, \xi_m \in \mathbb{R}$ are introduced. Problem (2.1) is modified to the convex quadratic programming problem as

$$\begin{aligned} \min_{\vec{w},b,\xi_i} \quad & \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t. } & y_i(\vec{w}^T\vec{x_i} + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \quad (i = 1, \ldots, m), \end{aligned} \tag{2.2}$$

where $y_i \in \{1, -1\}$, and $C > 0$ is a penalty parameter which manages the equilibrium between maximizing the margin (that is, minimizing $\|w\|$) and minimizing classification errors.

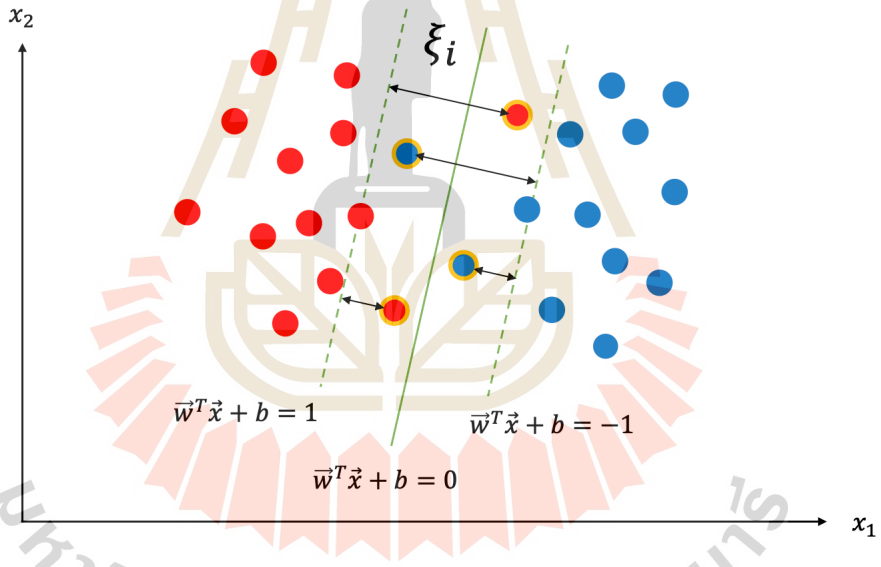


Figure 2.2 Support vector machine with slack variables.

The optimization problem (2.2) of a linear SVM can be written in matrix notation as

$$\begin{aligned} \min_{\vec{w},b,\vec{\xi_1},\vec{\xi_2}} \quad & \frac{1}{2}\|\vec{w}\|^2 + C(\vec{e_1}^T\vec{\xi_1} + \vec{e_2}^T\vec{\xi_2}) \\ \text{s.t. } & A\vec{w} + b\vec{e_1} \geq \vec{e_1} - \vec{\xi_1}, \quad (\vec{\xi_1} \geq 0), \\ & -(B\vec{w} + b\vec{e_2}) \geq \vec{e_2} - \vec{\xi_2}, \quad (\vec{\xi_2} \geq 0), \end{aligned} \tag{2.3}$$

where $C > 0$ is a penalty parameter, $\overrightarrow{\xi_1} \in \mathbb{R}^{m_1}$ and $\overrightarrow{\xi_2} \in \mathbb{R}^{m_2}$ are column vectors of slack variables, $\overrightarrow{e_1}$ and $\overrightarrow{e_2}$ are column vectors of ones of dimensions m_1, respectively m_2, and A and B are the $m_1 \times n$ and $m_2 \times n$ matrices contain the data of the $+1$ and -1 classes, respectively. The decision function of a linear SVM is the sign function (sgn) designed as

$$\text{class}(x) = \text{sgn}(\overrightarrow{w}^T x + b). \tag{2.4}$$

2.3 Twin Support Vector Machine

Consider a binary classification dataset X which contains m_1 positive (class $+1$) and m_2 negative (class -1) samples respectively, and represented by matrices A and B respectively. In a Twin Support Vector Machine (TWSVM), two non-parallel hyperplanes are located in a manner where each hyperplane is nearer to one of the two classes and maintains a minimum distance of at least one from the other. The two nonparallel classification hyperplanes are defined as

$$f_1(\overrightarrow{x}) = \overrightarrow{w_1}^T \overrightarrow{x} + b_1 = 0 \quad \text{and} \quad f_2(\overrightarrow{x}) = \overrightarrow{w_2}^T \overrightarrow{x} + b_2 = 0,$$

which need to separate the samples correctly, while each of these two hyperplane has only one supporting hyperplane, namely

$$f_1^-(\overrightarrow{x}) = \overrightarrow{w_1}^T \overrightarrow{x} + b_1 = -1 \quad \text{and} \quad f_2^+(\overrightarrow{x}) = \overrightarrow{w_2}^T \overrightarrow{x} + b_2 = 1,$$

respectively.

Introducing slack variables again, the TWSVM is determined by the following two optimization problems

$$\min_{\overrightarrow{w_1}, b_1, \xi_i} \frac{1}{2} \sum_{i=1}^{m_1} (\overrightarrow{w_1}^T \overrightarrow{x_i}^{(1)} + b_1)^2 + c_1 \sum_{i=1}^{m_2} \xi_i \tag{2.5}$$

$$\text{s.t.} \ -(\overrightarrow{w_1}^T \overrightarrow{x_1}^{(2)} + b_1) \geq 1 - \xi_i, \quad \xi_i \geq 0, \ \ i = 1, 2, 3 \ldots, m_2,$$

$$\min_{\overrightarrow{w_2}, b_2, \eta_i} \frac{1}{2} \sum_{i=1}^{m_2} (\overrightarrow{w_2}^T \overrightarrow{x_i}^{(2)} + b_2)^2 + c_2 \sum_{i=1}^{m_1} \eta_i \tag{2.6}$$

$$\text{s.t.} \ (\overrightarrow{w_2}^T \overrightarrow{x_i}^{(1)} + b_2) \geq 1 - \eta_i, \quad \eta_i \geq 0, \ \ i = 1, 2, 3 \ldots, m_1,$$

where $\overrightarrow{x_i}^{(1)}$ and $\overrightarrow{x_i}^{(2)}$ denote the data samples in the positive and negative class, respectively, $c_1, c_2 > 0$ are penalty parameters, and the ξ_i and η_i are slack variables.

The TWSVM model can be written in vector form,

$$\min_{\overrightarrow{w_1}, b_1, \overrightarrow{\xi_2}} \quad \frac{1}{2}\|A\overrightarrow{w_1} + b_1\overrightarrow{e_1}\|^2 + c_1\overrightarrow{e_2}^T\overrightarrow{\xi_2} \tag{2.7}$$

$$\text{s.t.} \quad -(B\overrightarrow{w_1} + b_1\overrightarrow{e_2}) \geq \overrightarrow{e_2} - \overrightarrow{\xi_2}, \quad (\overrightarrow{\xi_2} \geq 0),$$

$$\min_{\overrightarrow{w_2}, b_2, \overrightarrow{\xi_1}} \quad \frac{1}{2}\|B\overrightarrow{w_2} + b_2\overrightarrow{e_2}\|^2 + c_2\overrightarrow{e_1}^T\overrightarrow{\xi_1} \tag{2.8}$$

$$\text{s.t.} \quad (A\overrightarrow{w_2} + b_2\overrightarrow{e_1}) \geq \overrightarrow{e_1} - \overrightarrow{\xi_1}, \qquad (\overrightarrow{\xi_1} \geq 0),$$

where $c_1, c_2 > 0$ are penalty parameters, $\overrightarrow{\xi_1} \in \mathbb{R}^{m_1}$ and $\overrightarrow{\xi_2} \in \mathbb{R}^{m_2}$ are column vectors of slack variables, $\overrightarrow{\xi_1} = (\xi_1, ..., \xi_{m_2})^T$ and $\overrightarrow{\xi_2} = (\eta_1, ..., \eta_{m_1})^T$, and $\overrightarrow{e_1}$ and $\overrightarrow{e_2}$ are column vectors of correspondingly sized ones. The quadratic term in the objective functions in (2.7) and (2.8) aim to minimize the distance between a data sample and the hyperplane of its class. The inequality constraint in (2.5), respectively (2.7), ensures that a data sample from the negative class lies beyond the hyperplane $f_1^-(x) = -1$, and the inequality constraint in (2.6), respectively (2.8), ensures that a data sample from the positive class lies beyond the hyperplane $f_2^+(x) = 1$.

The decision function of linear TWSVM is

$$\text{class}(\overrightarrow{x}) = \text{sign}\left(\frac{\overrightarrow{w_1}^T\overrightarrow{x} + b_1}{\|\overrightarrow{w_1}\|} + \frac{\overrightarrow{w_2}^T\overrightarrow{x} + b_2}{\|\overrightarrow{w_2}\|}\right). \tag{2.9}$$

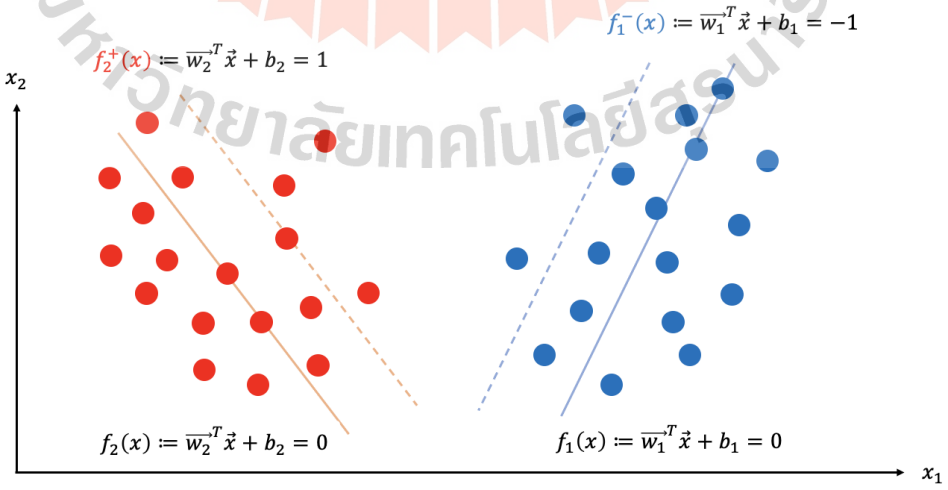


Figure 2.3 Twin support vector machine with supporting hyperplanes.

2.4 Twin Bounded Support Vector Machine

The objective function of the TWSVM saw the integration of a regularization term by Shao and his research team in 2011, resulting in the introduction of a novel machine learning model called the twin bounded support vector machine (TBSVM).

Let $\overrightarrow{x}^{(1)}$ and $\overrightarrow{x}^{(2)}$ denote data samples on the hyperplanes $f_1^-(x) = -1$ and $f_2^+(x) = 1$, respectively. Then the distances of these points from the hyperplanes $f_1(x) = 0$ and $f_2(x) = 0$, respectively, are

$$d(\overrightarrow{x}^{(1)}, f_1(x)) = \frac{|\overrightarrow{w_1}^T \overrightarrow{x}^{(1)} + b_1|}{\|\overrightarrow{w_1}\|} = \frac{|-1|}{\|\overrightarrow{w_1}\|} = \frac{1}{\|\overrightarrow{w_1}\|},$$

$$d(\overrightarrow{x}^{(2)}, f_2(x)) = \frac{|\overrightarrow{w_2}^T \overrightarrow{x}^{(2)} + b_2|}{\|\overrightarrow{w_2}\|} = \frac{|1|}{\|\overrightarrow{w_2}\|} = \frac{1}{\|\overrightarrow{w_2}\|}.$$

The TBSVM wants to maximize these distances, i.e. keep the samples from the "opposite class" as far away from the classifying hyperplane.

Thus, the objective functions for the TBSVM contain an additional quadratic term $\|\overrightarrow{w_i}\|^2 + b_i^2$:

$$\min_{\overrightarrow{w_1}, b_1, \xi_i} \frac{1}{2}\sum_{i=1}^{m_1}(\overrightarrow{w_1}^T \overrightarrow{x_i}^{(1)} + b_1)^2 + c_3(\|\overrightarrow{w_1}\|^2 + b_1^2) + c_1 \sum_{i=1}^{m_2} \xi_i \tag{2.10}$$

$$\text{s.t. } -(\overrightarrow{w_1}^T \overrightarrow{x_1}^{(2)} + b_1) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, 2, 3 \ldots, m_2,$$

$$\min_{\overrightarrow{w_2}, b_2, \eta_i} \frac{1}{2}\sum_{i=1}^{m_2}(\overrightarrow{w_2}^T \overrightarrow{x_i}^{(2)} + b_2)^2 + c_4(\|\overrightarrow{w_2}\|^2 + b_2^2) + c_2 \sum_{i=1}^{m_1} \eta_i \tag{2.11}$$

$$\text{s.t. } (\overrightarrow{w_2}^T \overrightarrow{x_i}^{(1)} + b_2) \geq 1 - \eta_i, \quad \eta_i \geq 0, \quad i = 1, 2, 3 \ldots, m_1,$$

where $c_1, c_2, c_3, c_4 > 0$ are penalty parameters, ξ_i and η_i are slack variables.

The TBSVM can be expressed in vector form as follows:

$$\min_{\overrightarrow{w_1}, b_1, \overrightarrow{\xi_2}} \frac{1}{2}\|A\overrightarrow{w_1} + b_1\overrightarrow{e_1}\|^2 + c_3\left(\|\overrightarrow{w_1}\|^2 + b_1^2\right) + c_1 \overrightarrow{e_2}^T \overrightarrow{\xi_2} \tag{2.12}$$

$$\text{s.t. } -(B\overrightarrow{w_1} + b_1\overrightarrow{e_2}) \geq \overrightarrow{e_2} - \overrightarrow{\xi_2}, \qquad \overrightarrow{\xi_2} \geq 0,$$

$$\min_{\overrightarrow{w_2}, b_2, \overrightarrow{\xi_1}} \frac{1}{2}\|B\overrightarrow{w_2} + b_2\overrightarrow{e_2}\|^2 + c_4\left(\|\overrightarrow{w_2}\|^2 + b_2^2\right) + c_2 \overrightarrow{e_1}^T \overrightarrow{\xi_1} \tag{2.13}$$

$$\text{s.t. } A\overrightarrow{w_2} + b_2\overrightarrow{e_1} \geq \overrightarrow{e_1} - \overrightarrow{\xi_1}, \qquad \overrightarrow{\xi_1} \geq 0,$$

where $c_1, c_2, c_3, c_4 > 0$ are penalty parameters, $\overrightarrow{\xi_1} \in \mathbb{R}^{m_1}$ and $\overrightarrow{\xi_2} \in \mathbb{R}^{m_2}$ are column vectors of slack variables, $\overrightarrow{e_1}$ and $\overrightarrow{e_2}$ are column vectors of appropriately sized ones.

However, there are several key differences between TBSVM and TWSVM:

1. In the primal problems of TWSVM, the focus is on minimizing the empirical risk, whereas in TBSVM, structural risk is minimized while also incorporating a regularization term aimed at maximizing the margin.

2. The dual problems corresponding to TBSVM's primal problems can be derived without additional assumptions or modifications, making this approach more rigorous and complete than TWSVM.

3. To reduce training time, an efficient method known as successive over-relaxation (SOR) is applied in TBSVM.

2.5 The SVM Model as an Unconstrained Optimization Problem

We now explain the concept of loss function by the simplest model, the SVM model (2.2); the twin SVM models can be modified in a similar way.

Since the objective function is to be minimized, solving the constraints in (2.2) for ξ_i leads to the optimization problem

$$\min_{\overrightarrow{w},b} \quad \frac{1}{2}\|\overrightarrow{w}\|^2 + C\sum_{i=1}^{m} \max\{0, 1 - y_i(\overrightarrow{w}^T\overrightarrow{x_i} + b)\}$$

or equivalently, to

$$\min_{\overrightarrow{w},b} \quad \frac{1}{2}\|\overrightarrow{w}\|^2 + C\sum_{i=1}^{m} L_{\text{hinge}}\big(1 - y_i(\overrightarrow{w}^T\overrightarrow{x_i} + b)\big), \tag{2.14}$$

where

$$L_{\text{hinge}}(u) = \max\{0, u\}$$

or equivalently,

$$L_{\text{hinge}}(u) = \begin{cases} u, & u \geq 0, \\ 0, & u < 0. \end{cases} \tag{2.15}$$

In fact, if one defines $L : \mathbb{R}^n \times \mathbb{R} \to [0, \infty)$ by

$$L(\overrightarrow{w}, b) = \sum_{i=1}^{m} L_{\text{hinge}}\big(1 - y_i(\overrightarrow{w}^T \overrightarrow{x_i} + b)\big)$$

then (2.14) becomes

$$\min_{\overrightarrow{w},b} \quad \frac{1}{2}\|\overrightarrow{w}\|^2 + CL(\overrightarrow{w}, b). \tag{2.16}$$

L is called a *loss function* or *cost function*, as it specifies the cost caused by the discrepancy between observation and prediction. Equation (2.16) states that the margin is to be maximized while the cost is to be minimized; the parameter C determines the balance between those two competing goals.

2.6 Loss Functions

The function L_{hinge} above is called the *hinge loss function*, as its graph looks like a hinge:

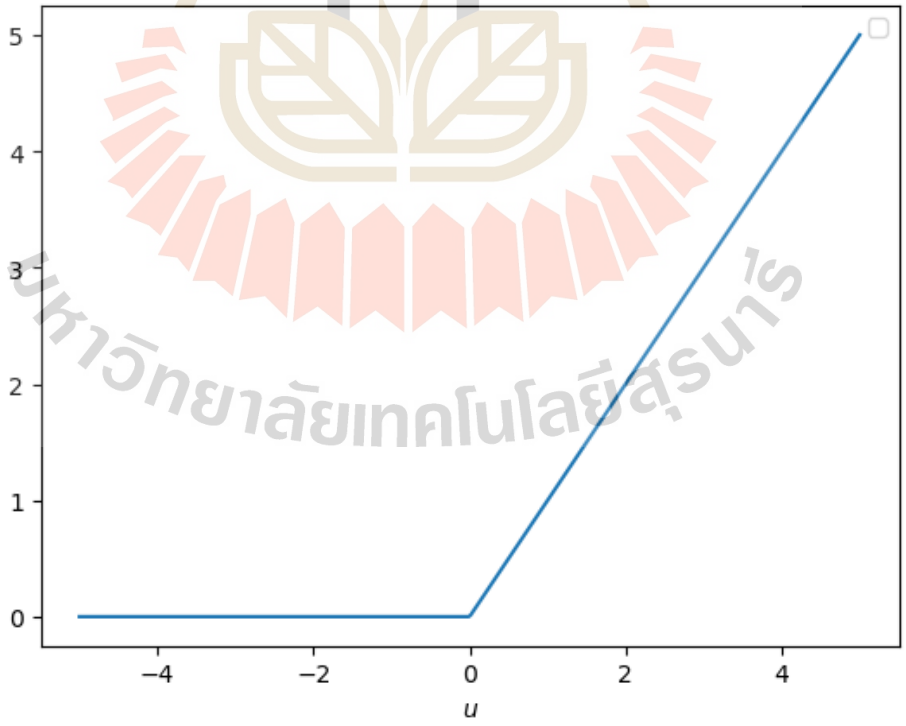


Figure 2.4 Hinge loss function.

The hinge loss function is instable for resampling and sensitive to noise. To resolve this problem, in 2013, Huang, Shi, and Suykens (2013) modified it to the pinball loss function in the SVM classifier, and demonstrated its effectiveness in mitigating noise sensitivity by experiments. The pinball loss function is defined as follows

$$L_\tau(u) = \begin{cases} u, & u \geq 0, \\ -\tau u, & u < 0, \end{cases} \tag{2.17}$$

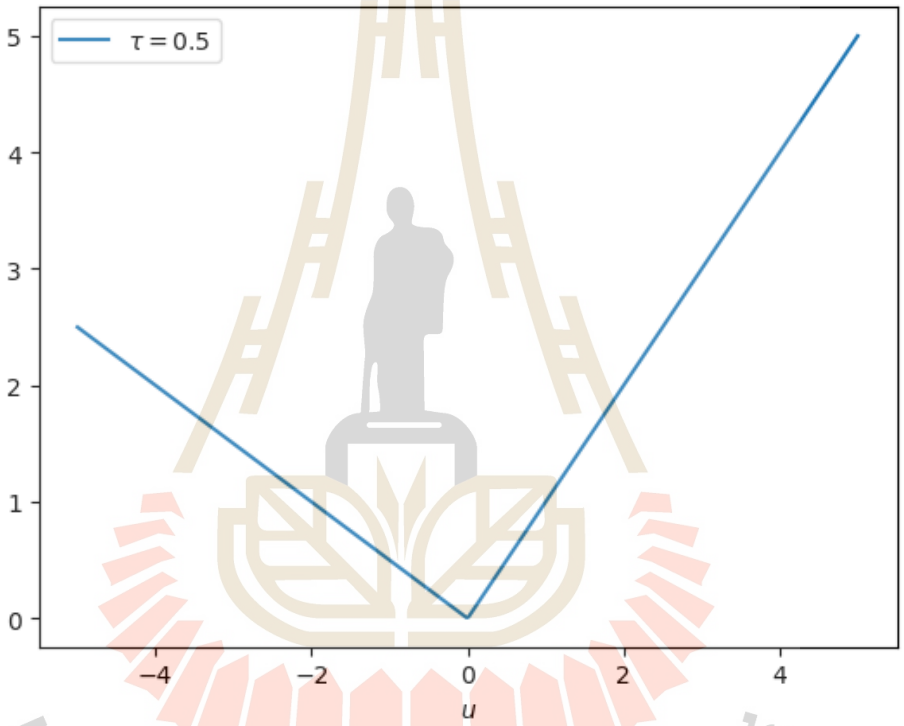


Figure 2.5 Pinball loss function.

where τ is non-negative real parameter, usually chosen less than one. However, the pinball loss function still shows some noise sensitivity. Thus, the ϵ-insensitive pinball loss function was created, defined as

$$L_\tau^\epsilon(u) = \begin{cases} u - \epsilon, & u > \epsilon, \\ 0, & -\frac{\epsilon}{\tau} \leq u \leq \epsilon, \\ -\tau(u + \frac{\epsilon}{\tau}), & u < -\frac{\epsilon}{\tau}, \end{cases} \tag{2.18}$$

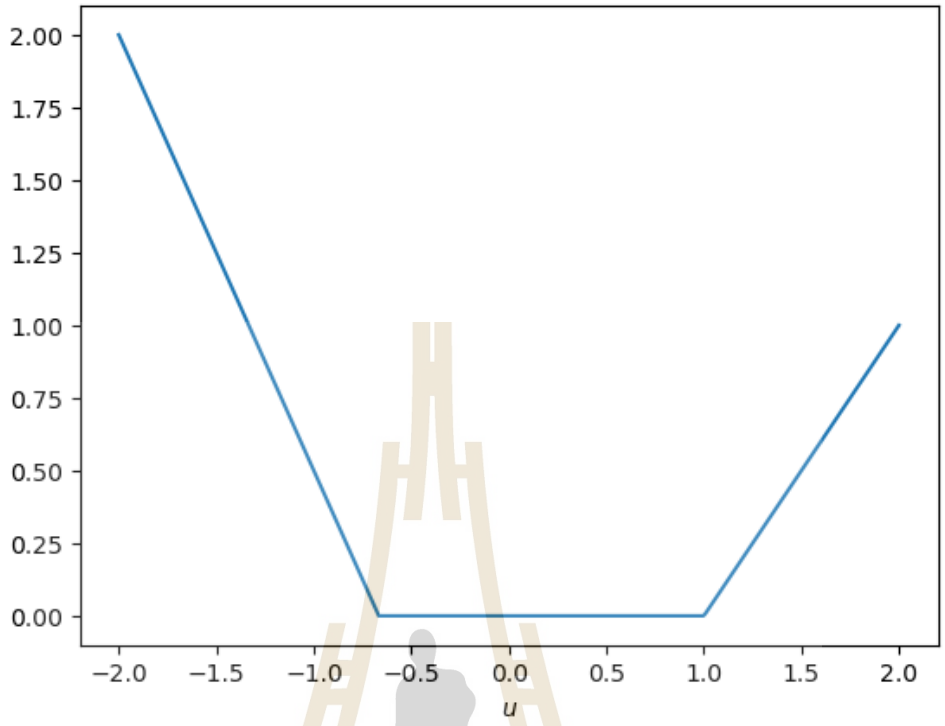


Figure 2.6 ϵ-insensitive pinball loss function ($\tau = 1.5$ and $\epsilon = 1$).

where τ, ϵ are non-negative real values. Recently, Rastogi, Pal, and Chandra (2018) introduced a novel loss function generalizing the ϵ-insesitive pinball loss by introducing one more parameter. This innovative loss function is referred to as the generalized pinball loss, offering attributes such as noise insensitivity, sparsity, stability for resampling, and is defined as follows,

$$L^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(u) = \begin{cases} \tau_1(u - \frac{\epsilon_1}{\tau_1}), & u > \frac{\epsilon_1}{\tau_1}, \\ 0, & -\frac{\epsilon_2}{\tau_2} \leq u \leq \frac{\epsilon_1}{\tau_1}, \\ -\tau_2(u + \frac{\epsilon_2}{\tau_2}), & u < -\frac{\epsilon_2}{\tau_2}, \end{cases} \tag{2.19}$$

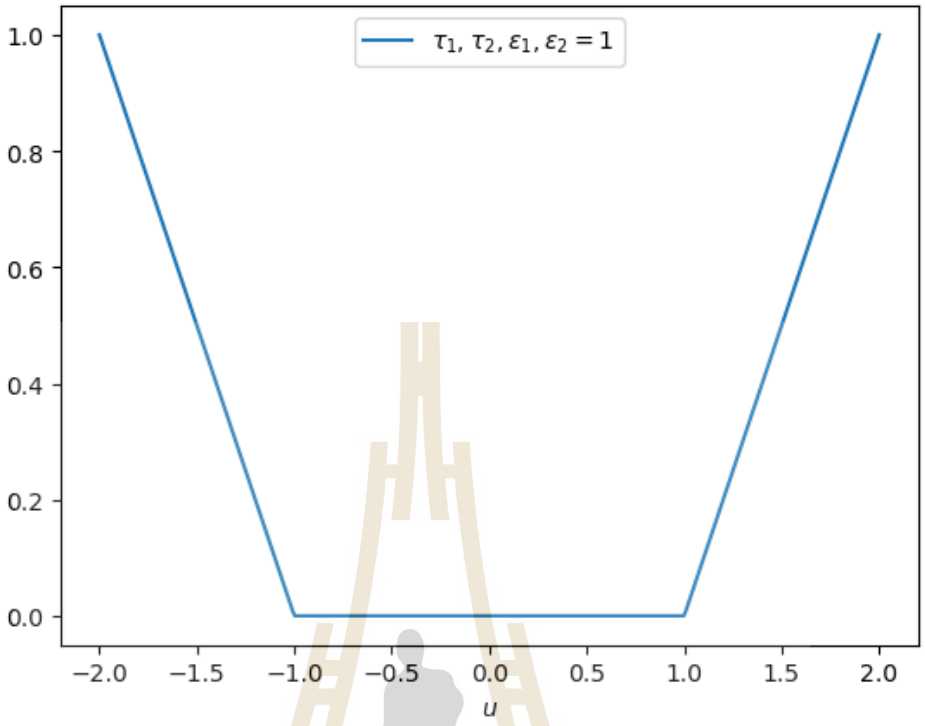


Figure 2.7 Generalized pinball loss function.

where $\tau_1, \tau_2, \epsilon_1, \epsilon_2$ are non-negative real values.

Recently, Makmuang, Ratiphaphongthon and Wangkeeree (2023) introduced a new C^1-smooth loss function approximating the generalized pinball loss function into the SVM. They substituted $L^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(u)$ by its Moreau proximal approximation represented as $Q^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2} : \mathbb{R} \times \mathbb{R}_+ \to \mathbb{R}_+$, and which is given by

$$Q^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(u,\mu) = \begin{cases} \tau_1(u - \frac{\epsilon_1}{\tau_1}) - \frac{\tau_1^2}{2}\mu, & \frac{\epsilon_1}{\tau_1} + \tau_1\mu \leq u, \\ \frac{1}{2\mu}(u - \frac{\epsilon_1}{\tau_1})^2, & \frac{\epsilon_1}{\tau_1} \leq u \leq \frac{\epsilon_1}{\tau_1} + \tau_1\mu, \\ 0, & -\frac{\epsilon_2}{\tau_2} \leq u \leq \frac{\epsilon_1}{\tau_1}, \\ \frac{1}{2\mu}(u + \frac{\epsilon_2}{\tau_2})^2, & -\frac{\epsilon_2}{\tau_2} - \tau_2\mu \leq u \leq -\frac{\epsilon_2}{\tau_2}, \\ -\tau_2(u + \frac{\epsilon_2}{\tau_2}) - \frac{\tau_2^2}{2}\mu, & u \leq -\frac{\epsilon_2}{\tau_2} - \tau_2\mu, \end{cases} \tag{2.20}$$

where $\tau_1, \tau_2, \epsilon_1, \epsilon_2$ are non-negative real values, $u \in \mathbb{R}$, and $\mu \in \mathbb{R}_+$ is a parameter of how close $Q^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(\cdot,\mu)$ approximates $L^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(\cdot)$. Figure 2.8 displays the graphic representation of $Q^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(u,\mu)$ for various values of the parameter μ.

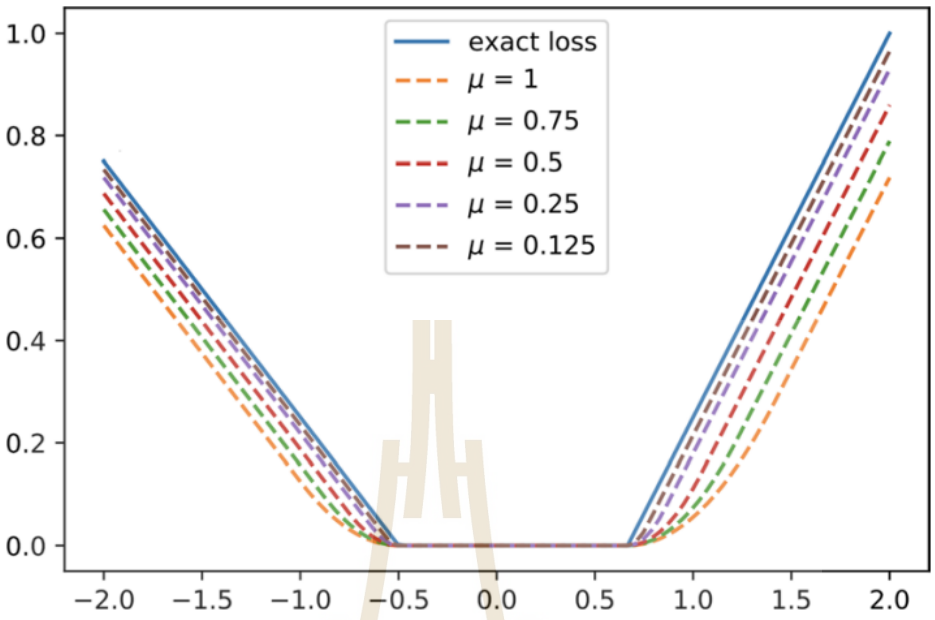


Figure 2.8 Graph of $L^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(\cdot)$ and $Q^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(\cdot,\mu)$ with differnt μ.

Alternatively, Kai and Zhen (2021) proposed an infinitely differentiable approximation of the pinball loss function within the twin bounded support vector machine model to mitigate the noise sensitivity and resampling instability inherent in the pinball loss function. Their smooth approximation of the pinball loss function is defined as

$$\phi_\tau(u,\epsilon) = \frac{u + \sqrt{u^2 + 4\epsilon^2}}{2} + \frac{-\tau u + \sqrt{\tau^2 u^2 + 4\epsilon^2}}{2}, \tag{2.21}$$

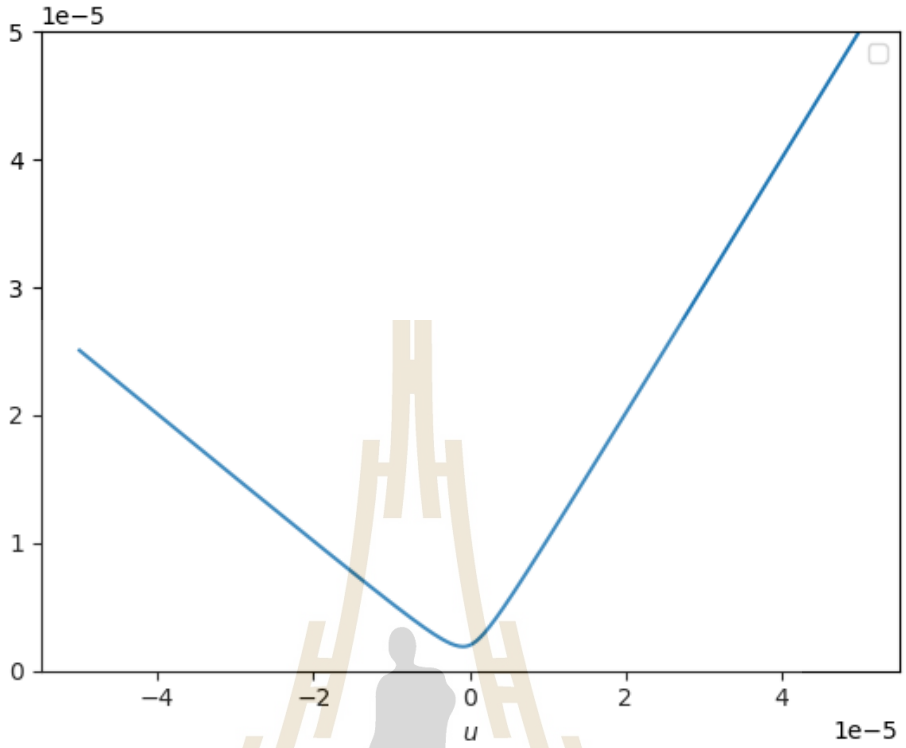


Figure 2.9 Smooth approximation of the pinball loss function ($\tau = 0.5$, and $\epsilon = 10^{-6}$).

where τ, ϵ are non-negative real values.

2.7 The BFGS Method

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) method is one of the most widely used quasi-Newton algorithms, named after its developers. BFGS iteratively updates an approximation of the Hessian matrix to efficiently converge toward an optimal solution. This section provides a derivation of the BFGS algorithm used to solve our problems.

We begin the derivation by forming the second-order Taylor polynomial of the objective function at the current iterate, x_k:

$$p_k(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T F(x_k)(x - x_k), \tag{2.22}$$

where $F(x_k)$ is the Hessian matrix of f that will be updated at every iteration. In Newton's Method, when choosing the next iteration as

$$x_{k+1} = x_k + d_k, \qquad d_k = -F(x_k)^{-1}\nabla f(x_k)$$

it is not garanteed that $f(x_{k+1}) < f(x_k)$. But if $F(x_k) > 0$ and $\nabla f(x_k) \neq 0$, then d_k is a descent direction. So we choose

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2.23}$$

where α_k satisfies the Armijo condition: $\alpha_k > 0$ is decreased until

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c \cdot \alpha_k \cdot \nabla f(x_k)^T d_k$$

for some given constant $c \sim 0.1$.

The Hessian matrix is usually very expensive to compute. One therefore replaces $F(x_k)$ by an approximation B_k (or replaces $F(x_k)^{-1}$ by an approximation H_k) in a way so that B_k (H_k) can be easily computed at each iteration. For example, working with B_k, then (2.22) becomes

$$p_k(x) = f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T B_k (x - x_k). \tag{2.24}$$

So

$$\nabla p_k(x) = \nabla f(x_k) + B_k(x - x_k). \tag{2.25}$$

The iteration making use of B_{k+1} is

$$p_{k+1}(x) = f(x_{k+1}) + \nabla f(x_{k+1})^T (x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^T B_{k+1}(x - x_{k+1}).$$

What requirements should one impose on B_{k+1}? A reasonable requirement is that the gradient of p_{k+1} should align with the gradient of the objective function f at the two latest iterates, x_k and x_{k+1}. Since $\nabla p_{k+1}(x_{k+1})$ is exactly $\nabla f(x_{k+1})$, this condition is automatically satisfied for the second iterate. The first condition can be expressed mathematically as

$$\nabla p_{k+1}(x_k) = \nabla f(x_{k+1}) + B_{k+1}(x_k - x_{k+1}) = \nabla f(x_k).$$

We obtain

$$B_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k). \tag{2.26}$$

To simplify the notation, we set

$$\Delta x_k = x_{k+1} - x_k, \qquad \Delta g_k = \nabla f(x_{k+1}) - \nabla f(x_k), \tag{2.27}$$

so (2.26) becoms

$$B_{k+1}\Delta x_k = \Delta g_k. \tag{2.28}$$

This formula is known as the secant equation.

Multiply by Δx_k^T, we get

$$\Delta x_k^T B_{k+1}\Delta x_k = \Delta x_k^T \Delta g_k,$$

the secant equation requires the symmetric positive definite matrix $B_{k+1} > 0$. This is only feasible if Δx_k and Δg_k satisfy the curvature condition,

$$\Delta x_k^T \Delta g_k > 0. \tag{2.29}$$

When f is strongly convex, then (2.29) will be satisfied.

The BFGS formula is widely regarded as the most effective of all quasi-Newton updating methods. Rather than imposing conditions on the Hessian approximations B_k, one can apply similar conditions to their inverses, H_k. The updated approximation H_{k+1} must be symmetric, positive definite, and satisfy the secant equation (2.28), which can now be written as:

$$H_{k+1}\Delta g_k = \Delta x_k. \tag{2.30}$$

One chooses H_{k+1} to be solution of

$$\begin{cases} H_{k+1} = \min \|H - H_k\| \\ H^T = H, \quad H_{k+1}\Delta g_k = \Delta x_k. \end{cases} \tag{2.31}$$

The unique solution H_{k+1} to (2.31) is given by

$$\text{(BFGS 1)} \quad H_{k+1} = (I - \rho_k \Delta x_k \Delta g_k^T) H_k (I - \rho_k \Delta g_k \Delta x_k^T) + \rho_k \Delta x_k \Delta x_k^T, \tag{2.32}$$

where

$$\rho_k = \frac{1}{\Delta g_k^T \Delta x_k.} \tag{2.33}$$

To avoid matrix multiplication, rewrite:

$$H_{k+1} = H_k + \rho_k(1 + \rho_k \Delta g_k^T H_k \Delta g_k)\Delta x_k \Delta x_k^T - \rho_k\big(H_k \Delta g_k \Delta x_k^T + (H_k \Delta g_k \Delta x_k^T)^T\big). \tag{2.34}$$

The proposed BFGS Method using the matrices H_k is summarized in Algorithm 1.

Algorithm 1 BFGS Method

Require: Choose starting point x_0, inverse Hessian approximation H_0, and stopping criterion $\epsilon > 0$

while $\|\nabla f(x_k)\| \geq \epsilon$ **do**;

$$d_k = -H_k \nabla f(x_k);$$

Compute α_k to satisfy the Armijo condition;

Set $x_{k+1} = x_k + \alpha_k d_k$;

Set $\Delta x_k = x_{k+1} - x_k = \alpha_k d_k$ and $\Delta g_k = \nabla f(x_{k+1}) - \nabla f(x_k)$;

Compute H_{k+1} by means of (2.32);

Set $k = k + 1$;

end while

Applying the Sherman–Morrison–Woodbury formula to (2.32) one gets

$$B_{k+1} = B_k - \frac{B_k \Delta x_k \Delta x_k^T B_k}{\Delta x_k^T B_k \Delta x_k} + \frac{\Delta g_k \Delta g_k^T}{\Delta x_k^T \Delta g_k}. \tag{2.35}$$

This is the BFGS update of B_k.

Since H_k is the inverse of B_k, to derive the BFGS 1 update for the inverse Hessian approximation, we take the inverse of B_{k+1} to obtain:

$$\begin{aligned} \text{(BFGS 2)} \quad H_{k+1} &= \left(B_{k+1}\right)^{-1} \\ &= \left(B_k - \frac{B_k \Delta x_k \Delta x_k^T B_k}{\Delta x_k^T B_k \Delta x_k} + \frac{\Delta g_k \Delta g_k^T}{\Delta x_k^T \Delta g_k} \right)^{-1}. \end{aligned} \tag{2.36}$$

To calculate H_{k+1} by inverting the right-hand side of this equation, we utilize the Sherman–Morrison–Woodbury formula for matrix inversion.

2.8 Friedman Test

The Friedman test is a non-parametric statistical test commonly used in machine learning to compare the performance of multiple models across multiple datasets. This test evaluates differences across three or more dependent groups by ranking the data

within each group and analyzing these rankings through the Friedman statistic. The steps of the nonparametric Friedman test can be summarized as follows:

1. Collect the performance measures for each model (or method) on the same set of datasets.

2. For each dataset, rank the performance of the models. Assign ranks based on their performance values, the classifier corresponding to the best performing model has the smallest rank. In case of ties, assign the average rank to tied values.

3. Compute the Friedman test statistic χ_F^2 by

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_{j=1}^{k} R_j^2 - \frac{k(k+1^2)}{4}\right], \tag{2.37}$$

   where:

   - k is number of methods or models,
   - N is number of datasets,
   - $r_i(j)$ is the rank of method j on dataset i,
   - $R_j = \frac{1}{N}\sum_{i=1}^{N} r_i(j)$ is the average rank of method j.

4. Based on the Friedman test, the F distribution is subsequently used for further testing. The F distribution is a probability distribution that arises frequently in the analysis of variance (ANOVA) and is used to compare variances or test the overall significance of model differences. The F distribution is

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}, \tag{2.38}$$

   with $k-1$ and $(k-1)(N-1)$ degrees of freedom.

5. If the value of F_F is larger than the corresponding critical values $F\big(k-1,(k-1)(N-1)\big)$ at the confidence level α, reject the null hypothesis. That is, there are significant differences among the k models. If not, fail to reject the null hypothesis, meaning there is no significant difference.

CHAPTER III

RESEARCH METHODOLOGY

This chapter presents the steps used in this research. The procedure consists of 8 steps:

1. Comparative Analysis of Loss Functions;
2. Data Collection;
3. Mathematical Reformulation;
4. Algorithm Development;
5. Computational Tools;
6. Hardware Used;
7. Numerical Experiments;
8. Statistical Validation.

3.1 Comparative Analysis of Loss Functions

Hinge loss, Pinball loss, ϵ- insensitive loss, and generalized pinball loss functions are non-differentiable, unstable for resampling, sensitivity to noise. To address these issues, we modify all these loss functions into a C^2-smooth generalized pinball loss function (abbreviated SGPL). We propose a twice continuously differentiable approximate loss function of the generalized pinball loss function that resolves the problems of non-differentiability, resampling instability, and noise sensitivity.

3.2 Data Collection

In this research, we selected nine datasets from https://archive.ics.uci.edu/. The UCI Machine Learning Repository offers a wide variety of datasets in terms of size, number of features, and data types such as binary-class, multi-class, or regression problems. This diversity allows for comprehensive testing of the model's performance across various scenarios. UCI datasets are widely recognized in Machine Learning and are considered benchmark datasets. Using these datasets enables direct comparison of results with other research studies.

3.3 Mathematical Reformulation

The reformulation of SVM into TBSVM adapts the classification framework to use two non-parallel hyperplanes instead of a single hyperplane that individually optimize for their respective classes, enhancing flexibility and performance for complex and imbalanced datasets. To enhance the performance of the SVM, a novel TBSVM with smooth generalized pinball loss function (BFGS-TBSVM) is introduced. We demonstrate that the generalized pinball loss function can be arbitrarity approximated using our smooth approximation function and that the solutions of our model converge to those of the exact model.

3.4 Algorithm Development

The Newton algorithm is an efficient optimization technique that has quadratic rate of convergence. However, it requires the objective function to be twice continuously differentiable.

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is an optimization technique widely derived from Newton's algorithm and used in machine learning for solving unconstrained optimization problems, particularly when the objective function is smooth and differentiable. It belongs to the family of quasi-Newton methods, which approximate the Hessian matrix (second-order derivatives) of the objective function to guide the op-

timization process. Consequently, the proposed algorithm is effectively solved using the quasi-Newton (BFGS) method.

3.5 Computational Tools

The computer code was implemented in Python3 using the numpy and sklearn packages under the Linux operating system

3.6 Hardware Used

All experiments were performed on a Desktop computer with Ryzen 5700G CPU and 32GB of memory.

3.7 Numerical Experiments

In machine learning, Accuracy (Acc), Standard Deviation (sd), and Training Time (time(s)) are used as key evaluation metrics to comprehensively assess the performance of a model. Therefore, we compared the accuracy, sd, and time(s) of the TBSVM models.

3.8 Statistical Validation

In the statistical analysis, we use the Friedman test to evaluate the classification performance. The Friedman Test is a non-parametric statistical test used to compare the median values of multiple related groups. It is particularly useful in comparative analyses, such as evaluating different models or parameter tuning processes in machine learning.

CHAPTER IV

RESULTS AND DISCUSSION

This chapter presents the proof of the convergence of the proposed loss function to the generalized pinball loss function with decreasing parameter μ, and evaluates the performance of TBSVM under different loss functions.

4.1 Support Vector Machine

One of the quasi-Newton algorithms, BFGS, is used to solve strongly convex differentiable problems. First, we construct a smooth approximate loss function of the generalized pinball loss function, followed by proving, as a theorem, that this function can serve as an estimator for the generalized pinball loss function.

As stated in chapter II, section 6, the usual loss functions are non-differentiable, which makes numerical solutions difficult. We introduce a new smooth loss function that approximates the generalized pinball loss function into the SVM and TBSVM. The substitution of $L^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(u)$ is carried out by employing a smooth approximation, represented as $P^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2} : \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R}^+$, and defined as

$$P^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(u,\mu) = \begin{cases} \tau_1(u - \frac{\epsilon_1}{\tau_1}) - \frac{5}{8}\tau_1^2\mu, & \frac{\epsilon_1}{\tau_1} + \tau_1\mu < u, \\ \frac{5}{8\tau_1^2\mu^3}(u - \frac{\epsilon_1}{\tau_1})^4 - \frac{1}{4\tau_1^4\mu^5}(u - \frac{\epsilon_1}{\tau_1})^6, & \frac{\epsilon_1}{\tau_1} < u \leq \frac{\epsilon_1}{\tau_1} + \tau_1\mu, \\ 0, & -\frac{\epsilon_2}{\tau_2} \leq u \leq \frac{\epsilon_1}{\tau_1}, \\ \frac{5}{8\tau_2^2\mu^3}(u + \frac{\epsilon_2}{\tau_2})^4 - \frac{1}{4\tau_2^4\mu^5}(u + \frac{\epsilon_2}{\tau_2})^6, & -\frac{\epsilon_2}{\tau_2} - \tau_2\mu \leq u < -\frac{\epsilon_2}{\tau_2}, \\ -\tau_2(u + \frac{\epsilon_2}{\tau_2}) - \frac{5}{8}\tau_2^2\mu, & u < -\frac{\epsilon_2}{\tau_2} - \tau_2\mu. \end{cases} \tag{4.1}$$

where the parameters $\tau_1, \tau_2, \epsilon_1, \epsilon_2$ are non-negative real values, $u \in \mathbb{R}$ is the variable, and $\mu \in \mathbb{R}^+$ the approximation parameter. Note that when $\mu = 0$, we recover the generalized pinball loss $L^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(u)$. Figure 4.1 displays the graphic representation of $P^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(u,\mu)$ for various values of the parameter μ, with $\tau_1, \tau_2, \epsilon_1, \epsilon_2$ fixed ($\tau_1, \tau_2, \epsilon_1, \epsilon_2 = 1$).

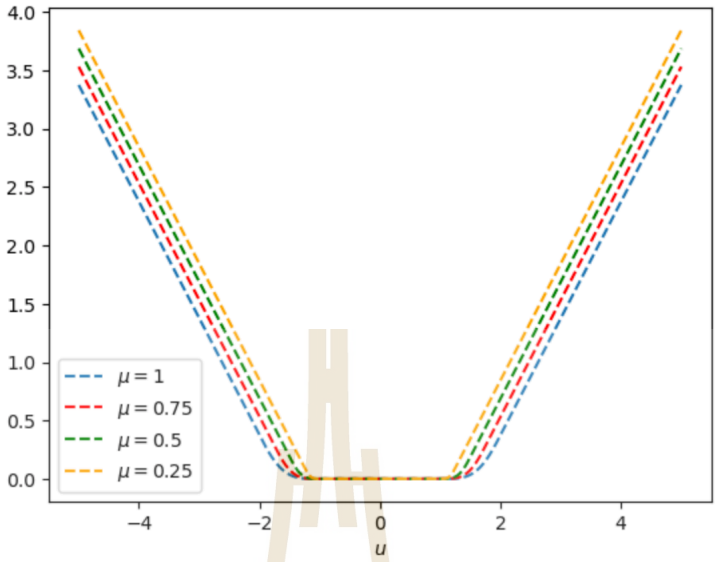


Figure 4.1 $P^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(u,\mu)$ for various values of μ.

It is easy to verify that this is a C^2-function and belongs to a category of smoothing functions for $L^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(u)$. Figure 4.1 hints that the mapping $P^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(u,\mu)$ converges to $L^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(u)$ as $\mu \to 0^+$, which we will prove.

In the proof, we will use the symbols $L(u)$ and $P(u,\mu)$ in place of $L^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(u)$ and $P^{\epsilon_1,\epsilon_2}_{\tau_1,\tau_2}(u,\mu)$, respectively, since the parameters $\tau_1, \tau_2, \epsilon_1, \epsilon_2$ remain fixed.

Lemma 4.1. *Let $L(u)$ and $P(u,\mu)$ be defined as in (2.19) and (4.1), respectively. Then*

(i) for all $u \in \mathbb{R}$, $0 \le L(u) - P(u,\mu) \le \frac{5}{8}\tau_0^2\mu$, where $\tau_0 = \max\{\tau_1,\tau_2\}$,

(ii) $\lim_{\mu\to 0^+} P(u,\mu) = L(u)$ uniformly on $\mathbb{R}$.

Proof. (i) Let $\epsilon_1, \epsilon_2 \ge 0, \tau_1, \tau_2 > 0$ be fixed and $\mu > 0$. In list of the definitions of $L(u)$ and $P(u,\mu)$, we divide the proof into five cases, depending on the valus of u.

Case 1: $\frac{\epsilon_1}{\tau_1} + \tau_1\mu < u$. We get

$$L(u) - P(u,\mu) = \tau_1(u - \frac{\epsilon_1}{\tau_1}) - \left[\tau_1(u - \frac{\epsilon_1}{\tau_1}) - \frac{5}{8}\tau_1^2\mu\right] = \frac{5}{8}\tau_1^2\mu \le \frac{5}{8}\tau_0^2\mu.$$

Case 2: $\frac{\epsilon_1}{\tau_1} < u \le \frac{\epsilon_1}{\tau_1} + \tau_1\mu$. We have

$$L(u) - P(u,\mu) = \tau_1(u - \frac{\epsilon_1}{\tau_1}) - \left[\frac{5}{8\tau_1^2\mu^3}(u - \frac{\epsilon_1}{\tau_1})^4 - \frac{1}{4\tau_1^4\mu^5}(u - \frac{\epsilon_1}{\tau_1})^6\right].$$

Set $I(u) = L(u) - P(u,\mu)$, we obtain that

$$I'(u) = \tau_1 - \left[\frac{5}{2\tau_1^2\mu^3}(u - \frac{\epsilon_1}{\tau_1})^3 - \frac{3}{2\tau_1^4\mu^5}(u - \frac{\epsilon_1}{\tau_1})^5\right].$$

Let $x = u - \frac{\epsilon_1}{\tau_1}$. Then, x lies in the interval $(0, \tau_1\mu]$ and a $I(x) = \tau_1 - \left[\frac{5}{8\tau_1^2\mu^3}x^4 - \frac{1}{4\tau_1^4\mu^5}x^6\right]$ on the interval $\left(0, \tau_1\mu\right]$. So, $I'(x) = \tau_1 - \left[\frac{5}{2\tau_1^2\mu^3}x^3 - \frac{3}{2\tau_1^4\mu^5}x^5\right]$. The critical points of the function $I'(x)$ are $x = 0, \tau_1\mu$ and $-\tau_1\mu$, so that $I'(x)$ is monotone on $[0, \tau_1\mu]$. As $I'(0) = \tau_1 > 0$ and $I'(\tau_1\mu) = 0$, therefore, $I'(x) \geq 0$ on $\left(0, \tau_1\mu\right]$, and hence $I(u)$ is an increasing function on $\frac{\epsilon_1}{\tau_1} < u \leq \frac{\epsilon_1}{\tau_1} + \tau_1\mu$. Then

$$L(u) - P(u,\mu) \leq I(u) \mid_{u = \frac{\epsilon_1}{\tau_1} + \tau_1\mu} = \frac{5}{8}\tau_1^2\mu \leq \frac{5}{8}\tau_0^2\mu.$$

Case 3: $\frac{\epsilon_2}{\tau_2} \leq u \leq \frac{\epsilon_1}{\tau_1}$. We get

$$L(u) - P(u,\mu) = 0.$$

Case 4: $-\frac{\epsilon_2}{\tau_2} - \tau_2\mu \leq u < \frac{\epsilon_2}{\tau_2}$. Proceeding similar to case 2, we obtain that $I(u)$ is a decreasing function on $\left(-\frac{\epsilon_2}{\tau_2} - \tau_2\mu, \frac{\epsilon_2}{\tau_2}\right]$. Then

$$L(u) - P(u,\mu) \leq I(-\frac{\epsilon_2}{\tau_2} - \tau_2\mu) = \frac{5}{8}\tau_2^2\mu \leq \frac{5}{8}\tau_0^2\mu.$$

Case 5: $u < -\frac{\epsilon_2}{\tau_2} - \tau_2\mu$. Proceeding similar to case 1, we obtain

$$L(u) - P(u,\mu) = -\tau_2(u + \frac{\epsilon_2}{\tau_2}) - \left[-\tau_2(u + \frac{\epsilon_1}{\tau_1}) - \frac{5}{8}\tau_2^2\mu\right] = \frac{5}{8}\tau_2^2\mu \leq \frac{5}{8}\tau_0^2\mu.$$

All 5 cases show that

$$0 \leq \sup_{u\in\mathbb{R}}\left[L(u) - P(u,\mu)\right] \leq \frac{5}{8}\tau_0^2\mu. \tag{4.2}$$

(ii) By the Squeeze Theorem, if follows that

$$\lim_{\mu\to 0^+} P(u,\mu) = L(u),$$

uniformly on $\mathbb{R}$, as $\mu \to 0^+$. □

Consider a dataset $X = \{(\overrightarrow{x_i}, y_i) | i = 1, 2, ..., m\}, \overrightarrow{x_i} \in \mathbb{R}^n, y_i \in \{+1, -1\}$. We replace the usage of the generalized pinball loss function $L(u)$ with our smooth approximation $P(u, \mu)$, where μ is the parameter. The SVM model (2.14) changes to the optimization problem

$$\min_{\vec{\boldsymbol{\omega}}} \varphi_\mu(\vec{\boldsymbol{\omega}}) := \frac{1}{2}\|\vec{\boldsymbol{\omega}}\|^2 + \frac{C}{m}\sum_{i=1}^{m} P(1 - y_i \vec{\boldsymbol{\omega}}^T \vec{\boldsymbol{x_i}}, \mu), \tag{4.3}$$

where $\vec{\boldsymbol{\omega}} = [\overrightarrow{w}^T, b]^T \in \mathbb{R}^{n+1}, \vec{\boldsymbol{x_i}} = [\overrightarrow{x_i}, 1]^T$. To simplify notation, we will denote $\vec{\boldsymbol{\omega}}$ simply by $\overrightarrow{\omega}$ and $\vec{\boldsymbol{x_i}}$ by $\overrightarrow{x_i}$ from here on. In the following, we will demonstrate that each optimization problem in the family defined by (4.3) admits a unique solution, and that the sequence of solutions for this family converges to the solution of the exact problem:

$$\min_{\overrightarrow{\omega}} \psi_0(\overrightarrow{\omega}) := \frac{1}{2}\|\overrightarrow{\omega}\|^2 + \frac{C}{m}\sum_{i=1}^{m} L(1 - y_i \overrightarrow{\omega}^T \overrightarrow{x_i}), \tag{4.4}$$

as $\mu \to 0^+$.

The next theorem shows that the objective function of (4.3) converges to the objective function of (4.4) uniformly as $\mu \to 0^+$.

Theorem 4.2. *Let $\tau_0 = max\{\tau_1, \tau_2\}$, then for all $\overrightarrow{\omega} \in \mathbb{R}^{n+1}$ and $\mu > 0$,* $\|\psi_0(\overrightarrow{\omega}) - \varphi_\mu(\overrightarrow{\omega})\|_\infty \leq C\left(\frac{5}{8}\tau_0^2\mu\right)$.

Proof. For all $\overrightarrow{\omega} \in \mathbb{R}^{n+1}$ as $L(u) - P(u, \mu) \geq 0$,

$$\begin{aligned}
0 \leq \psi_0(\overrightarrow{\omega}) - \varphi_\mu(\overrightarrow{\omega}) &= \frac{C}{m}\sum_{i=1}^{m} L(1 - y_i \overrightarrow{\omega}^T \overrightarrow{x_i}) - \frac{C}{m}\sum_{i=1}^{m} P(1 - y_i \overrightarrow{\omega}^T \overrightarrow{x_i}, \mu) \\
&= \frac{C}{m}\sum_{i=1}^{m}\Big[L(u_i) - P(u_i, \mu)\Big] \quad [u_i = 1 - y_i \overrightarrow{\omega}^T \overrightarrow{x_i}] \\
&\leq \frac{C}{m}\sum_{i=1}^{m}\frac{5}{8}\tau_0^2\mu \quad \text{[By (4.2) in Lemma 4.1]} \\
&= C\left(\frac{5}{8}\tau_0^2\mu\right).
\end{aligned}$$

It follows that $\|\psi_0(\overrightarrow{\omega}) - \varphi_\mu(\overrightarrow{\omega})\|_\infty \leq C\left(\frac{5}{8}\tau_0^2\mu\right)$. □

Finally, we show that the solution of (4.3) converges to the solution of (4.4) as $\mu \to 0^+$.

Theorem 4.3. *Let $\varphi_\mu(\overrightarrow{\omega})$ and $\psi_0(\overrightarrow{\omega})$ be defined as in (4.3) and (4.4), respectively, and let $\overrightarrow{\omega}^*$ be an optimal solution of problem (4.4). Then*

(i) there exists a unique solution $\overrightarrow{\omega}^*_\mu$ of problem (4.3),

(ii) $\|\overrightarrow{\omega}^*_\mu - \overrightarrow{\omega}^*\|_2^2 \leq C\left(\frac{5}{8}\tau_0^2\mu\right)$,

(iii) $\overrightarrow{\omega}^*_\mu \to \overrightarrow{\omega}^*$ *as* $\mu \to 0^+$.

Proof. Let $\epsilon_1, \epsilon_2 \geq 0, \tau_1, \tau_2 > 0$ be fixed.

(i) Let $\mu > 0$ be arbitrary, but given. For $c \in \mathbb{R}$, the set

$$S_c(\varphi_\mu) := \{\overrightarrow{\omega} \in \mathbb{R}^{n+1} : \varphi_\mu(\overrightarrow{\omega}) \leq c\}$$

is called a sublevel set. Clearly, we can pick $c > 0$ so that $S_c(\varphi_\mu) \neq \emptyset$. Since φ_μ is continuous, then the sublevel set $S_c(\varphi_\mu)$ is closed. By definition of φ_μ,

$$S_c(\varphi_\mu) \subseteq \{\overrightarrow{\omega} \in \mathbb{R}^{n+1} : \overrightarrow{\omega}^T\overrightarrow{\omega} = ||\overrightarrow{\omega}||^2 \leq 2c\}$$

Since the set $\{\overrightarrow{\omega} \in \mathbb{R}^{n+1} : \overrightarrow{\omega}^T\overrightarrow{\omega} = ||\overrightarrow{\omega}||^2 \leq 2c\}$ is bounded, it follows that $S_c(\varphi_\mu)$ is bounded. By $S_c(\varphi_\mu)$ closed and bounded, we obtain that $S_c(\varphi_\mu)$ is a compact set in $\mathbb{R}^{n+1}$. By the Extreme Value Theorem, it follows that φ_μ has a minimizer $\overrightarrow{\omega}^*_\mu$ on $S_c(\varphi_\mu)$. i.e. Problem (4.3) has a solution. On the other hand, $P(u, \mu)$ is convex and $u_i = 1 - y_i\overrightarrow{\omega}^T\overrightarrow{x_i}$ is affine in $\overrightarrow{\omega}$, we get that $C\sum_{i=1}^{m} P(1 - y_i\overrightarrow{\omega}^T\overrightarrow{x_i}, \mu)$ is convex. As $\frac{1}{2}||\overrightarrow{\omega}||^2$ is strongly convex, it follows that $\varphi_\mu(\overrightarrow{\omega})$ is also strongly convex. By the strong convexity, we get uniqueness of the solution of problem (4.3).

(ii) Let that $\overrightarrow{\omega}^*_\mu$ and $\overrightarrow{\omega}^*$ be optimal solutions of (4.3) and (4.4), respectively. Let $\nabla\varphi_\mu(\overrightarrow{\omega})$ be the gradient of $\varphi_\mu(\overrightarrow{\omega})$ and $\nabla_0(\overrightarrow{\omega})$ be subgradients of $\psi_0(\overrightarrow{\omega})$, that is

$$\nabla_0(\overrightarrow{\omega}) \in \partial\psi_0(\overrightarrow{\omega}) = \overrightarrow{\omega} - C\sum_{i=1}^{m} y_i\overrightarrow{x_i}\partial L(u_i), \quad [u_i = 1 - y_i\overrightarrow{\omega}^T\overrightarrow{x_i}]$$

where

$$\partial L(u) = \begin{cases} \{\tau_1\}, & \frac{\epsilon_1}{\tau_1} < u, \\ [0, \tau_1], & u = \frac{\epsilon_1}{\tau_1}, \\ \{0\}, & -\frac{\epsilon_2}{\tau_2} \leq u \leq \frac{\epsilon_1}{\tau_1}, \\ [-\tau_2, 0], & u = -\frac{\epsilon_2}{\tau_2}, \\ \{-\tau_2\}, & u < -\frac{\epsilon_2}{\tau_2}. \end{cases}$$

By the strong convexity with the parameter 1, we obtain that

$$\psi_0(\overrightarrow{\omega}^*_\mu) - \psi_0(\overrightarrow{\omega}^*) \geq (\overrightarrow{\omega}^*_\mu - \overrightarrow{\omega}^*)\nabla_0(\overrightarrow{\omega}^*) + \frac{1}{2}||\overrightarrow{\omega}^*_\mu - \overrightarrow{\omega}^*||_2^2,$$

and

$$\varphi_\mu(\overrightarrow{\omega}^*) - \varphi_\mu(\overrightarrow{\omega}^*_\mu) \geq (\overrightarrow{\omega}^* - \overrightarrow{\omega}^*_\mu)\nabla\varphi_\mu(\overrightarrow{\omega}^*_\mu) + \frac{1}{2}||\overrightarrow{\omega}^*_\mu - \overrightarrow{\omega}^*||_2^2.$$

The first-order necessary condition for optimality states that the gradient and sub-gradient of a strongly convex function at local minimum points must be zero. Thus, we get

$$\psi_0(\overrightarrow{\omega}^*_\mu) - \psi_0(\overrightarrow{\omega}^*) \geq \frac{1}{2}||\overrightarrow{\omega}^*_\mu - \overrightarrow{\omega}^*||_2^2,$$

and

$$\varphi_\mu(\overrightarrow{\omega}^*) - \varphi_\mu(\overrightarrow{\omega}^*_\mu) \geq \frac{1}{2}||\overrightarrow{\omega}^*_\mu - \overrightarrow{\omega}^*||_2^2.$$

By Lemma 4.1. we have

$$\psi_0(\overrightarrow{\omega}) - \varphi_\mu(\overrightarrow{\omega}) = \frac{C}{m}\sum_{i=1}^{m} L(u_i) - \frac{C}{m}\sum_{i=1}^{m} P(u_i, \mu) \geq 0, \quad \forall\overrightarrow{\omega}$$

Consider

$$\begin{aligned} ||\overrightarrow{\omega}^*_\mu - \overrightarrow{\omega}^*||_2^2 \leq & \ \psi_0(\overrightarrow{\omega}^*_\mu) - \psi_0(\overrightarrow{\omega}^*) + \varphi_\mu(\overrightarrow{\omega}^*) - \varphi_\mu(\overrightarrow{\omega}^*_\mu) \\ = & \ \psi_0(\overrightarrow{\omega}^*_\mu) - \varphi_\mu(\overrightarrow{\omega}^*_\mu) - \left[\psi_0(\overrightarrow{\omega}^*) - \varphi_\mu(\overrightarrow{\omega}^*)\right] \\ \leq & \ \psi_0(\overrightarrow{\omega}^*_\mu) - \varphi_\mu(\overrightarrow{\omega}^*_\mu). \end{aligned}$$

and by Theorem 4.2. we get

$$\begin{aligned} 0 \leq ||\overrightarrow{\omega}^*_\mu - \overrightarrow{\omega}^*||_2^2 \leq \psi_0(\overrightarrow{\omega}^*_\mu) - \varphi_\mu(\overrightarrow{\omega}^*_\mu) = & \ |\psi_0(\overrightarrow{\omega}^*_\mu) - \varphi_\mu(\overrightarrow{\omega}^*_\mu)| \\ \leq & \ \|\psi_0 - \varphi_\mu\|_\infty \\ \leq & \ C\left(\frac{5}{8}\tau_0^2\mu\right). \end{aligned}$$

(iii) By the above and the Squeeze Theorem, we obtain that $\overrightarrow{\omega}^*_\mu \to \overrightarrow{\omega}^*$ as $\mu \to 0^+$. □

4.2 Twin Bounded Support Vector Machine

The combination of TBSVM with the smooth generalized pinball loss function $P(u, \mu)$ is designed to improve the algorithm's performance in handling complex or noisy data, which traditional loss functions (such as the hinge loss) often struggle with. Consequently, a novel TBSVM with smooth generalized pinball loss is introduced to enhance the algorithm's ability to manage noise sensibility and imbalanced data. This approach not only improves the model's predictive performance but also ensures faster and more stable computations during the optimization process.

4.2.1 Linear case

We reformulate (2.10) and (2.11) as unconstrained optimization problems, replacing slack variable ξ_i and η_i with the generalized pinball loss function $L(u)$. The TBSVM model has a following structure:

$$\min_{\vec{\boldsymbol{\omega_1}}} \Phi_0^{(1)}(\vec{\boldsymbol{\omega_1}}) := \frac{1}{2}\sum_{i=1}^{m_1}(\vec{\boldsymbol{\omega_1}}^T\vec{\boldsymbol{x_i}}^{(1)})^2 + c_3\|\vec{\boldsymbol{\omega_1}}\|^2 + \frac{c_1}{m_2}\sum_{i=1}^{m_2} L(\vec{\boldsymbol{\omega_1}}^T\vec{\boldsymbol{x_i}}^{(2)} + 1) \tag{4.5}$$

$$\min_{\vec{\boldsymbol{\omega_2}}} \Phi_0^{(2)}(\vec{\boldsymbol{\omega_2}}) := \frac{1}{2}\sum_{i=1}^{m_2}(\vec{\boldsymbol{\omega_2}}^T\vec{\boldsymbol{x_i}}^{(2)})^2 + c_4\|\vec{\boldsymbol{\omega_2}}\|^2 + \frac{c_2}{m_1}\sum_{i=1}^{m_1} L(1 - \vec{\boldsymbol{\omega_2}}^T\vec{\boldsymbol{x_i}}^{(1)}) \tag{4.6}$$

where $\vec{\boldsymbol{\omega_1}} = [\overrightarrow{w_1}^T, b_1]^T, \vec{\boldsymbol{\omega_2}} = [\overrightarrow{w_2}^T, b_2]^T$ and $\vec{\boldsymbol{x_i}}^{(1)} = [\overrightarrow{x_i}^{(1)}, 1]^T, \vec{\boldsymbol{x_i}}^{(2)} = [\overrightarrow{x_i}^{(2)}, 1]^T$.

Again, for simplicity we drop the "bold" symbols. We replace the last terms with our novel loss function $P(\cdot, \mu)$, and obtain the following structure:

$$\min_{\overrightarrow{\omega_1}} \phi_\mu^{(1)}(\overrightarrow{\omega_1}) := \frac{1}{2}\sum_{i=1}^{m_1}(\overrightarrow{\omega_1}^T\overrightarrow{x_i}^{(1)})^2 + c_3\|\overrightarrow{\omega_1}\|^2 + \frac{c_1}{m_2}\sum_{i=1}^{m_2} P(\overrightarrow{\omega_1}^T\overrightarrow{x_i}^{(2)} + 1, \mu) \tag{4.7}$$

$$\min_{\overrightarrow{\omega_2}} \phi_\mu^{(2)}(\overrightarrow{\omega_2}) := \frac{1}{2}\sum_{i=1}^{m_2}(\overrightarrow{\omega_2}^T\overrightarrow{x_i}^{(2)})^2 + c_4\|\overrightarrow{\omega_2}\|^2 + \frac{c_2}{m_1}\sum_{i=1}^{m_1} P(1 - \overrightarrow{\omega_2}^T\overrightarrow{x_i}^{(1)}, \mu) \tag{4.8}$$

Theorem 4.4. *Let $\tau_0 = max\{\tau_1, \tau_2\}$. Then $\|\Phi_0^{(1)}(\overrightarrow{\omega_1}) - \phi_\mu^{(1)}(\overrightarrow{\omega_1})\|_\infty \leq c_1\left(\frac{5}{8}\tau_0^2\mu\right)$ and $\|\Phi_0^{(2)}(\overrightarrow{\omega_2}) - \phi_\mu^{(2)}(\overrightarrow{\omega_2})\|_\infty \leq c_2\left(\frac{5}{8}\tau_0^2\mu\right)$.*

Proof. Recall from the proof of Lemma 4.1 that $L(u) - P(u,\mu) \leq \frac{5}{8}\tau_0^2\mu, \forall u \in \mathbb{R}$. Then

$$\begin{aligned} 0 \leq \Phi_0^{(1)}(\overrightarrow{\omega_1}) - \phi_\mu^{(1)}(\overrightarrow{\omega_1}) &= \frac{c_1}{m_2}\sum_{i=1}^{m_2} L(\overrightarrow{\omega_1}^T\overrightarrow{x_i}^{(2)} + 1) - \frac{c_1}{m_2}\sum_{i=1}^{m_2} P(\overrightarrow{\omega_1}^T\overrightarrow{x_i}^{(2)} + 1, \mu) \\ &= \frac{c_1}{m_2}\sum_{i=1}^{m_2}\Big[L(u_i) - P(u_i,\mu)\Big] \quad [u_i = \overrightarrow{\omega_1}^T\overrightarrow{x_i}^{(2)} + 1] \\ &\leq \frac{c_1}{m_2}\sum_{i=1}^{m_2}\frac{5}{8}\tau_0^2\mu \\ &= c_1 \cdot \frac{5}{8}\tau_0^2\mu \end{aligned}$$

i.e. $\|\Phi_0^{(1)}(\overrightarrow{\omega_1}) - \phi_\mu^{(1)}(\overrightarrow{\omega_1})\|_\infty \leq c_1\left(\frac{5}{8}\tau_0^2\mu\right)$.

In a similar way, one shows that $\|\Phi_0^{(2)}(\overrightarrow{\omega_2}) - \phi_\mu^{(2)}(\overrightarrow{\omega_2})\|_\infty \leq c_2\left(\frac{5}{8}\tau_0^2\mu\right)$. □

Theorem 4.5. *Let $\overrightarrow{\omega}_1^*$ and $\overrightarrow{\omega}_2^*$ be the optimal solutions of problems (4.5) and (4.6), respectively. Then*

(i) there exist unique solutions of problems (4.7) and (4.8), denoted $\overrightarrow{\omega}_1^\mu$ and $\overrightarrow{\omega}_2^\mu$, respectively.

(ii) $\|\overrightarrow{\omega}_1^\mu - \overrightarrow{\omega}_1^*\|_2^2 \leq c_1\left(\frac{5}{8}\tau_0^2\mu\right)$ and $\|\overrightarrow{\omega}_2^\mu - \overrightarrow{\omega}_2^*\|_2^2 \leq c_2\left(\frac{5}{8}\tau_0^2\mu\right)$,

(iii) $\overrightarrow{\omega}_1^\mu \to \overrightarrow{\omega}_1^*$ and $\overrightarrow{\omega}_2^\mu \to \overrightarrow{\omega}_2^*$ as $\mu \to 0^+$.

Proof. (i) Let $\mu > 0$ be arbitrary, but fixed. Pick $\nu > 0$ so that the sublevel set

$$S_\nu(\phi_\mu^{(1)}) := \{\overrightarrow{\omega_1} \in \mathbb{R}^{n+1} : \phi_\mu^{(1)}(\overrightarrow{\omega_1}) \leq \nu\}$$

is not empty. Since $\phi_\mu^{(1)}(\overrightarrow{\omega_1})$ is continuous, then, the sublevel set $S_\nu(\phi_\mu^{(1)})$ is closed. Furthermore, by definition of $\phi_\mu^{(1)}$,

$$S_\nu(\phi_\mu^{(1)}) \subseteq \{\overrightarrow{\omega_1} \in \mathbb{R}^{n+1} : \overrightarrow{\omega_1}^T\overrightarrow{\omega_1} = ||\overrightarrow{\omega_1}||^2 \leq \frac{\nu}{c_3}\}.$$

Since the set $\{\overrightarrow{\omega_1} \in \mathbb{R}^{n+1} : \overrightarrow{\omega_1}^T\overrightarrow{\omega_1} = ||\overrightarrow{\omega_1}||^2 \leq \frac{\nu}{c_3}\}$ is bounded, then $S_\nu(\phi_\mu^{(1)})$ is bounded. By $S_\nu(\phi_\mu^{(1)})$ closed and bounded, we obtain that $S_\nu(\phi_\mu^{(1)})$ is a compact set in $\mathbb{R}^{n+1}$. By the Extreme Value Theorem, a solution of problem (4.7) exists, call the minimizer $\overrightarrow{\omega}_1^\mu$. As before, since $P(u,\mu)$ is convex, then $\frac{c_1}{m_2}\sum_{i=1}^{m_2} P(u_i,\mu)$ is convex.

Next we show that the first term in (4.7) is convex in $\overrightarrow{\omega_1}$. In fact, as the function $y = x^2$ is convex, then

$$\begin{aligned}\left[\left(t\overrightarrow{v_1} + (1-t)\overrightarrow{v_2}\right)^T \overrightarrow{x_i}^{(1)}\right]^2 &= \left[t\overrightarrow{v_1}^T \overrightarrow{x_i}^{(1)} + (1-t)\overrightarrow{v_2}^T \overrightarrow{x_i}^{(1)}\right]^2 \\ &\leq t(\overrightarrow{v_1}^T \overrightarrow{x_i}^{(1)})^2 + (1-t)(\overrightarrow{v_2}^T \overrightarrow{x_i}^{(1)})^2\end{aligned}$$

$\forall \overrightarrow{v_1}, \overrightarrow{v_2} \in \mathbb{R}^{n+1}, \forall t \in [0,1]$. This shows that $\overrightarrow{\omega_1} \mapsto (\overrightarrow{\omega_1}^T \overrightarrow{x_i}^{(1)})^2$ is convex $\forall i$, so that $\frac{1}{2}\sum_{i=1}^{m_1}(\overrightarrow{\omega_1}^T \overrightarrow{x_i}^{(1)})^2$ is convex. As $c_3||\overrightarrow{\omega_1}||^2$ is strongly convex and the remaining terms in (4.7) are convex, it follows that $\phi_\mu^{(1)}(\overrightarrow{\omega_1})$ is also strongly convex. By the strong convexity, we obtain that the solution of problem (4.7) is unique. For uniqueness of the solution of problem (4.8), we proceed in a similar way, to obtain a unique solution $\overrightarrow{\omega}_2^\mu$.

(ii) Let $\overrightarrow{\omega}_1^*$ and $\overrightarrow{\omega}_1^\mu$ be the optimal solutions of (4.5) and (4.7), respectively. Let $\nabla\phi_\mu^{(1)}(\overrightarrow{\omega}_1)$ be gradient of $\phi_\mu^{(1)}(\overrightarrow{\omega}_1)$ and $\nabla_0^{(1)}$ be subgradients of $\Phi_0^{(1)}(\overrightarrow{\omega}_1)$, that is

$$\nabla_0^{(1)}(\overrightarrow{\omega}_1) \in \partial\Phi_0^{(1)}(\overrightarrow{\omega}_1) = \sum_{i=1}^{m_1} \overrightarrow{x_i}^{(1)}\left[\overrightarrow{x_i}^{(1)}\right]^T \overrightarrow{\omega_1} + 2c_3\overrightarrow{\omega_1} + \frac{c_1}{m_2}\sum_{i=1}^{m_2} \overrightarrow{x_i}^{(2)} \partial L(\overrightarrow{\omega_1}^T \overrightarrow{x_i}^{(2)} + 1).$$

By the strong convexity with the parameter 1, we obtain that

$$\Phi_0^{(1)}(\overrightarrow{\omega}_1^\mu) - \Phi_0^{(1)}(\overrightarrow{\omega}_1^*) \geq (\overrightarrow{\omega}_1^\mu - \overrightarrow{\omega}_1^*)\nabla_0^{(1)}(\overrightarrow{\omega}_1^*) + \frac{1}{2}||\overrightarrow{\omega}_1^\mu - \overrightarrow{\omega}_1^*||_2^2,$$

and

$$\phi_\mu^{(1)}(\overrightarrow{\omega}_1^*) - \phi_\mu^{(1)}(\overrightarrow{\omega}_1^\mu) \geq (\overrightarrow{\omega}_1^* - \overrightarrow{\omega}_1^\mu)\nabla\phi_\mu^{(1)}(\overrightarrow{\omega}_1^\mu) + \frac{1}{2}||\overrightarrow{\omega}_1^* - \overrightarrow{\omega}_1^\mu||_2^2.$$

The first-order necessary condition for optimality states that the gradient and sub-gradient of a strongly convex function at local minimum points must be zero. Thus, we get

$$\Phi_0^{(1)}(\overrightarrow{\omega}_1^\mu) - \Phi_0^{(1)}(\overrightarrow{\omega}_1^*) \geq \frac{1}{2}||\overrightarrow{\omega}_1^\mu - \overrightarrow{\omega}_1^*||_2^2,$$

and

$$\phi_\mu^{(1)}(\overrightarrow{\omega}_1^*) - \phi_\mu^{(1)}(\overrightarrow{\omega}_1^\mu) \geq \frac{1}{2}||\overrightarrow{\omega}_1^* - \overrightarrow{\omega}_1^\mu||_2^2 = \frac{1}{2}||\overrightarrow{\omega}_1^\mu - \overrightarrow{\omega}_1^*||_2^2.$$

By Lemma 4.1. we have

$$\Phi_0^{(1)}(\overrightarrow{\omega_1}) - \phi_\mu^{(1)}(\overrightarrow{\omega_1}) = \frac{c_1}{m_2}\sum_{i=1}^{m_2} L(u_i^{(1)}) - \frac{c_1}{m_2}\sum_{i=1}^{m_2} P(u_i^{(1)}, \mu) \geq 0,$$

where $u_i^{(1)} = \overrightarrow{\omega_1}^T \overrightarrow{x_i}^{(2)} + 1$.

Consider

$$\begin{aligned} ||\overrightarrow{\omega}_1^\mu - \overrightarrow{\omega}_1^*||_2^2 \leq & \ \Phi_0^{(1)}(\overrightarrow{\omega}_1^\mu) - \Phi_0^{(1)}(\overrightarrow{\omega}_1^*) + \phi_\mu^{(1)}(\overrightarrow{\omega}_1^*) - \phi_\mu^{(1)}(\overrightarrow{\omega}_1^\mu) \\ = & \ \Phi_0^{(1)}(\overrightarrow{\omega}_1^\mu) - \phi_\mu^{(1)}(\overrightarrow{\omega}_1^\mu) - \left[\Phi_0^{(1)}(\overrightarrow{\omega}_1^*) - \phi_\mu^{(1)}(\overrightarrow{\omega}_1^*)\right] \\ \leq & \ \Phi_0^{(1)}(\overrightarrow{\omega}_1^\mu) - \phi_\mu^{(1)}(\overrightarrow{\omega}_1^\mu). \end{aligned}$$

and by Theorem 4.4. we get

$$\begin{aligned} 0 \leq ||\overrightarrow{\omega}_1^\mu - \overrightarrow{\omega}_1^*||_2^2 \leq \Phi_0^{(1)}(\overrightarrow{\omega}_1^\mu) - \phi_\mu^{(1)}(\overrightarrow{\omega}_1^\mu) = & \ |\Phi_0^{(1)}(\overrightarrow{\omega}_1^\mu) - \phi_\mu^{(1)}(\overrightarrow{\omega}_1^\mu)| \\ \leq & \ \|\Phi_0^{(1)} - \phi_\mu^{(1)}\|_\infty \\ \leq & \ c_1\left(\frac{5}{8}\tau_0^2\mu\right). \end{aligned}$$

The inequality $\|\overrightarrow{\omega}_2^\mu - \overrightarrow{\omega}_2^*\|_2^2 \leq c_2\left(\frac{5}{8}\tau_0^2\mu\right)$, can be proven in the same way.

(iii) By the above and the Squeeze Theorem, we have $\overrightarrow{\omega}_1^\mu \to \overrightarrow{\omega}_1^*$ as $\mu \to 0^+$. By the above and in a similar way, we obtain that $\overrightarrow{\omega}_2^\mu \to \overrightarrow{\omega}_2^*$ as $\mu \to 0^+$. □

An unknown sample point $x \in \mathbb{R}^n$ is assigned to class i ($i = +1$ or $i = -1$) by the following:

$$\text{class}(x) = \text{sgn}\left[\frac{\overrightarrow{w_1}^T x + b_1}{\|\overrightarrow{w_1}\|^2} + \frac{\overrightarrow{w_2}^T x + b_2}{\|\overrightarrow{w_2}\|^2}\right],$$

or

$$\text{class}(x) = \text{sgn}\left[\frac{\vec{\boldsymbol{\omega}_1}^T \vec{\boldsymbol{x}}}{\|\overrightarrow{w_1}\|^2} + \frac{\vec{\boldsymbol{\omega}_2}^T \vec{\boldsymbol{x}}}{\|\overrightarrow{w_2}\|^2}\right],$$

where $\vec{\boldsymbol{\omega}_i} = [\overrightarrow{w_i}^T, b_i]^T \in \mathbb{R}^{n+1}, \vec{\boldsymbol{x}} = [x, 1]^T$.

4.2.2 Quasi-Newton smooth generalized pinball twin bounded support vector machine

The focus of this section is on the application of the BFGS method, referred to as BFGS-SGPTBSVM, which we employed to solve a strongly convex differentiable problem. This approach is used in optimization to find the minimum of a strongly convex and C^2 function. As before, we focus on the strongly convex differentiable problem (4.7):

$$\min_{\overrightarrow{\omega_1}} \phi_\mu^{(1)}(\overrightarrow{\omega_1}) := \frac{1}{2}\sum_{i=1}^{m_1}(\overrightarrow{\omega_1}^T \overrightarrow{x_i}^{(1)})^2 + c_3\|\overrightarrow{\omega_1}\|^2 + \frac{c_1}{m_2}\sum_{i=1}^{m_2} P(\overrightarrow{\omega_1}^T \overrightarrow{x_i}^{(2)} + 1, \mu)$$

If $\overrightarrow{\omega_{1k}}$ denotes the value of $\overrightarrow{\omega_1}$ obtained in the k-th iteration step, then the gradient of the objective function $\phi_\mu^{(1)}$ at $\overrightarrow{\omega_{1k}}$ is

$$\nabla\phi_\mu^{(1)}(\overrightarrow{\omega_{1k}}) = \sum_{i=1}^{m_1} \overrightarrow{x_i}^{(1)}\left[\overrightarrow{x_i}^{(1)}\right]^T\overrightarrow{\omega_1} + 2c_3\overrightarrow{\omega_{1k}} + \frac{c_1}{m_2}\sum_{i=1}^{m_2}\overrightarrow{x_i}^{(2)}\partial P(u_{i_k},\mu),$$

where $u_{i_k} = \overrightarrow{\omega_{1k}}^T\overrightarrow{x_i}^{(2)} + 1$ and the partial derivative of $P(u,\mu)$ is

$$\partial P(u,\mu) = \begin{cases} \tau_1, & \frac{\epsilon_1}{\tau_1} + \tau_1\mu < u, \\ \frac{5}{2\tau_1^2\mu^3}(u - \frac{\epsilon_1}{\tau_1})^3 - \frac{3}{2\tau_1^4\mu^5}(u - \frac{\epsilon_1}{\tau_1})^5, & \frac{\epsilon_1}{\tau_1} < u \le \frac{\epsilon_1}{\tau_1} + \tau_1\mu, \\ 0, & -\frac{\epsilon_2}{\tau_2} \le u \le \frac{\epsilon_1}{\tau_1}, \\ \frac{5}{2\tau_2^2\mu^3}(u + \frac{\epsilon_2}{\tau_2})^3 - \frac{3}{2\tau_2^4\mu^5}(u + \frac{\epsilon_2}{\tau_2})^5, & -\frac{\epsilon_2}{\tau_2} - \tau_2\mu \le u < -\frac{\epsilon_2}{\tau_2}, \\ -\tau_2, & u < -\frac{\epsilon_2}{\tau_2} - \tau_2\mu. \end{cases}$$

The BFGS method is outlined as follows:

$$\overrightarrow{\omega_{1k+1}} = \overrightarrow{\omega_{1k}} + \alpha_k d_k. \tag{4.9}$$

where $d_k = -F(x_k)^{-1}\nabla f(x_k)$ and α_k is determined by the Armijo condition.

Let B_k be an approximate of the Hessian matrix $\nabla^2\phi_\mu^{(1)}(\overrightarrow{\omega_{1k}})$. For the next iteration, the Hessian matrix is updated by (2.35), that is,

$$B_{k+1} = B_k - \frac{B_k\Delta x_k\Delta x_k^T B_k}{\Delta x_k^T B_k\Delta x_k} + \frac{\Delta g_k\Delta g_k^T}{\Delta x_k^T\Delta g_k}.$$

where

$$\Delta x_k = \overrightarrow{\omega_{1k+1}} - \overrightarrow{\omega_{1k}},$$

and

$$\Delta g_k = \nabla\phi_\mu^{(1)}(\overrightarrow{\omega_{1k+1}}) - \nabla\phi_\mu^{(1)}(\overrightarrow{\omega_{1k}}).$$

The following list the details of the BFGS-SGPTBSVM algorithm.

Algorithm 2 BFGS-SGPTBSVM

Require: Given the starting point $\overrightarrow{\omega}_{1_0}$ and stopping criterion $\epsilon > 0$.

while $\|\nabla\phi_{\mu}^{(1)}(\overrightarrow{\omega_{1k}})\| \geq \epsilon$ **do**;

$$\begin{aligned} d_k = & \quad -H_k\nabla\phi_{\mu}^{(1)}(\overrightarrow{\omega_{1k}}) \\ = & \quad -\left(B_k\right)^{-1}\nabla\phi_{\mu}^{(1)}(\overrightarrow{\omega_{1k}}); \end{aligned}$$

Compute α_k to satisfy Armijo condition;

Set $\overrightarrow{\omega_{1k+1}} = \overrightarrow{\omega_{1k}} + \alpha_k d_k$;

Set $\Delta x_k = \overrightarrow{\omega_{1k+1}} - \overrightarrow{\omega_{1k}} = \alpha_k d_k$ and $\Delta g_k = \nabla\phi_{\mu}^{(1)}(\overrightarrow{\omega_{1k+1}}) - \nabla\phi_{\mu}^{(1)}(\overrightarrow{\omega_{1k}})$;

Compute H_{k+1} by means of (2.32), (2.36), or compute $\left(B_{k+1}\right)^{-1}$ by means of (2.35);

Set $k = k + 1$;

end while

The BFGS-SGPTBSVM algorithm of strongly convex differentiable problem (4.8) can be computed similar to the problem (4.7).

4.2.3 Nonlinear case

Many real-world datasets have complex structures where classes cannot be separated by a hyperplane. To address this, in the SVM or TBSVM models one maps the original feature space into a higher dimensional space, where a separating hyperplane can be found. It turns out that the mapping need not be computed, this is called the kernel trick, as we explain now.

Let $\Phi : \mathbb{R}^n \to \mathcal{H}$ be a mapping, where $\mathcal{H}$ is a Hilbert space. Here n is number of features in the given dataset. Let $\widetilde{\mathcal{H}}$ denote the linear span of $\{\Phi(\overrightarrow{x_1}), \ldots, \Phi(\overrightarrow{x_m})\}$. So

$\widetilde{\mathcal{H}}$ is a finite dimensional subspace of $\mathcal{H}$. We build our TBSVM in $\widetilde{\mathcal{H}}$;

$$\min_{\overrightarrow{w_1},b_1} \frac{1}{2}\sum_{i=1}^{m_1}(\overrightarrow{w_1}^T\Phi(\overrightarrow{x_i})+b_1)^2 + c_3\big(\|\overrightarrow{w_1}\|^2+b_1^2\big) + \frac{c_1}{m_2}\sum_{i=1}^{m_2} P\big((\overrightarrow{w_1}^T\Phi(\overrightarrow{x_i})+b_1)+1,\mu\big) \tag{4.10}$$

$$\min_{\overrightarrow{w_2},b_2} \frac{1}{2}\sum_{i=m_1+1}^{m}(\overrightarrow{w_2}^T\Phi(\overrightarrow{x_i})+b_2)^2 + c_4\big(\|\overrightarrow{w_2}\|^2+b_2^2\big) + \frac{c_2}{m_1}\sum_{i=1}^{m_1} P\big(1-(\overrightarrow{w_2}^T\Phi(\overrightarrow{x_i})+b_2),\mu\big) \tag{4.11}$$

Consider the symmetric mapping $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, called the kernel, given by

$$K(\overrightarrow{y_1},\overrightarrow{y_2}) = \Phi(\overrightarrow{y_1})^T\Phi(\overrightarrow{y_2}) = \Phi(\overrightarrow{y_2})^T\Phi(\overrightarrow{y_1}) = K(\overrightarrow{y_2},\overrightarrow{y_1}), y_i \in \mathbb{R}^n.$$

The Gramian Matrix is

$$\mathbb{X} = \big[K(\overrightarrow{x_i},\overrightarrow{x_j})\big]_{i,j} = \begin{bmatrix} K(\overrightarrow{x_1},\overrightarrow{x_1}) & K(\overrightarrow{x_1},\overrightarrow{x_2}) & \cdots & K(\overrightarrow{x_1},\overrightarrow{x_m}) \\ K(\overrightarrow{x_2},\overrightarrow{x_1}) & K(\overrightarrow{x_2},\overrightarrow{x_2}) & \cdots & K(\overrightarrow{x_2},\overrightarrow{x_m}) \\ \vdots & \vdots & \vdots & \vdots \\ K(\overrightarrow{x_m},\overrightarrow{x_1}) & K(\overrightarrow{x_m},\overrightarrow{x_2}) & \cdots & K(\overrightarrow{x_m},\overrightarrow{x_m}) \end{bmatrix}.$$

As $\overrightarrow{w_1},\overrightarrow{w_2} \in \widetilde{\mathcal{H}}$, we can write

$$\overrightarrow{w_1} = \sum_{j=1}^{m}\alpha_j\Phi(\overrightarrow{x_j}) \text{ and } \overrightarrow{w_2} = \sum_{j=1}^{m}\beta_j\Phi(\overrightarrow{x_j})$$

where $\alpha_j, \beta_j \in \mathbb{R}$ are not unique. Then (4.10) becomes

$$\begin{aligned}\min_{\overrightarrow{\alpha},b_1} \quad & \tfrac{1}{2}\sum_{i=1}^{m_1}\left(\left(\sum_{j=1}^{m}\alpha_j\Phi(\overrightarrow{x_j})\right)^T\Phi(\overrightarrow{x_i})+b_1\right)^2 + c_3\left(\left(\sum_{j=1}^{m}\alpha_j\Phi(\overrightarrow{x_j})\right)^T\left(\sum_{k=1}^{m}\alpha_k\Phi(\overrightarrow{x_k})\right)+b_1^2\right) \\ & + \tfrac{c_1}{m_2}\sum_{i=m_1+1}^{m} P\left(\sum_{j=1}^{m}\alpha_j\Phi(\overrightarrow{x_j})^T\Phi(\overrightarrow{x_i})+b_1)+1,\mu\right),\end{aligned} \tag{4.12}$$

where $\overrightarrow{a} = (a_1,a_2,...,a_m)^T \in \mathbb{R}^m, b_1 \in \mathbb{R}$, and we get

$$\begin{aligned}\min_{\overrightarrow{\alpha},b_1} \quad & \tfrac{1}{2}\sum_{i=1}^{m_1}\left(\sum_{j=1}^{m}K(\overrightarrow{x_i},\overrightarrow{x_j})\alpha_j+b_1\right)^2 + c_3\left(\sum_{i=1}^{m}\left(\sum_{k=1}^{m}\alpha_j\alpha_k K(\overrightarrow{x_i},\overrightarrow{x_j})\right)+b_1^2\right) \\ & + \tfrac{c_1}{m_2}\sum_{i=m_1+1}^{m} P\left(\sum_{j=1}^{m}K(\overrightarrow{x_i},\overrightarrow{x_j})\alpha_j+b_1)+1,\mu\right),\end{aligned} \tag{4.13}$$

i.e.

$$\min_{\overrightarrow{\alpha},b_1} \frac{1}{2}\sum_{i=1}^{m_1}\left(\mathbb{X}_i\overrightarrow{\alpha}+b_1\right)^2 + c_3\left(\overrightarrow{\alpha}^T\mathbb{X}\overrightarrow{\alpha}+b_1^2\right)+\frac{c_1}{m_2}\sum_{i=m_1+1}^{m} P\left((\mathbb{X}_i\overrightarrow{\alpha}+b_1)+1,\mu\right). \tag{4.14}$$

where $\mathbb{X}_i$ denotes the i-th row of $\mathbb{X}$. Similarly, (4.11) changes to

$$\min_{\overrightarrow{\beta},b_2} \frac{1}{2}\sum_{i=m_1+1}^{m}\left(\mathbb{X}_i\overrightarrow{\beta}+b_2\right)^2 + c_4\left(\overrightarrow{\beta}^T\mathbb{X}\overrightarrow{\beta}+b_2^2\right)+\frac{c_2}{m_1}\sum_{i=1}^{m_1} P\left(1-(\mathbb{X}_i\overrightarrow{\beta}+b_2),\mu\right). \tag{4.15}$$

An unknown sample point $\overrightarrow{x} \in \mathbb{R}^n$ is assigned to class i ($i=+1$ or $i=-1$) by the following:

$$\text{class}(\overrightarrow{x}) = \text{sgn}\left[\frac{\overrightarrow{w_1}^T\Phi(\overrightarrow{x})+b_1}{\|\overrightarrow{w_1}\|^2}+\frac{\overrightarrow{w_2}^T\Phi(\overrightarrow{x})+b_2}{\|\overrightarrow{w_2}\|^2}\right]. \tag{4.16}$$

Since $\overrightarrow{w_1}=\sum_{i=1}^{m}\alpha_i\Phi(\overrightarrow{x_i})$, then

$$\overrightarrow{w_1}^T\Phi(\overrightarrow{x}) = \sum_{i=1}^{m}\alpha_i K(\overrightarrow{x_i},\overrightarrow{x})$$

and

$$\|\overrightarrow{w_1}\|^2 = \overrightarrow{w_1}^T\overrightarrow{w_1} = \sum_{i,j=1}^{m}\alpha_i K(\overrightarrow{x_i},\overrightarrow{x})\alpha_j = \overrightarrow{\alpha}^T K\overrightarrow{\alpha}$$

and similarly for $\overrightarrow{w_2}$, so that (4.16) becomes

$$\text{class}(\overrightarrow{x}) = \text{sgn}\left[\frac{\sum_{i=1}^{m}\alpha_i K(\overrightarrow{x_i},\overrightarrow{x})}{\sqrt{\alpha^T K\alpha_k}}+\frac{\sum_{i=1}^{m}\beta_i K(\overrightarrow{x_i},\overrightarrow{x})}{\sqrt{\beta^T K\beta_k}}\right].$$

4.3 Numerical Experiments

In this chapter, we will present the performance of our proposed algorithm (BFGS-SGPTBSVM). We selected 9 datasets from the UCI dataset collection, namely the Australian, Diabetes, Ionosphere, Monk2, Phoneme, Ring, Saheart, Spectfheart, and Twonorm datasets for experimentation, as shown in Table 4.1. Moreover, we will present the comparison of 3 algorithms, including the BFGS-SPTBSVM (loss function is the smooth pinball loss), BFGS-GPTBSVM (loss function is the generalized pinball loss), and BFGS-SGPTBSVM (loss function is the smooth generalized pinball loss). These algorithms use the smooth approximate loss function of the pinball loss function, generalized pinball loss function, and our smooth approximate loss function of the generalized pinball loss function, respectively. All of these algorithms are trained with the above 9 datasets on TBSVM.

Table 4.1 9 datasets.

Datasets	Instances	Features
Australian	690	14
Diabetes	768	8
Monk2	432	6
Phoneme	5404	5
Saheart	462	9
Spectfheart	267	44
Twonorm	7400	20
Ring	7400	20
Ionosphere	351	33

4.3.1 Performance on UCI datasets

In this step, to assess the performance of the classifiers, a 5-fold cross-validation technique was employed for all experiments. First we used random grid search to optimize the parameters of the TBSVM model and the parameter μ of our smooth generalized pinball loss function $P(u, \mu)$, as shown in Tables 4.2 and 4.3. After having obtained the best parameters, we trained our proposed BFGS-SGPTBSVM algorithm using these parameters. We divided the experiment into 2 cases, "fixed splits" and "variable splits". In the case of fixed splits, the same 5-fold cross-validation splits were used as in parameter optimization. In case of variable splits, the average results of 50 different tests are reported, where in each test, the 5-fold cross-validation splits were different, which is more realistic than with fixed splits. The experimental results are presented in Tables 4.4, 4.5, 4.6, and 4.7. The accuracy (%), standard deviation, and training time (in second) are used for evaluation, denoted as Acc, sd, and time(s), respectively.

Table 4.2 Description of the parameters in the linear case.

Datasets	Model / Parameter		
	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
	C_1, C_2, C_3, C_4, ϵ^+, τ^+, ϵ^-, τ^-	C_1, C_2, C_3, C_4, ${\epsilon_1}^+$, ${\epsilon_2}^+$, ${\tau_1}^+$, ${\tau_2}^+$, ${\epsilon_1}^-$, ${\epsilon_2}^-$, ${\tau_1}^-$, ${\tau_2}^-$	C_1, C_2, C_3, C_4, ${\epsilon_1}^+$, ${\epsilon_2}^+$, ${\tau_1}^+$, ${\tau_2}^+$, μ^+, ${\epsilon_1}^-$, ${\epsilon_2}^-$, ${\tau_1}^-$, ${\tau_2}^-$, μ^-
Australian	0.8, 0.4, 0.2, 0.2, 0.2, 0.5, 0.6, 0.5	0.8, 0.2, 1.6, 1.2, 0.1, 2, 0.9, 0.7, 0.25, 2, 0.4, 0.7	0.2, 50, 1.6, 1.6, 0.7, 1, 0.8, 0.9, 0.01, 1, 0.7, 1.2, 1, 2
Diabetes	3.2, 1.6, 0.1, 1.6, 0.2, 0.7, 0.1, 0.1	1.6, 0.4 , 0.2, 0.4 0.1, 0.45, 1, 0.3, 0.1, 0.5, 0.9, 0.1	50, 0.2, 0.4, 0.8, 0.8, 2, 0.9, 0.5, 2, 0.8, 2, 0.9, 0.5, 1
Monk2	0.2, 0.4, 0.1, 0.1, 1, 0.9, 0.2, 0.1	6, 0.4, 0.1, 3.2, 0.25, 0.7, 0.7, 0.3, 0.01, 0.7, 0.6, 1	6, 0.8, 0.1, 0.1, 0.25, 2, 0.6, 0.3, 0.01, 0.01, 0.1, 0.7, 0.3, 0.4
Phoneme	3.2, 0.8, 1.6, 0.1, 0.1, 0.3, 0.2, 0.1	0.4, 0.4, 0.2, 2, 0.45, 1, 0.6, 0.1, 0.05, 0.7, 0.4, 1	0.2, 0.1, 0.4, 1.6, 0.7, 0.25, 0.9, 0.9, 0.4, 0.45, 0.2, 0.7, 0.1, 0.4
Saheart	3.2, 0.2, 1.6, 0.4, 0.01, 0.3, 0.6, 0.5	1.6, 0.8, 3.2, 1.6 0.45, 0.7, 0.8, 0.5, 0.25, 0.45, 0.7, 0.7	3.2, 0.2, 0.4, 0.1, 0.8, 0.25, 0.9, 0.3, 0.4, 0.25, 1, 0.7, 0.9, 0.4
Spectfheart	6, 10, 3.2, 0.4, 0.01, 0.5, 0.2, 0.7	50, 80, 3.2, 3.2 0.1, 0.45, 0.9, 0.7, 0.8, 0.45, 1, 0.7	0.6, 20, 3.2, 0.4, 0.7, 0.45, 0.8, 0.1, 1, 0.45, 0.7, 0.6, 0.1, 0.4
Twonorm	0.8, 0.8, 0.2, 0.2, 2, 0.3, 0.1, 0.5	0.2, 1.6, 1.6, 3.2, 0.05, 1.5, 0.9, 0.7, 0.25, 0.8, 0.9, 1	0.2, 0.4, 0.8, 0.8, 0.1, 0.8, 0.9, 1, 0.8, 0.25, 1, 0.9, 0.9, 0.6
Ring	0.4, 0.8, 1.6, 0.4, 0.6, 0.9, 0.01, 0.7	0.2, 0.4, 3.2, 1.6, 0.1, 2, 0.6, 1, 0.25, 0.8, 0.4, 1	0.2, 0.4, 3.2, 1.6, 0.5, 0.8, 0.4, 0.5, 0.01, 0.1, 0.45, 1, 1, 2

Table 4.2 (Continued).

Datasets	Model / Parameter		
	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
	C_1, C_2, C_3, C_4, ϵ^+, τ^+, ϵ^-, τ^-	C_1, C_2, C_3, C_4, ${\epsilon_1}^+$, ${\epsilon_2}^+$, ${\tau_1}^+$, ${\tau_2}^+$, ${\epsilon_1}^-$, ${\epsilon_2}^-$, ${\tau_1}^-$, ${\tau_2}^-$	C_1, C_2, C_3, C_4, ${\epsilon_1}^+$, ${\epsilon_2}^+$, ${\tau_1}^+$, ${\tau_2}^+$, μ^+, ${\epsilon_1}^-$, ${\epsilon_2}^-$, ${\tau_1}^-$, ${\tau_2}^-$, μ^-
Ionosphere	0.1, 0.8, 0.2, 3.2, 2, 0.1, 0.1, 0.5	0.1, 0.1, 0.4, 3.2, 0.1, 0.1, 0.3, 3.5, 0.5, 0.8, 0.4, 0.1	0.1, 0.2, 0.2, 3.2, 0.5, 0.8, 0.4, 1, 0.4, 0.45, 0.1, 0.7, 0.5, 0.2

Table 4.3 Description of the parameters using an RBFSmapler.

Datasets	Model / Parameter		
	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
	C_1, C_2, C_3, C_4, ϵ^+, τ^+, ϵ^-, τ^-	C_1, C_2, C_3, C_4, ${\epsilon_1}^+$, ${\epsilon_2}^+$, ${\tau_1}^+$, ${\tau_2}^+$, ${\epsilon_1}^-$, ${\epsilon_2}^-$, ${\tau_1}^-$, ${\tau_2}^-$	C_1, C_2, C_3, C_4, ${\epsilon_1}^+$, ${\epsilon_2}^+$, ${\tau_1}^+$, ${\tau_2}^+$, μ^+, ${\epsilon_1}^-$, ${\epsilon_2}^-$, ${\tau_1}^-$, ${\tau_2}^-$, μ^-
Australian	0.2, 1.6, 1.6, 1.6, 0.1, 0.3, 0.01, 0.5	0.4, 0.4, 0.8, 0.4, 0.05, 0.1, 0.7, 0.9, 0.8, 0.1, 1, 0.1	3.2, 25, 0.1, 0.1, 0.25, 1.5, 0.4, 0.9, 0.6, 0.1, 1, 0.8, 0.1, 2
Diabetes	0.8, 1.6, 0.2, 0.1, 0.1, 0.7, 0.1, 0.5	1.6, 0.2 , 0.1, 0.2 0.25, 0.8, 0.8, 0.7, 0.25, 0.1, 0.4, 1	3.2, 6, 0.2, 0.2, 0.01, 2, 0.7, 0.3, 1, 0.25, 2, 0.8, 0.1, 1
Monk2	20, 0.2, 0.4, 1.6, 0.01, 0.5, 0.4, 0.5	25, 3.2, 0.1, 0.1, 0.45, 1, 0.6, 0.5, 0.1, 0.8, 0.6, 0.1	6, 20, 0.1, 0.1, 0.1, 0.25, 0.7, 0.1, 0.2, 0.45, 0.8, 0.8, 0.7, 0.4
Phoneme	0.2, 0.8, 0.4, 0.2, 2, 0.1, 0.01, 0.3	1.6, 0.1, 0.2, 0.4, 0.1, 2, 0.9, 0.3, 0.25, 0.8, 0.4, 0.7	25, 3.2, 0.1, 0.1, 1, 0.8, 1.2, 0.3, 0.4, 0.7, 0.7, 1.2, 1, 0.2
Saheart	0.2, 0.2, 0.4, 0.1, 0.01, 0.5, 0.4, 0.3	1.6, 0.2, 0.1, 0.2 0.01, 0.25, 1.2, 0.1, 0.05, 1, 1.2, 0.3	10, 0.2, 0.2, 0.2, 0.8, 1, 1, 0.7, 0.6, 0.1, 1.5, 1, 1, 0.4
Spectfheart	10, 20, 0.4, 1.6, 0.01, 0.5, 0.4, 0.1	0.8, 0.4, 0.1, 0.2 0.25, 0.1, 0.6, 0.1, 0.25, 2, 1, 0.9	50, 0.8, 0.1, 0.2, 0.05, 0.8, 1, 0.9, 2, 0.05, 0.8, 0.4, 0.1, 0.6
Twonorm	0.2, 0.1, 0.1, 0.1, 0.01, 0.9, 0.1, 0.3	3.2, 0.4, 0.1, 0.1, 0.1, 2, 0.9, 0.1, 0.8, 0.7, 0.9, 0.5	0.1, 20, 0.2, 0.1, 0.45, 2, 0.9, 0.1, 0.01, 0.05, 0.7, 0.9, 0.1, 1
Ring	0.2, 0.4, 0.4, 1.6, 0.1, 0.3, 0.1, 0.1	3.2, 10, 0.1, 0.1, 0.01, 0.7, 0.9, 0.5, 0.7, 0.25, 1, 0.1	3.2, 20, 0.1, 0.1, 0.01, 2, 0.9, 0.1, 0.01, 0.45, 1, 1, 0.5, 0.01

Table 4.3 (Continued).

Datasets	Model / Parameter		
	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
	C_1, C_2, C_3, C_4, ϵ^+, τ^+, ϵ^-, τ^-	C_1, C_2, C_3, C_4, ${\epsilon_1}^+$, ${\epsilon_2}^+$, ${\tau_1}^+$, ${\tau_2}^+$, ${\epsilon_1}^-$, ${\epsilon_2}^-$, ${\tau_1}^-$, ${\tau_2}^-$	C_1, C_2, C_3, C_4, ${\epsilon_1}^+$, ${\epsilon_2}^+$, ${\tau_1}^+$, ${\tau_2}^+$, μ^+, ${\epsilon_1}^-$, ${\epsilon_2}^-$, ${\tau_1}^-$, ${\tau_2}^-$, μ^-
Ionosphere	0.4, 20, 0.2, 0.1, 0.2, 0.7, 0.01, 0.5	1.6, 10, 0.1, 0.1, 0.05, 0.7, 0.8, 1, 0.45, 0.25, 0.6, 0.1	10, 20, 0.1, 0.2, 0.1, 0.1, 1.2, 0.1, 0.01, 0.25, 1, 1.2, 0.9, 0.4

Table 4.4 Fixed splits; Performance of different algorithms, linear model.

Datasets	Acc±sd		
	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
Australian	**88.4058±2.2915**	87.8261±2.1201	87.8261±2.5681
time(s)	**0.21679**	0.370989	0.877043
Diabetes	77.8593±2.5482	77.4688±3.6535	**77.8627±2.2755**
time(s)	**0.092696**	0.163853	0.253656
Monk2	83.3280±4.5861	**87.7252±3.9495**	87.0356±4.0527
time(s)	**0.047582**	0.230813	0.178027
Phoneme	77.8127±0.7475	**77.9793±0.4456**	77.9237±0.5031
time(s)	**0.335106**	0.480017	0.678361
Saheart	**74.0159±5.2307**	73.8125±4.4559	73.8055±3.6154
time(s)	**0.132691**	0.186257	0.246309
Spectfheart	81.6562±5.7405	81.6771±4.9851	**83.9064±5.6250**
time(s)	**0.851100**	4.725302	1.593111
Twonorm	97.8649±0.2550	97.9054±0.3963	**97.9324±0.3954**
time(s)	**0.430177**	0.704223	0.658462
Ring	76.7973±1.7075	**76.9189±1.5664**	76.8378±1.7058
time(s)	**0.556805**	0.667775	0.902595
Ionosphere	**89.4567±2.6563**	89.1751±2.7936	88.8893±3.3052
time(s)	**0.399795**	0.56281	0.680536

Table 4.5 Fixed splits; Performance of different algorithms, RBFSampler.

Datasets	Acc±sd		
	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
Australian	82.4638±2.3098	83.6232±2.8102	**84.0580±1.8332**
time(s)	1.736451	**0.666318**	2.079387
Diabetes	74.9979±1.8926	75.3875±3.2180	**75.9087±4.1264**
time(s)	5.969881	1.443527	**1.220819**
Monk2	91.4274±1.7705	**95.5948±1.5548**	94.8998±1.5939
time(s)	**0.772924**	3.983829	1.956448
Phoneme	79.4781±0.5841	79.8668±0.6885	**80.3479±0.9708**
time(s)	**2.184906**	2.325309	3.43196
Saheart	74.0112±4.6302	74.0089±5.8561	**74.2380±4.2839**
time(s)	1.177418	1.585177	**1.247391**
Spectfheart	83.5360±4.6123	83.1586±4.0473	**84.6681±5.5610**
time(s)	**0.471961**	1.081096	1.35906
Twonorm	93.7162±0.8547	**94.6486±0.5881**	94.4595±0.7914
time(s)	4.962333	**2.749745**	2.804496
Ring	95.0676±0.4441	**96.0000±0.4041**	95.7973±0.5429
time(s)	**1.994101**	3.135659	3.292427
Ionosphere	90.3219±2.8753	**93.1630±2.4558**	91.7384±3.5440
time(s)	**0.955793**	2.926564	2.245855

Table 4.6 Variable splits; Performance of different algorithms, linear model.

Datasets	Acc±sd		
	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
Australian	86.6000±2.3183	86.7130±2.3182	**86.9768±2.5594**
time(s)	**0.218793**	0.362228	0.813517
Diabetes	75.6744±3.1206	**76.6450±2.7918**	75.4544±2.8918
time(s)	**0.114313**	0.159036	0.195348
Monk2	79.8147±5.4047	**86.4165±3.6584**	83.7312±4.2069
time(s)	**0.047766**	0.210939	0.177863
Phoneme	77.4807±1.0521	**77.7872±1.1616**	76.9989±1.4930
time(s)	**0.338369**	0.497339	0.747009
Saheart	72.2705±3.7282	**72.6528±4.3875**	72.6122±3.9976
time(s)	**0.123355**	0.183273	0.270993
Spectfheart	77.2806±7.2642	**79.0070±4.9500**	78.5426±7.4243
time(s)	**0.772305**	5.034258	1.382591
Twonorm	97.7586±0.3190	**97.8059±0.3403**	97.7232±0.3141
time(s)	**0.413597**	0.710938	0.656289
Ring	76.3803±0.9844	76.6319±0.9745	**76.6335±0.9697**
time(s)	**0.538940**	0.678837	0.889948
Ionosphere	86.9708±3.6478	**88.3367±3.4795**	88.1153±3.2979
time(s)	**0.464421**	0.563288	0.726181

Table 4.7 Variable splits; Performance of different algorithms, RBFSampler.

Datasets	Acc±sd		
	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
Australian	79.9333±3.2045	81.1130±3.1104	**81.9826±3.0698**
time(s)	1.100957	**0.692429**	1.772433
Diabetes	70.8914±3.5184	**74.4581±3.1129**	73.7404±3.0545
time(s)	2.557176	1.513731	**1.235133**
Monk2	89.4627±4.2806	**94.3527±2.2290**	93.3532±2.7422
time(s)	**1.751879**	4.059142	2.015005
Phoneme	78.1296±2.2966	**79.8060±1.0184**	78.9168±1.7498
time(s)	2.553766	**1.772117**	4.101333
Saheart	69.8903±4.3887	71.9447±4.3071	**72.1198±3.8393**
time(s)	1.996333	1.579952	**1.461621**
Spectfheart	72.5744±10.0496	79.9280±4.8487	**80.0032±4.6938**
time(s)	**0.864064**	1.085297	1.819914
Twonorm	93.3878±0.7616	**94.0049±0.9483**	93.8722±0.7545
time(s)	6.243437	2.917959	**2.576174**
Ring	**94.9878±0.4759**	94.9262±1.6240	93.4192±3.6378
time(s)	**1.546431**	3.376859	3.962855
Ionosphere	86.9302±5.1465	**91.2884±2.8932**	90.3624±3.3991
time(s)	**1.512256**	4.066119	2.140556

Moreover, normally distributed noise with a mean of zero and a standard deviation (r) was added to the selected UCI datasets at ratios of r=0.02, 0.05, 0.10, 0.15, 0.20, 0.25, and 0.30 standard deviation to test the noise sensitivity of the algorithms. Figure 4.2 shows the performance at different noise levels. As the Twonorm dataset is the dataset that gives the best performance, its results are presented first.

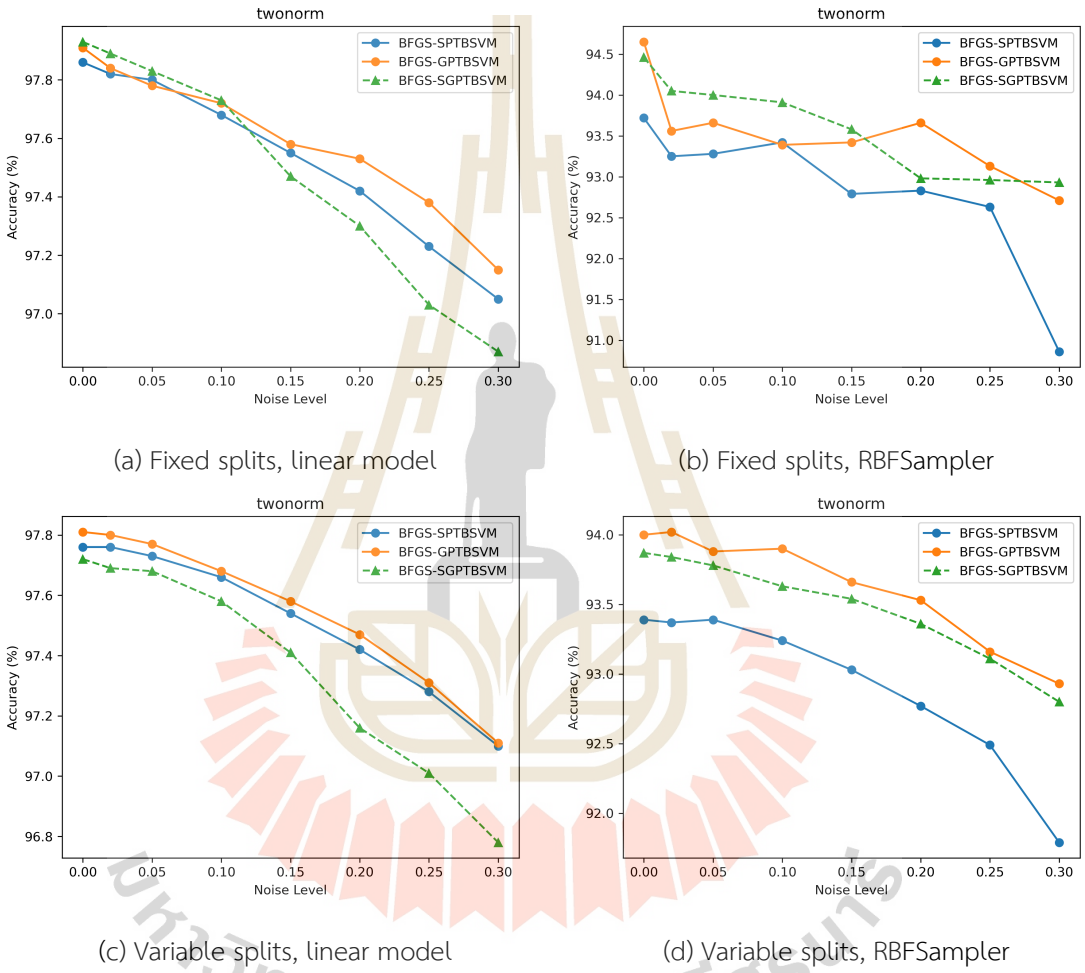
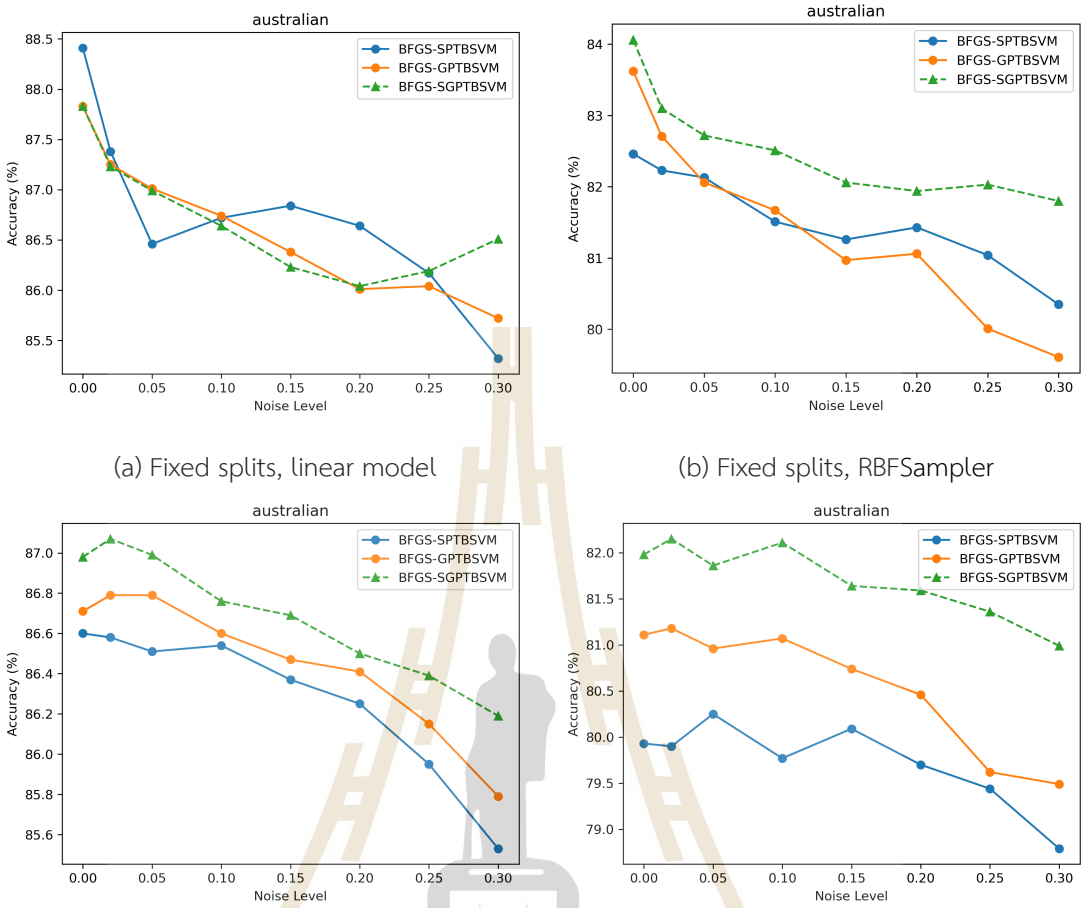


Figure 4.2 Performance of different noise levels for the Twonorm dataset.

From Figure 4.2, we observe that as noise increases, the accuracy of all algorithms tends to decrease. Figure 4.3 shows the performance of different noise levels for the Australian dataset. We observe that our smooth generalized pinball loss function retains the noise insensitivity property of the generalized pinball loss function.



(a) Fixed splits, linear model

(b) Fixed splits, RBFSampler

(c) Variable splits, linear model

(d) Variable splits, RBFSampler

Figure 4.3 Performance of different noise levels for the Australian dataset.

4.3.2 Friedman Test

For the statistical analysis, we use the Friedman test to evaluate the classification performance of the 3 methods on the various datasets. The Friedman test is a non-parametric statistical test designed to detect differences in the performance of multiple models across multiple datasets. It is commonly applied when evaluating the effectiveness of SVM models against other algorithms or variations of SVM.

First, we set:

H_0 : there is no significant difference between the 3 algorithms.

H_1 : there is a significant difference between the 3 algorithms.

We evaluate $k = 3$ algorithms across $N = 9$ datasets for the purpose of compar-

ison.

Fixed splits

The ranks of the 3 algorithms on accuracy for the 9 datasets, is shown in Tables 4.8 and 4.9.

Table 4.8 Fixed splits; Average ranks of different algorithms, linear model.

Datasets	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
Australian	1	2.5	2.5
Diabetes	2	3	1
Monk2	3	1	2
Phoneme	3	1	2
Saheart	1	2	3
Spectfheart	3	2	1
Twonorm	3	2	1
Ring	3	1	2
Ionosphere	1	2	3
Average rank	2.2222	1.8333	1.9444

Table 4.9 Fixed splits; Average ranks of different algorithms, RBFSampler.

Datasets	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
Australian	3	2	1
Diabetes	3	2	1
Monk2	3	1	2
Phoneme	3	2	1
Saheart	2	3	1
Spectfheart	2	3	1
Twonorm	3	1	2
Ring	3	1	2
Ionosphere	3	1	2
Average rank	2.7778	1.7778	1.4444

In the linear case, we get $\chi_F^2 = 0.7187$ and $F_F = 0.3327$. After looking up the F-table, we obtain that the critical value of $F(2, 16)$ is 3.63 for the confidence level $\alpha = 0.05$ and $0.3327 < 3.63$. Therefore, the null hypothesis H_0 is accepted. That is, there is no significant difference between the 3 algorithms.

On the contrary, when using the RBFSampler, we get $\chi_F^2 = 8.6673$ and $F_F = 7.4296$. After looking up the F-table, we obtain that the critical value of $F(2, 16)$ is 3.63 for the confidence level $\alpha = 0.05$ and $7.4296 > 3.63$. Therefore, the null hypothesis H_0 is rejected. That is, there is a significant difference between the 3 algorithms. In addition, Table 4.9 shows that the average rank of the BFGS-SGPTBSVM algorithm is lower than that of other methods.

Using the same method, the average ranks are calculated based on the performance results with different noise ratios, as presented in Tables 4.10 and 4.11.

Table 4.10 Fixed splits; Average ranks of different algorithms with noise, linear model.

Noise ratio	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
r=0.02	2.1111	1.6667	2.2222
r=0.05	2.4444	1.7778	1.7778
r=0.1	2.4444	1.5556	2.1111
r=0.15	2.1111	1.5556	2.3333
r=0.20	2.0000	1.8889	2.1111
r=0.25	2.2222	2.0000	1.7778
r=0.30	2.2222	1.8889	1.8889

Table 4.11 Fixed splits; Average ranks of different algorithms with noise, RBFSampler.

Noise ratio	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
r=0.02	2.4444	1.8889	1.6667
r=0.05	2.5556	1.6667	1.7778
r=0.1	2.5556	1.7778	1.6667
r=0.15	2.5556	1.6667	1.7778
r=0.20	2.6667	1.5556	1.7778
r=0.25	2.6667	1.3333	2.0000
r=0.30	2.6667	1.6667	1.6667

In the linear case, the test values are $\chi^2_{F(r=0.02)} = 1.5556$, $F_{F(r=0.02)} = 0.7568$, $\chi^2_{F(r=0.05)} = 2.6667$, $F_{F(r=0.05)} = 1.3913$, $\chi^2_{F(r=0.10)} = 7.6667$, $F_{F(r=0.10)} = 5.9355$, $\chi^2_{F(r=0.15)} = 2.8889$, $F_{F(r=0.15)} = 1.5294$, $\chi^2_{F(r=0.20)} = 0.2222$, $F_{F(r=0.20)} = 0.1000$, $\chi^2_{F(r=0.25)} = 0.8889$, $F_{F(r=0.25)} = 0.4156$, $\chi^2_{F(r=0.30)} = 0.6667$, and $F_{F(r=0.30)} = 0.3077$. Since $F_{F(r=0.02)}$, $F_{F(r=0.05)}$, $F_{F(r=0.15)}$, $F_{F(r=0.20)}$, $F_{F(r=0.25)}$, and $F_{F(r=0.30)}$ are less than 3.63, Therefore, the null hypotheses H_0 is accepted. Conversely, $F_{F(r=0.10)}$

is greater than 3.63, the null hypothesis H_0 is rejected. That is, there is a significant difference between the 3 algorithms on the datasets with 10% noise added.

In the RBFSampler, the test values are $\chi^2_{F(r=0.02)} = 2.8889$, $F_{F(r=0.02)} = 1.5294$, $\chi^2_{F(r=0.05)} = 4.2222$, $F_{F(r=0.05)} = 2.4516$, $\chi^2_{F(r=0.10)} = 4.2222$, $F_{F(r=0.10)} = 2.4516$, $\chi^2_{F(r=0.15)} = 4.2222$, $F_{F(r=0.15)} = 2.4516$, $\chi^2_{F(r=0.20)} = 6.2222$, $F_{F(r=0.20)} = 4.2264$, $\chi^2_{F(r=0.25)} = 8.0000$, $F_{F(r=0.25)} = 6.4000$, $\chi^2_{F(r=0.30)} = 6.0000$, and $F_{F(r=0.30)} = 4.0000$. Since $F_{F(r=0.02)}$, $F_{F(r=0.05)}$, $F_{F(r=0.10)}$, and $F_{F(r=0.15)}$ are less than 3.63, Therefore, the null hypotheses H_0 is accepted. Conversely, $F_{F(r=0.20)}$,$F_{F(r=0.25)}$, and $F_{F(r=0.30)}$ are greater than 3.63, the null hypotheses H_0 is rejected. That is, there is a significant difference between the 3 algorithms on the datasets with 20%, 25%, and 30% noise added.

Variable splits

The ranks of the 3 algorithms on accuracy for the 9 datasets, is shown in Tables 4.12 and 4.13.

Table 4.12 Variable splits; Average ranks of different algorithms, linear model.

Datasets	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
Australian	3	2	1
Diabetes	2	1	3
Monk2	3	1	2
Phoneme	2	1	3
Saheart	3	1	2
Spectfheart	3	1	2
Twonorm	2	1	3
Ring	3	2	1
Ionosphere	3	1	2
Average rank	2.6667	1.2222	2.1111

Table 4.13 Variable splits; Average ranks of different algorithms, RBFSampler.

Datasets	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
Australian	3	2	1
Diabetes	3	1	2
Monk2	3	1	2
Phoneme	3	1	2
Saheart	3	2	1
Spectfheart	1	2	1
Twonorm	3	1	2
Ring	1	2	3
Ionosphere	3	1	2
Average rank	2.7778	1.4444	1.7778

In the linear case, we get $\chi_F^2 = 9.5562$ and $F_F = 9.0539$. After looking up the F-table, we obtain that the critical value of $F(2, 16)$ is 3.63 for the confidence level $\alpha = 0.05$ and $9.0539 > 3.63$. Therefore, the null hypothesis H_0 is rejected. That is, there is a significant difference between the 3 algorithms.

In the same way, in the RBFSampler, we get $\chi_F^2 = 8.6673$ and $F_F = 7.4296$. After looking up the F-table, we obtain the critical value of $F(2, 16)$ is 3.63 for the confidence level $\alpha = 0.05$ and $7.4296 > 3.63$. Therefore, the null hypothesis H_0 is rejected. That is, there is a significant difference between the 3 algorithms.

Using the same method, the average ranks are calculated based on the performance results with different noise ratios, as presented in Tables 4.14 and 4.15.

Table 4.14 Fixed splits; Average ranks of different algorithms with noise, linear model.

Noise ratio	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
r=0.02	2.5556	1.2222	2.2222
r=0.05	2.5556	1.2222	2.2222
r=0.1	2.5556	1.2222	2.2222
r=0.15	2.7778	1.2222	2.0000
r=0.20	2.7778	1.2222	2.0000
r=0.25	2.6667	1.4444	1.8889
r=0.30	2.5556	1.6667	1.7778

Table 4.15 Fixed splits; Average ranks of different algorithms with noise, RBFSampler.

Noise ratio	BFGS-SPTBSVM	BFGS-GPTBSVM	BFGS-SGPTBSVM
r=0.02	2.7778	1.2222	2.0000
r=0.05	2.7778	1.4444	1.7778
r=0.1	2.7778	1.4444	1.7778
r=0.15	2.6667	1.3333	2.0000
r=0.20	2.6667	1.4444	1.8889
r=0.25	2.8889	1.3333	1.7778
r=0.30	2.6667	1.4444	1.8889

In the linear case, the test values are $\chi^2_{F(r=0.02)} = 8.6667$, $F_{F(r=0.02)} = 7.4286$, $\chi^2_{F(r=0.05)} = 8.6667$, $F_{F(r=0.05)} = 7.4286$, $\chi^2_{F(r=0.10)} = 8.6667$, $F_{F(r=0.10)} = 7.4286$, $\chi^2_{F(r=0.15)} = 10.8889$, $F_{F(r=0.15)} = 12.2500$, $\chi^2_{F(r=0.20)} = 10.8889$, $F_{F(r=0.20)} = 12.2500$, $\chi^2_{F(r=0.25)} = 6.8889$, $F_{F(r=0.25)} = 4.9600$, $\chi^2_{F(r=0.30)} = 4.2222$, and $F_{F(r=0.30)} = 2.4516$. Since $F_{F(r=0.30)}$ is less than 3.63, Therefore, the null hypothesis H_0 is accepted. Conversely, $F_{F(r=0.02)}$, $F_{F(r=0.05)}$, $F_{F(r=0.10)}$, $F_{F(r=0.15)}$, $F_{F(r=0.20)}$,

and $F_{F(r=0.25)}$, are greater than 3.63, the null hypotheses H_0 is rejected. That is, there is a significant difference between the 3 algorithms on the datasets with 2%, 5%, 10%, 15%, 20 and 25% noise added.

In the RBFSampler, the test values are $\chi^2_{F(r=0.02)} = 10.8889$, $F_{F(r=0.02)} = 12.2500$, $\chi^2_{F(r=0.05)} = 8.6667$, $F_{F(r=0.05)} = 7.4286$, $\chi^2_{F(r=0.10)} = 8.6667$, $F_{F(r=0.10)} = 7.4286$, $\chi^2_{F(r=0.15)} = 8.0000$, $F_{F(r=0.15)} = 6.4000$, $\chi^2_{F(r=0.20)} = 6.8889$, $F_{F(r=0.20)} = 4.9600$, $\chi^2_{F(r=0.25)} = 11.5556$, $F_{F(r=0.25)} = 14.3448$, $\chi^2_{F(r=0.30)} = 6.8888$, and $F_{F(r=0.30)} = 4.9600$. Since all of the test values are greater than 3.63, the null hypotheses H_0 is rejected. That is, there is a significant difference between the 3 algorithms on the datasets with all noise added.

CHAPTER V

CONCLUSION

This research has introduced a family of smooth generalized pinball loss functions to address the issues associated with non-differentiability found in traditional loss functions like hinge loss, pinball loss, and generalized pinball loss function. All of these loss functions cause the objective function of SVM models with them to become non-differentiable. Based on this, a novel smooth generalized pinball loss function is proposed, and a novel twin bounded support vector machine model with smooth generalized pinball loss function is obtained. Moreover, we proved that the generalized pinball loss function can be approximated by the proposed smooth generalized pinball loss function in the uniform norm with arbitrary precision, and the solution of our model is unique and still converges to the exact problem. In experiments, we selected 9 UCI datasets and used a quasi-Newton method to solve the corresponding strongly convex unconstrained optimization problems with twice continuously differentiable objective function. We compare our proposed BFGS-SGPTBSVM algorithm with BFGS-GPTBSVM and BFGS-SPTBSVM algorithms in terms of classification performance, accuracy and computational speed. Furthermore, from this experiment we found the proposed BFGS-SGPTBSVM algorithm has the best performance in the TBSVM with RBFSampler.

For future research, we plan to improve the classification model using our loss function and find the better technique parameters optimization, so that our loss function to improve our algorithm is better.

REFERENCES

REFERENCES

Cortes, C., and Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273-297. https://doi.org/10.1007/BF00994018

Huang, X., Shi, L., and Suykens, J. A. (2013). Support vector machine classifier with pinball loss. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36*(5), 984-997. https://doi.org/10.1109/TPAMI.2013.178

Joachims, T. (1998, April). Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Nachine Learning* (pp. 137-142). Berlin, Heidelberg: Springer Berlin Heidelberg.

Khemchandani, R., and Chandra, S. (2007). Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(5), 905-910.

Li, K., and Lv, Z. (2021). Smooth twin bounded support vector machine with pinball loss. *Applied Intelligence, 51*, 5489-5505.

Ma, J., Xia, D., Guo, H., Wang, Y., Niu, X., Liu, Z., and Jiang, S. (2022). Metaheuristic-based support vector regression for landslide displacement prediction: A comparative study. *Landslides, 19*(10), 2489-2511.

Makmuang, D., Ratiphaphongthon, W., and Wangkeeree, R. (2023). Smooth support vector machine with generalized pinball loss for Pattern Classification. *The Journal of Supercomputing, 79*(11), 11684-11706.

Nocedal, J., and Wright, S. J. (Eds.). (1999). *Numerical Optimization*. New York, NY: Springer New York.

Panup, W., Ratipapongton, W., and Wangkeeree, R. (2022). A novel twin support vector

machine with generalized pinball loss function for pattern classification. *Symmetry, 14*(2), 289.

Rastogi, R., Pal, A., and Chandra, S. (2018). Generalized Pinball Loss SVMs. *Neurocomputing, 322*, 151-165. https://doi.org/10.1016/j.neucom.2018.08.079

Shao, Y. H., Zhang, C. H., Wang, X. B., and Deng, N. Y. (2011). Improvements on twin support vector machines. *IEEE Transactions on Neural Networks, 22*(6), 962-968.

Shifei, D., Junzhao, Y., Bingjuan, Q., and Huajuan, H. (2014). An overview on twin support vector machines. *Artificial Intelligence Review, 42*, 245-252.

Xu, Y., Yang, Z., and Pan, X. (2016). A novel twin support-vector machine with pinball loss. *IEEE Transactions on Neural Networks and Learning Systems, 28*(2), 359-370.

Yin, H., Jiao, X., Chai, Y., and Fang, B. (2015). Scene classification based on single-layer SAE and SVM. *Expert Systems with Applications, 42*(7), 3368-3380.

APPENDICES

APPENDIX A

THE RESULTS OF THE PERFORMANCE OF MACHINE LEARNING MODELS

A.1 The Performance of Different Noise for the 9 datasets in case of Fixed splits

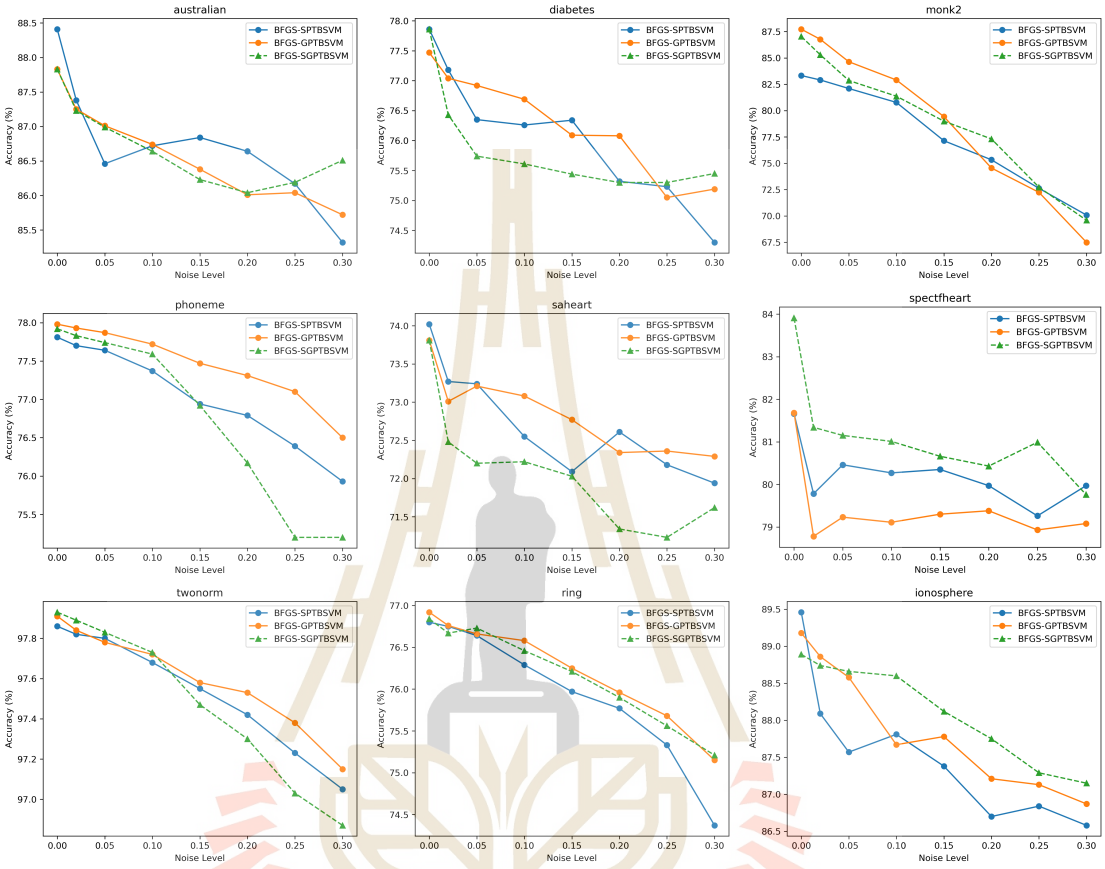


Figure A.1 Performance of different noise, linear model.

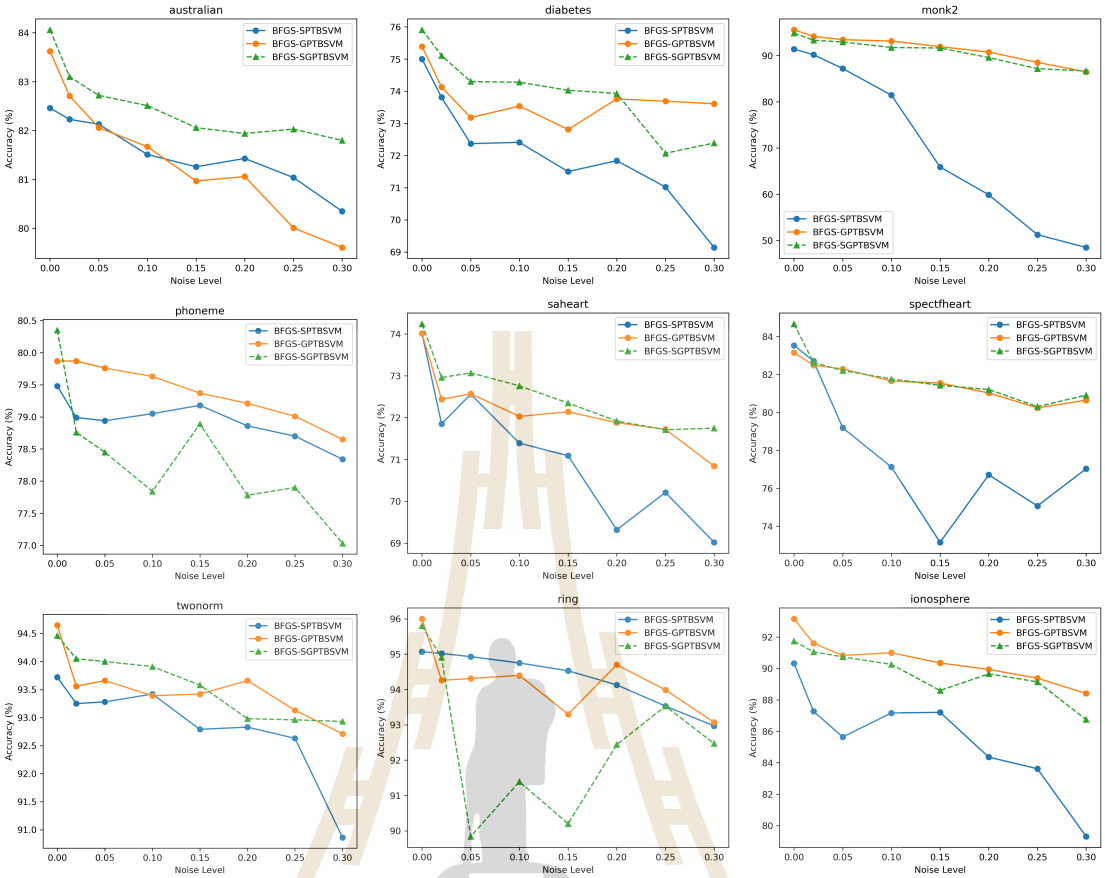


Figure A.2 Performance of different noise, RBFSampler.

A.2 The Performance of Different Noise for the 9 datasets in case of Variable splits

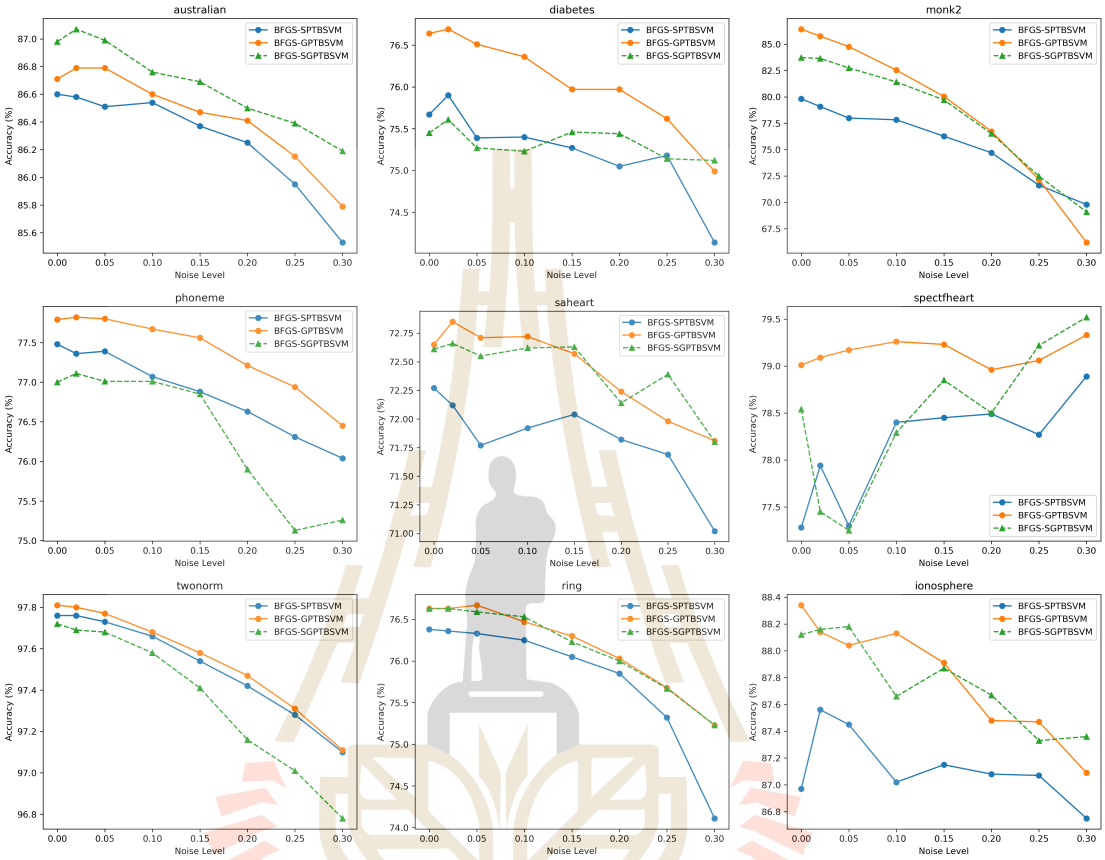


Figure A.3 Performance of different noise, linear model.

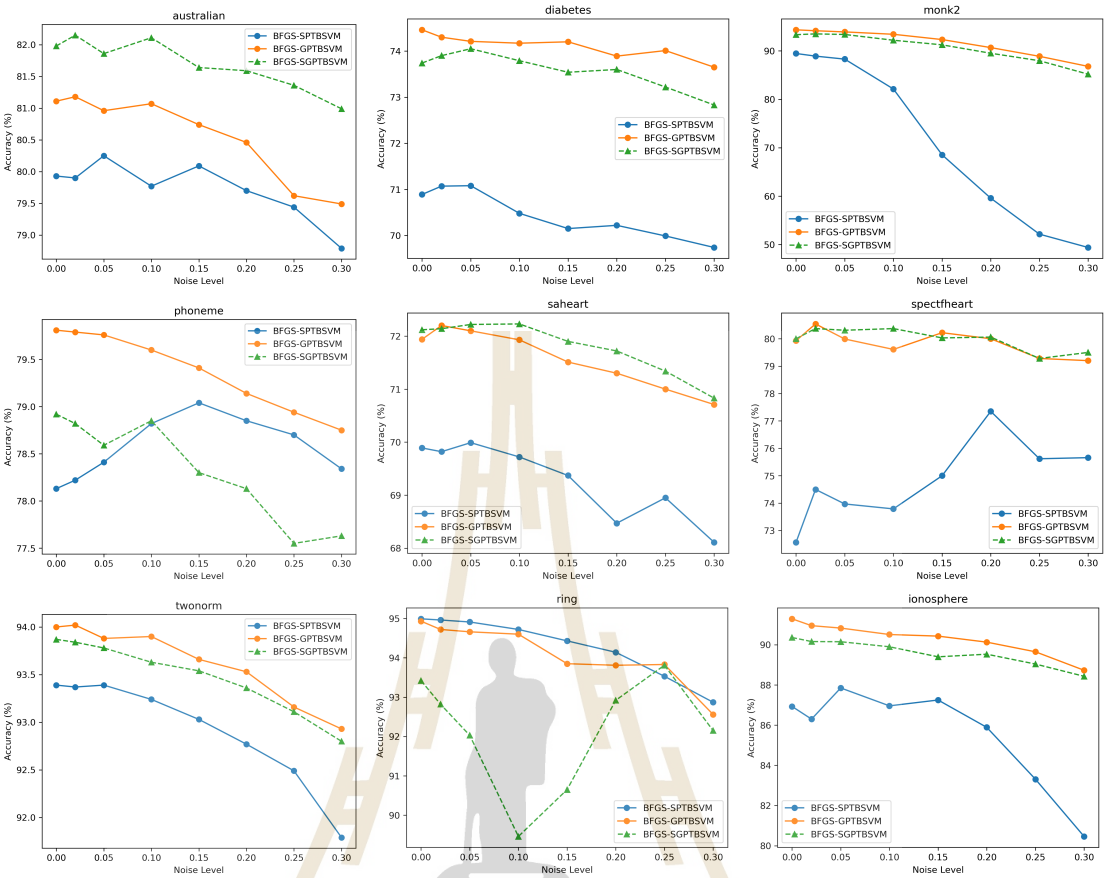


Figure A.4 Performance of different noise, RBFSampler.

CURRICULUM VITAE

NAME : Patcharapa Srichok **GENDER :** Female

EDUCATION BACKGROUND:

- Bachelor of Science (Mathematics), Suranaree University of Technology, Thailand, 2021

SCHOLARSHIP:

- Development and Promotion of Science and Technology Talents Project scholarship

CONFERENCE:

- Srichok, P., and Yimmuang, P. (2023). Levitin-Polyak well-posedness for generalized (η, g, φ)-mixed vector variational-type inequality., *The 11th Asian Conference on Fixed Point Theory and Optimization (ACFPTO2023)*, Department of Mathematics, Faculty of Science, Naresuan University, Phitsanulok, 2-5 August 2023.
- Srichok, P., Yimmuang, P., and Schulz, E. (2024) A novel twin bounded support vector machine with smooth generalized pinball loss., *The 4th International Conference and Workshop on Applied Nonlinear Analysis (ICWANA2024)*, Bangsaen Heritage Hotel, Bangsean, Chonburi, 6-8 August 2024, 79.

EXPERIENCE:

- Teaching assistant at Suranaree University of Technology, Analytical Calculus I, Analytical Calculus II, Calculus I, Calculus II, Calculus III.