Article

Capped Linex Metric Twin Support Vector Machine for Robust Classification

Yifan Wang, Guolin Yu * and Jun Ma

School of Mathematics and Information Science, North Minzu University, Yinchuan 750021, China
* Correspondence: yuguolin@nmu.edu.cn

Abstract: In this paper, a novel robust loss function is designed, namely, capped linear loss function $L_{a_{\varepsilon}}$. Simultaneously, we give some ideal and important properties of $L_{a_{\varepsilon}}$, such as boundedness, non-convexity and robustness. Furthermore, a new binary classification learning method is proposed via introducing $L_{a_{\varepsilon}}$, which is called the robust twin support vector machine (Linex-TSVM). Linex-TSVM can not only reduce the influence of outliers on Linex-SVM, but also improve the classification performance and robustness of Linex-SVM. Moreover, the effect of outliers on the model can be greatly reduced by introducing two regularization terms to realize the structural risk minimization principle. Finally, a simple and efficient iterative algorithm is designed to solve the non-convex optimization problem Linex-TSVM, and the time complexity of the algorithm is analyzed, which proves that the model satisfies the Bayes rule. Experimental results on multiple datasets demonstrate that the proposed Linex-TSVM can compete with the existing methods in terms of robustness and feasibility.

Keywords: capped linex loss function; robustness; classification; outliers

1. Introduction

Data collecting and reasonable processing are becoming increasingly crucial as modern computer technology advances. As an excellent machine learning tool, support vector machine (SVM) [1–4] has been widely used in financial forecast, bioinformatics, computer vision, image annotation, data mining and other fields in recent years. The main idea of SVM classification based on statistical learning theory and optimization theory is to construct a pair of parallel hyperplanes to maximize the minimum distance between two classes of samples. Generally speaking, the optimal hyperplane is realized by solving an optimization problem with inequality constraints. In order to avoid overfitting, scholars extend support vector machines to soft difference support vector machines (C-SVM) [5], introduce relaxation variables to relax constraints, and increase the penalty term of relaxation variables in the objective function. However, the loss function adopted by C-SVM is generally a hinge loss, which makes it very sensitive to noise. In the following research, C-SVM is extended to deal with the problem of function estimation, and a support vector interpretation of ridge regression [6] is proposed, which is different from the inequality constraints in C-SVM, as it uses equivalent constraints. Similarly, Suykens [7] considered equality constraints in the sense of least squares and proposed a least squares support vector machine (LSSVM). Unlike C-SVM, it does not use the non-support vector machine to optimize the classifier; LSSVM makes full use of the information of all data points and uses $L_2$ loss to punish both data points symmetrically. In order to further improve the performance of classification, researchers need to impose heavier penalties on samples that are misclassified. For this reason, in the literature [8], Ma et al. considered asymmetric linear exponential loss (LINEX) to be used to achieve this goal, and Linex-TSVM is proposed to study the binary classification problem. However, both SVM, C-SVM, LSSVM and Linex-TSVM have their own advantages, but they all have to solve a large-scale QP (Quadratic Programming problem), which requires a lot of time to learn and is not suitable for dealing with practical problems.
Because all the above models need to solve a big QPP, to further improve the computing speed, Jayadeva et al. [9] proposed a twin support vector machine (TSVM) for pattern classification based on the generalized eigenvalue approximation support vector machine (GEPSVM). Since TSVM solves two smaller QP problems instead of a single large QPP, it can theoretically learn four times faster than a standard SVM. The main goal of TSVM is to find two parallel hyperplanes, each of which is as close as possible to the corresponding class in the sample data, while being as far away from the other classes as possible. Therefore, TSVM is more suitable for the classification of large-scale data.

It is well known that distance metrics play a crucial role in many machine learning algorithms. Although the above algorithms demonstrate good performance in pattern classification, it is worth noting that most of them adopt the $L_2$-norm distance metric, whose squaring operation will exaggerate the impact of outliers on model performance. To effectively alleviate the impact of the $L_2$-norm distance metric on the robustness of the algorithm, the $L_1$-norm distance metric with the bounded derivative has received extensive attention and research in many fields of machine learning in recent years [10–12]. Recently, more and more researchers have paid attention to the capped $L_1$-norm. The capped $L_1$-norm can solve the deficiency of $L_1$-norm unboundedness. In particular, Wang et al. [13] proposed a new robust TSVM (CTSVM) by the applying the capped $L_1$-norm.

Inspired by the successful application of capped $L_1$-norm and linex loss function [14–17], meanwhile, the latest research shows that no scholar has extended the Linex loss function to twin support vector machines, therefore, a new robust twin support vector machine is established in this paper. The details and the main contributions of this work are as follows:

1. A novel robust loss function is designed, namely, capped linear loss function $L_{\alpha\epsilon}$.
2. A novel robust twin support vector machine, namely capped linex twin support vector machine (Linex-TSVM) is proposed.
3. An efficient iterative algorithm is designed to solve Linex-TSVM, which is not only easy to implement, but also theoretically guarantees the existence of a reasonable optimal solution. We analyze the computational complexity of the algorithm and prove that the model satisfies the Bayesian rule.
4. Extensive experiments conducted across multiple datasets demonstrates that the proposed Linex-TSVM is competitive with state-of-the-art methods in terms of robustness and feasibility. Therefore, the Linex-TSVM is feasible for practical applications.

The rest of this article is organized as follows. In Section 2, we briefly review Linex-SVM and TSVM. In Section 3, we describe in detail the proposed capped linex loss function and Linex-TSVM, and give the relevant theoretical analysis. After the experimental results on multiple data sets are presented in Section 4, we conclude this paper in Section 5.

2. Related Work

In this section, we are warranted to review Linex-SVM and TSVM.

2.1. Linex-SVM

Linex loss function is a typical asymmetric loss function, defined as:

$$L_{\text{linex}}(x) = e^{ax} - ax - 1$$

where $a \neq 0$ is a parameter. If $a < 0$, the left side of Linex loss is steeper than the right side, and the opposite is true when $a > 0$. The value of $|a|$ decides the symmetry of Linex loss function. This shows that the symbol of $a$ determines the shape of the function. When $a$ takes an appropriate value, it can be reduced to square loss. Linex loss function is not only asymmetric, but also convex and derivable; thus, it is widely used in statistics.

For the dichotomy problem in n-dimensional Euclidean space, the training set can be expressed as

$$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$$
where \( x_i \in \mathbb{R}^n \) is the feature vector of the data \( i \), and \( y_i \in \{-1,+1\} \) is the label of the data \( i \).

For the training set Equation (2), the Linex-support vector machine model can be written as a convex optimization problem with equation constraints in Equation (3) by introducing a linear loss function:

\[
\begin{align*}
\min_{\omega,b,\xi} & \quad \frac{1}{2} ||\omega||^2 + C \sum_{i=1}^{m} (\rho^\beta_i - a \xi_i - 1), \\
\text{s.t.} \quad & y_i (\omega^T x_i + b) - 1 = \xi_i, \quad i = 1, 2, \ldots, m.
\end{align*}
\]

where \( \xi = (\xi_1, \xi_2, \ldots, \xi_m) \) is a slack variable, \( C \) is a penalty parameter and \( a \) is a parameter of the Linex loss. Furthermore, we can use the Nesterov accelerated gradient (NAG) method to obtain the optimal solution \( (\omega_1, b_1) \) and construct the decision function \( f(x) = \text{sgn}(\omega_1^T x + b_1) \).

### 2.2. TSVM

The support vector machine is not suitable for dealing with large-scale data, to improve the practical application of the model and further shorten the learning time. Jayadeva et al. proposed a twin support vector machine (TSVM) for pattern classification based on the generalized eigenvalue approximation support vector machine (GEPSVM). The details are as follows:

Considering \( n \) dimensional Euclidean space \( \mathbb{R}^n \) the binary classification problem, the training set is \( T = \{x_i, y_i | i = 1, 2, \ldots, m\} \), \( x_i \in \mathbb{R}^n \), where \( y_i \in \{-1, 1\} \). \( A \in \mathbb{R}^{m \times n} \) represents all positive samples; \( B \in \mathbb{R}^{m_2 \times n} \) represents all negative samples. TSVM identifies two non-parallel hyper planes in the feature space:

\[
\begin{align*}
f_1(x) = \omega_1^T x + b_1 &= 0, \\
f_2(x) = \omega_2^T x + b_2 &= 0.
\end{align*}
\]

where \( \omega_1, \omega_2 \in \mathbb{R}^n, b_1, b_2 \in \mathbb{R} \).

The TSVM classifier is obtained by solving the following pair of QPPS:

\[
\begin{align*}
\min_{\omega_1,b_1} & \quad \frac{1}{2} ||A\omega_1 + e_1 b_1||^2_2 + C_1 \xi_1^T, \\
\text{s.t.} \quad & -(B\omega_1 + e_2 b_1) + \xi_1 \geq e_2, \quad \xi_1 \geq 0.
\end{align*}
\]

\[
\begin{align*}
\min_{\omega_2,b_2} & \quad \frac{1}{2} ||B\omega_2 + e_2 b_2||^2_2 + C_2 \xi_2^T, \\
\text{s.t.} \quad & (A\omega_2 + e_1 b_2) + \xi_2 \geq e_1, \quad \xi_2 \geq 0.
\end{align*}
\]

where \( C_1 \geq 0, C_2 \geq 0 \) represent regularization parameters, \( e_1, e_2 \) are all unit vectors, \( \xi_1, \xi_2 \) are the slack vectors.

Then, the dual problem of TSVM is obtained by dual theory:

\[
\begin{align*}
\min_{\alpha} & \quad -\frac{1}{2} \alpha^T G (H^T H + \lambda I)^{-1} G^T \alpha + e_2^T \alpha, \\
\text{s.t.} \quad & 0 \leq \alpha \leq C_1 e_2.
\end{align*}
\]

\[
\begin{align*}
\min_{\beta} & \quad -\frac{1}{2} \beta^T H (G^T G + \lambda I)^{-1} H^T \beta + e_1^T \beta, \\
\text{s.t.} \quad & 0 \leq \beta \leq C_2 e_1.
\end{align*}
\]

\( \alpha \in \mathbb{R}^{m_2} \) and \( \beta \in \mathbb{R}^{m_1} \) are lagrange multipliers. At the same time, matrix \( G \) and \( H \) respectively defined as: \( G = [B e_2] \) and \( H = [A e_1] \).
Furthermore, by introducing the kernel method, TSVM can be extended to nonlinear space, and it can be decided whether the sample data \( x \) belongs to positive class or negative class according to the shortest distance between the sample data \( x \) and two non-parallel planes. The decision function is
\[
f(x) = \arg\min_{k=1,2} \left| x\omega_k + b_k \right| / \| \omega_k \|.
\] (10)

3. Main Contribution

3.1. Capped Linex Loss Function

In this section, in order to minimize the influence of abnormal values on the classification results of the model, we propose a novel robust loss function, that is, the capped linex loss function. The details are as follows:

Definition 1. The capped linex loss function is defined as
\[
L_{\text{cap}}(x) = \min \left( \sum_i a^x_i - ax_i - 1, \epsilon \right)
\] (11)

where \( a \neq 0 \) is a parameter, when \( a < 0 \), the left side of the loss function is steeper than the right side; when \( a > 0 \), the right side of the loss function is steeper than the left side; see Figure 1. \( \epsilon > 0 \) is a thresholding parameter; \( x_i \) is the component of \( x \).

Figure 1 shows the comparison between the capped linex loss function and the linex loss function. Obviously, we can observe that the improved linear loss has an upper bound, and when the error tends to be consistent, even if there are outliers, the loss will not increase to a certain extent, which improves the robustness of the model.

![Figure 1](image)

(a) \( a = 1, \epsilon = 0.5 \)  
(b) \( a = -0.5, \epsilon = 0.25 \)

Figure 1. Linex loss capped linex loss.

3.2. Capped Linex Twin Support Vector Machine

Linex-SVM model still needs to be improved: linex loss is an unbounded function, and the loss tends to be consistent with the increase in error. However, in practical applications, datasets are often accompanied by noise, and the unboundedness of the linex loss function will affect the overall performance of the model. In other words, Linex-SVM is a relatively weak method in dealing with training sets with outliers. In addition, almost all the instances in the Linex-SVM contribute to the final optimal hyperplane, which will greatly reduce the training speed.

In order to improve the classification performance of Linex-SVM, we first improve Linex loss to capped linex loss and introduce regularization term to enhance robustness. Secondly, we generalize Linex-SVM to twin support vector machine, and transform a large QPP into two small QP problems to improve the training speed. Based on the above
two points, a new twin support vector machine model, named the capped linex twin support vector machine (Linex-TSVM) is obtained:

$$\min_{\omega, c} \sum_{i=1}^{m} \min (\|\omega x_i + b \|_1, \eta_i) + C \sum_{i=1}^{n} \min (e^{\alpha_i^e} - a_i^e - 1, \epsilon_2) + \frac{C_3}{2} (||\omega||_2^2 + b_1^2),$$

s.t. \(- (b \omega + e_2 b_1) + \xi \geq \epsilon_2, \) \( \xi \in \mathbb{R}^m \)

$$\min_{\omega, c} \sum_{i=1}^{m} \min (\|\omega x_i + b_2 \|_1, \eta_3) + C \sum_{i=1}^{n} \min (e^{\alpha_i^e} - \eta_i - 1, \epsilon_4) + \frac{C_4}{2} (||\omega_2||_2^2 + b_2^2),$$

s.t. \((A \omega_2 + e_1 b_2) + \eta \geq \epsilon_1, \) \( \eta \in \mathbb{R}^m \)

where \( C_1, C_2, C_3, C_4 \geq 0, \) \( e_1 \in \mathbb{R}^{m_1} \) and \( e_2 \in \mathbb{R}^{m_2} \) are the unit vectors. \( \xi \) and \( \eta \) are slack vectors.

In addition, we also notice that when using the traditional convex optimization method, it is difficult to solve the problems Equations (12) and (13) simply and quickly. Here, in order to simplify the original problem to an approximate problem that is easier to solve, we use the re-weighted trick [12,18–20], the most important of which is the formula \( ||x||_1 = \frac{x^T F x}{x^T F x} \). Take Equation (12) as an example, for the distance measurement items, when the \( F = \frac{1}{x^T F x} \) holds, then \( ||x||_1 = x^T F x \). For the loss function terms, when \( e^{\alpha_i^e} - a_i^e - 1 \leq \epsilon_2 \), there are \( \sum_{i=1}^{m_2} \min (e^{\alpha_i^e} - a_i^e - 1, \epsilon_2) = \sum_{i=1}^{m_2} (e^{\alpha_i^e} - a_i^e - 1) \). Further, in order to simplify function \( e^{\alpha_i^e} - a_i^e - 1 \) into an easy-to-solve \( \xi^T Q \xi \), we define \( Q \) as diagonal matrices with \( i \)-th diagonal element as:

$$q_i = \begin{cases} \frac{e^{\alpha_i^e} - a_i^e - 1}{\xi_i^2}, & \text{if } e^{\alpha_i^e} - a_i^e - 1 \leq \epsilon_2 \\ 0, & \text{otherwise} \end{cases}$$

(14)

similarly,

$$u_i = \begin{cases} \frac{e^{\alpha_i^e} - \eta_i - 1}{\eta_i^2}, & \text{if } e^{\alpha_i^e} - \eta_i - 1 \leq \epsilon_4 \\ 0, & \text{otherwise} \end{cases}$$

(15)

Based on the above discussion and calculation, we can obtain the optimization problem Equations (16) and (17), as follows:

$$\min_{\omega, c} (A \omega_1 + e_1 b_1)^T F (A \omega_1 + e_1 b_1) + \frac{1}{2} C_1 \xi^T Q \xi + \frac{C_3}{2} (||\omega||_2^2 + b_1^2),$$

s.t. \(- (b \omega + e_2 b_1) + \xi \geq \epsilon_2, \) \( \xi \in \mathbb{R}^m \)

(16)

$$\min_{\omega, c} (B \omega_2 + e_2 b_2)^T K (B \omega_2 + e_2 b_2) + \frac{1}{2} C_2 \eta^T U \eta + \frac{C_4}{2} (||\omega_2||_2^2 + b_2^2),$$

s.t. \((A \omega_2 + e_1 b_2) + \eta \geq \epsilon_1, \) \( \eta \in \mathbb{R}^m \)

(17)

where \( e_1 \in \mathbb{R}^{m_1} \) and \( e_2 \in \mathbb{R}^{m_2} \) are the unit vectors, \( F \) and \( K \) are also two diagonal matrices:

$$f_i = \begin{cases} \frac{1}{|\omega_1 x_i + b_1|}, & \text{if } |\omega_1 x_i + b_1| \leq \epsilon_1 \\ 0, & \text{otherwise} \end{cases}$$

(18)

$$k_i = \begin{cases} \frac{1}{|\omega_2 x_i + b_2|}, & \text{if } |\omega_2 x_i + b_2| \leq \epsilon_3 \\ 0, & \text{otherwise} \end{cases}$$

(19)
Remark 1. What is more detailed is that in the objective functions Equations (16) and (17), we use diagonal matrices \( F, Q \) and \( K, U \), respectively, to reduce the influence of outliers and abnormal noise on the model. Specifically, if the points in the same class are far away from the hyperplane, they can be treated as noise and removed. In addition, the model mainly sets the elements in the diagonal matrix according to the distance from the data point \( x_i \) to the hyperplane. For \( F \), if \( f_i \) is greater than \( \varepsilon_1 \), the corresponding \( f_i \) is set to a smaller value (Smallval), which is almost equivalent to 0. Where ‘Smallval’ is a small constant, which will be set to \( 10^{-8} \) in the experiment.

The corresponding Lagrange function of the above optimization problem Equation (16) can be written as:

\[
L(\omega_1, b_1, \xi_1, \alpha) = \frac{1}{2} (A\omega_1 + e_1 b_1)^T F (A\omega_1 + e_1 b_1) + \frac{1}{2} C_1 \xi^T Q \xi + \frac{C_3}{2} (\|\omega_1\|^2 + b_1^2) - \alpha^T (-B\omega_1 + e_2 b_1) + \xi - e_2
\]

(20)

where \( \alpha \) is a Lagrange multiplier, derive the Lagrange function about \( \omega_1 \) and \( b_1 \), obtain the following Karush–Kuhn–Tucker Conditions.

\[
\begin{align*}
\frac{\partial L}{\partial \omega_1} &= A^T F (A\omega_1 + e_1 b_1) + B^T \alpha + C_3 \omega_1 = 0, \quad (i) \\
\frac{\partial L}{\partial b_1} &= e_1^T F (A\omega_1 + e_1 b_1) + e_2^T \alpha + C_3 b_1 = 0, \quad (ii) \\
\frac{\partial L}{\partial \xi_1} &= C_1 Q \xi \alpha = 0, \quad (iii) \\
\alpha^T (B\omega_1 + e_2 b_1 - \xi + e_2) &= 0, \quad (iv) \\
\alpha &\geq 0. \quad (v)
\end{align*}
\]

Through KKT condition, the dual problem of Equation (12) is as follows:

\[
\min_{\alpha} \frac{1}{2} \alpha^T (E(H^T F H + C_3 I)^{-1} E^T + \frac{1}{C_1} Q^{-1}) \alpha - e_2^T \alpha,
\]

s.t. \( 0 \leq \alpha \leq C_1 e_2 \).

(22)

Similarly, the dual problem of Equation (13) is:

\[
\min_{\beta} \frac{1}{2} \beta^T (H(E^T K E + C_4 I)^{-1} H^T + \frac{1}{C_2} U^{-1}) \beta - e_1^T \beta,
\]

s.t. \( 0 \leq \beta \leq C_2 e_1 \).

(23)

where \( \beta \) is a Lagrange multiplier, and

\[
H = [A \quad e_1], \quad E = [B \quad e_2].
\]

(24)

Thus, we get the vector \( Z_1 \) and \( Z_2 \), and gain the new data point \( x \in \mathbb{R}^n \) to a positive or negative category.

Based on the above discussion, our algorithm will be presented in Algorithm 1.
Algorithm 1 Iterative algorithm to solve Linex-CTSVM

**Input:** Training data $A \in \mathbb{R}^{m_1 \times n}$ and $B \in \mathbb{R}^{m_2 \times n}$; Parameters $C_i(i = 1, 2, 3, 4)$ and $\varepsilon_i(i = 1, 2, 3, 4)$. Establish matrixs $H = [A \varepsilon_1]$, $E = [B \varepsilon_2]$. Initialize $F \in \mathbb{R}^{m_1 \times m_2}$ and $K \in \mathbb{R}^{m_2 \times m_2}$. Let $k = 0$

Iterative

$$\left([\omega_1, b_1]^T\right)^{k+1} = -(H^T F H + C_3 I)^{-1} E^T a.$$ $$\left([\omega_2, b_2]^T\right)^{k+1} = (E^T K E + C_4 I)^{-1} H^T b.$$ 

Update matrix separately $Q, U, F, K$ by Equations (14), (15), (18) and (19)

Let $k = k + 1$ and go to step 2, until convergence stops.

**Output:** Optimal solution $[\omega_1, b_1]^T$ and $[\omega_2, b_2]^T$.

3.3. Bayes Rule

We want to prove that the model proposed in this paper can satisfy the Bayes rule, assuming that the sample $(x_i, y_i)$ are independent of the same probability $\phi$, and the probability $\phi$ is defined on $X \times Y$, where $X \in \mathbb{R}^n, Y = -1, 1$. Further, we assume that the conditional distribution $\phi(y|x)$ is a binomial distribution, including $\phi(-1|x)$ and $\phi(1|x)$. As we all know, the ultimate goal of the classification problem is to obtain a classifier $C : X \rightarrow Y$ with small error. Bayesian classifier [8] is defined as the classifier with the lowest probability of classification error among all kinds of classifiers.

$$f_C(x) = \begin{cases} 1, & \text{if } \phi(y = 1|x) \geq \phi(y = -1|x), \\ -1, & \text{if } \phi(y = 1|x) < \phi(y = -1|x). \end{cases} \quad (25)$$

For any loss function $L$, the expected risk of the classifier $f : X \rightarrow R$ can be defined as

$$R_{L, \phi} = \int_{X \times Y} L(1 - yf(x))d\phi. \quad (26)$$

Next, by minimizing the expected risk of all measurable classification functions, we can obtain

$$f_{L, \phi} = \arg \min_{\tau \in \mathbb{R}} \int L(1 - y(x)\tau)d\phi(y|x), \quad \forall x \in X. \quad (27)$$

Based on the above important definition of Bayes rule, we obtain Theorem 1 to prove that Bayes rule holds for capped linex loss function. The details of the proof are as follows.

**Theorem 1.** Function $f_{L, \phi}$, which minimizes the expected risk on all measurable functions $f : X \rightarrow Y$, making the result equivalent to that of a Bayes classifier, that is $f_{L, \phi}(x) = f_C(x), \forall x \in X$.

**Proof.** By the properties of capped linex loss function, when $e^{ax_i} - ax_i - 1 < \varepsilon$, we can obtain

$$L_{ae}(x) = e^{ax} - ax - 1. \quad (28)$$

So, there are

$$\int_Y L_{ae}(1 - y(x)\tau)d\phi(y|x) = L_{ae}(1 - \tau)\phi(y = 1|x) + L_{ae}(1 + \tau)\phi(y = -1|x) \quad (29)$$

$$= e^{a(1-\tau)} - a(1-\tau) - 1\phi(y = 1|x) + e^{a(1+\tau)} - a(1+\tau) - 1\phi(y = -1|x)$$

By Equation (29), when $\phi(y = 1|x) \geq \phi(y = -1|x)$ and $\phi(y = 1|x) < \phi(y = -1|x)$, obtain the minimum value at $\tau = -1$ and $\tau = 1$, respectively, and when $\phi(y = 1|x) = \phi(y = -1|x)$,
we obtain the minimum value at $\tau = -1$ or $\tau = 1$. Therefore, when $e^{a x_i} - a x_i - 1 < \epsilon$, the capped linex loss function can measure the minimum expected risk of $f_{L_{ax},\phi}(x)$. To sum up

$$
f_{L_{ax},\phi}(x) = \begin{cases} 
1, & \text{if } \phi(y = 1|x) \geq \phi(y = -1|x), \\
-1, & \text{if } \phi(y = 1|x) < \phi(y = -1|x).
\end{cases}
$$

(30)

### 3.4. Computational Complexity Analysis

This part mainly analyzes the computational complexity of Algorithm 1. As we all know, the computational complexity includes the number of iterations and the computational cost of iterations. The computational complexity of Algorithm 1 after one iteration is divided into two parts: (1) the time complexity of solving QPP is not more than $\frac{m^3}{4}$, and the inverse of matrix is not greater than $(n + 1)^3(n + 1)$. Therefore, the total time complexity of solving Linex-TSVM is about $O(t \cdot (\frac{m^3}{4} (n + 1)^3))$, where $t$ is the number of iterations, and the experimental results of this paper demonstrate that $t = 50$ meets the expectation. Under the condition of universality, the number of iterations of each algorithm is much less than the number of samples. Similarly, Linex-TSVM has cubic time complexity in the number of samples.

### 4. Experimental Results and Discussions

In this section, we first set the experimental parameters in Section 4.1, and in sections Sections 4.2 and 4.3, we give in detail the experimental results of the model Linex-TSVM with or without noise. Finally, we present some results on the data set in Section 4.4 to prove the convergence of the objective function.

#### 4.1. Experimental Setup

##### 4.1.1. Evaluation Criteria

In order to evaluate the classification performance of our proposed truncated linear loss support vector machine more accurately, we compare it with other mature methods, including SVM, LSSVM, C-SVM, Linex-TSVM, and TBSVM. For these five support vector machines and Linex-TSVM, the iterative process is stopped when the difference between the target values of the two iterations is less than 0.001 and the number of iterations is more than 50. At the same time, in order to measure the performance of all algorithms, the traditional precision index (ACC) is used to measure the performance of these algorithms, which is defined as follows:

$$
ACC = \frac{TP + TN}{TP + FN + TN + FP}
$$

(31)

Among them, TP and TN represent correct positive samples and negative samples, respectively. FN and FP represent wrong positive samples and negative samples, respectively. In order to make a more accurate comparison, we use the quadratic programming (QP) toolbox of matlab to solve the QP problem in related algorithms. The experimental environment consists of a Windows 10 machine and Intel i7 Processor (3.70 GHz) with 8 GB of RAM.

##### 4.1.2. Parameters Selection

For the learning algorithm, its performance is very sensitive to the parameters involved, so it is necessary to record the parameters of each algorithm in detail and list them as follows.

- SVM and LSSVM: the kernel parameter $\sigma$.
- C-SVM: the regularization parameter $c$, the kernel parameter $\sigma$. 

NPSVM and TBSVM: the regularization parameters $c_1, c_2, c_3$ and $c_4$, the kernel parameter $\sigma$. Linex-SVM: the regularization parameter $c$, a parameter $a$ of the linex loss, the kernel parameter $\sigma$. Linex-TSVM: the regularization parameter $c_1, c_2, c_3, c_4$, a parameter $a$ of the linex loss, the parameters $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4$ and the kernel parameter $\sigma$.

where $\epsilon_1 = \epsilon_2 = \epsilon_3 = \epsilon_4 = 10^{-5}$; $c_1, c_2, c_3, c_4 : \{10^i | -5, -4, ..., 4, 5\}$; $\sigma, \epsilon : \{10^i | i = -4, -3, ..., 3, 4\}$. The experimental parameters are selected by ten cross-validation methods, and the test accuracy is the average of 10 clusters of results in each dataset.

4.1.3. Description of the Datasets

To verify the effectiveness of Linex-TSVM, we conduct numerical simulations on different datasets, including seven benchmark datasets from the UCI machine learning repository and two artificial datasets. The datasets are described as follows:

Artificial datasets: In the artificial dataset (a) and (b), there are 50 positive samples and 50 negative samples, represented by ‘+’, ‘□’ and ‘○’, respectively, as shown in Figure 2. Because the outliers will have a certain impact on the classification performance, it is also the standard to measure the stability of the algorithm. Therefore, we introduce four outliers in the artificial dataset to evaluate its robustness, two of which belong to class +1 and two belong to class −1.

UCI datasets: Australian, Spect, Pima, German, Vote, CMC, Sonar, Spect and Large dataset(codrna). Details of the eight UCI datasets are given in Table 1. These UCI datasets are used to test the performance of our algorithms and related algorithms.

Table 1. Characteristics of UCI datasets.

| Datasets   | Samples | Attributes | Datasets   | Samples | Attributes |
|------------|---------|------------|------------|---------|------------|
| Australian | 690     | 14         | Spect      | 267     | 45         |
| Pima       | 768     | 8          | German     | 1000    | 24         |
| Sonar      | 198     | 60         | Vote       | 432     | 16         |
| CMC        | 1140    | 9          | codrna     | 59,535  | 9          |

We divide all the data sets into ten subsets, including nine training sets and one test set, that is, 10-fold cross-validation, so that the process is repeated ten times, and the average value of the final result is taken as the criterion to measure the performance of the model. At the same time, we normalize the eight participating data sets, which can avoid errors caused by different orders of magnitude and units, keeping the result within $[0, 1]$.

![Figure 2](image-url)  
(a) Artificial dataset1  
(b) Artificial dataset2

4.2. Experimental Results on the Employed Datasets without Outliers

Eight UCI datasets are selected and the running results are compared with the other six algorithms to verify the better classification performance of the proposed algorithm.
All experimental results presented in Table 2 are based on optimal parameters. Here, "Time(s)" denotes the average runtime in seconds taken by each algorithm according to the optimal parameters, "ACC ± S" denotes the average classification accuracy plus or minus standard deviations.

Table 2. Experimental results on UCI datasets without noise.

| Datasets | SVM | LSSVM | C-SVM | NPSVM | Linex-SVM | TBSVM | Linex-TSVM |
|----------|-----|-------|-------|-------|-----------|-------|------------|
|          | ACC ± S (%) | ACC ± S (%) | ACC ± S (%) | ACC ± S (%) | ACC ± S (%) | ACC ± S (%) | ACC ± S (%) |
|          | Times (s) | Times (s) | Times (s) | Times (s) | Times (s) | Times (s) | Times (s) |
| Australian | 82.0± 0.63 | 82.7± 1.52 | 83.2± 3.39 | 83.9± 2.53 | 83.7± 1.63 | 84.0± 1.14 | 84.1± 0.42 |
|          | 1.905 | 1.108 | 1.241 | 2.037 | 2.644 | 2.358 | 4.535 |
| Vote | 89.5± 0.73 | 90.1± 1.17 | 93.6± 1.53 | 94.0± 1.87 | 94.3± 0.97 | 95.0± 1.34 | 95.2± 1.05 |
|          | 0.943 | 0.793 | 1.832 | 1.834 | 1.846 | 1.115 | 2.062 |
| German | 70.6± 1.74 | 71.3± 0.21 | 72.1± 3.22 | 73.3± 1.26 | 73.2± 2.86 | 75.6± 0.87 | 75.8± 0.21 |
|          | 2.286 | 1.719 | 2.303 | 3.421 | 4.672 | 3.719 | 9.695 |
| Spect | 78.0± 1.62 | 77.9± 1.53 | 79.4± 5.25 | 80.1± 0.41 | 80.5± 2.34 | 80.7± 1.09 | 82.1± 1.01 |
|          | 0.639 | 0.945 | 1.347 | 2.014 | 1.739 | 1.235 | 1.438 |
| CMC | 55.1± 3.62 | 56.4± 2.79 | 57.5± 1.52 | 56.3± 3.15 | 57.1± 0.76 | 61.1± 5.71 | 60.97± 0.62 |
|          | 1.441 | 1.367 | 2.980 | 3.286 | 5.862 | 3.744 | 8.328 |
| Pima | 73.4± 0.72 | 72.7± 1.24 | 74.2± 1.03 | 73.4± 1.64 | 74.3± 1.69 | 75.8± 0.66 | 75.7± 1.31 |
|          | 1.117 | 1.036 | 1.125 | 2.185 | 2.092 | 1.626 | 5.032 |
| Sonar | 63.5± 2.54 | 63.6± 2.34 | 65.3± 0.69 | 64.1± 0.68 | 64.3± 1.42 | 65.6± 2.16 | 66.5± 0.68 |
|          | 0.350 | 0.137 | 0.659 | 0.517 | 0.813 | 0.534 | 0.913 |
| codrna | 82.1± 1.73 | 80.0± 2.61 | 83.6± 0.11 | 85.4± 2.24 | 86.1± 0.62 | 85.4± 3.11 | 88.7± 3.40 |
|          | 50.947 | 43.400 | 49.203 | 51.229 | 70.562 | 66.548 | 59.914 |

Intuitively, it can be observed from Table 2 that the classification performance of the twin support vector machine based on capped linear loss function proposed in this paper is better than that of the other six models. Except for CMC data sets, Linex-TSVM has better results on other data sets. At the same time, we also observe that the computing time of this model is not dominant, which is because the model is more complex. The time of LSSVM algorithm for solving a system of linear equations is shorter, and compared with SVM, it shortens the time while retaining accuracy, which is in line with the relevant theory. It is worth mentioning that the result of Linex-SVM is still good, which shows that the introduction of linear loss function is meaningful.

Through the detailed analysis of the above experimental results, we can obtain an objective and reasonable conclusion: the use of capped linear loss function on the basis of TBSVM can improve the classification performance, and the introduction of L1-norm distance metric can also enhance the robustness of the model; thus, our model is an effective supervision algorithm without the influence of outliers.

4.3. Experimental Results on the Employed Datasets with Outliers

4.3.1. Experimental Results on Artificial Dataset with Outliers

It is well known that outliers tend to have a certain impact on classification performance, which is also a measure of the stability of the algorithm. Therefore, we introduce outliers in artificial datasets (a) and (b), respectively, and Figure 2 is displayed visually. In order to further verify the robustness of the capped linear loss function, we show the classification accuracy of this algorithm on artificial data sets (a) and (b) in Figure 3, and compare the other five algorithms.
From Figure 3, we observe that the proposed Linex-TSVM has higher accuracy when considering outliers; on artificial datasets (a) and (b), the classification accuracy of Linex-TSVM is 68.06% and 91.97%, respectively, which is better than the other five algorithms, can deal with outliers well, and has stronger robustness and better classification ability.

In summary, the capped L_1-norm is robust to different types of outliers in the literature [21–25]. It can overcome the residual error of outliers in the experiment, and can help the model to eliminate the influence of outliers. In particular, the truncated linear loss function in this model can increase the punishment for outliers. In a word, Linex-TSVM can effectively improve the robustness of TBSVM.

4.3.2. Experimental Results on UCI Dataset with Outliers

In order to verify that this model is also suitable for large-scale data with outliers, we add 10% and 25% noises to the eight UCI data sets, respectively. The reason why the algorithm is introduced into the model is that in practical application, there are various kinds of data and there must be different degrees of noise. In order to verify that the model is suitable for data sets of different fields and different sizes to a certain extent, it is necessary to introduce different noise to compare the models. At the same time, we find that after adding noise, the accuracy will fluctuate to a certain extent, but the overall trend shows a slow decline, which shows that when the noise is relatively large, it will have a certain impact on the model, but the model in this paper is more stable. The results, such as Tables 3 and 4, show that after the introduction of outliers, the seven algorithms all have varying degrees of accuracy fluctuations, but show a downward trend as a whole, and the classification accuracy of Linex-TSVM is almost better than other algorithms. This shows that the model proposed in this paper has stronger robustness.

Specifically, in Tables 3 and 4, Linex-TSVM has the best accuracy in seven of the eight data sets, while the least squares support vector machine model has the shortest computing time under different noises. It is worth noting that compared with SVM, LSSVM, NPSVM, C-SVM, Linex-SVM, due to the use of capped linear loss function, the penalty for outliers is increased, so it has better classification accuracy. Linex-TSVM is better than Linex-SVM and TBSVM.
Furthermore, in order to more comprehensively analyze the robustness of the algorithm under different noises, we have carried out more experiments on Australian, Spect, Pima, German, Vote, CMC, Sonar, Codrna and Spect, and we use different noises to test the performance of the six algorithms. For an original dataset $X$, we changed it with $X + \lambda X$, where $\lambda = \frac{q}{||X||^2}$ and $q$ is a noise factor. Here, $X$ is the noise matrix whose elements are i.i.d. standard Gaussian variables. The value is $q \in \{0.1, 0.2, 0.3, 0.4\}$. Through Figure 4, we can observe that under different noise factors, Linex-TSVM shows better classification accuracy and stability, while the other six models are relatively more volatile.

Next, we introduce the box line diagram to verify that the model is better from another point of view. In Figure 5, we select six datasets to analyze the height of the box reflects the fluctuation of the data to a certain extent, that is, it represents the fluctuation of classification accuracy. The upper and lower edges represent the maximum and minimum values of the group of data, and the points outside the box can be understood as "outliers" in the data, so

| Datasets   | SVM   | LSSVM | C-SVM | NPSVM | Linex-SVM | TBSVM | Linex-TSVM |
|------------|-------|-------|-------|-------|-----------|-------|------------|
|            | ACC ± S (%) | ACC ± S (%) | ACC ± S (%) | ACC ± S (%) | ACC ± S (%) | ACC ± S (%) | ACC ± S (%) |
| Australian | 81.61 ± 0.82 | 81.51 ± 0.83 | 82.44 ± 2.60 | 81.27 ± 1.39 | 83.11 ± 1.45 | 83.61 ± 1.07 | 83.83 ± 0.31 |
|            | 1.866 | 1.154 | 1.346 | 1.302 | 2.842 | 2.775 | 4.279 |
| Vote       | 88.77 ± 0.63 | 89.25 ± 1.38 | 93.07 ± 1.63 | 93.60 ± 2.95 | 92.35 ± 1.68 | 93.27 ± 2.17 | 94.29 ± 1.01 |
|            | 1.193 | 0.810 | 1.893 | 1.027 | 1.088 | 1.607 | 2.034 |
| German     | 68.22 ± 0.89 | 69.15 ± 2.41 | 70.84 ± 2.41 | 74.30 ± 2.26 | 71.66 ± 2.50 | 73.14 ± 1.21 | 73.91 ± 0.82 |
|            | 2.876 | 2.274 | 2.507 | 2.460 | 4.339 | 3.105 | 9.148 |
| Spect      | 77.29 ± 1.87 | 77.91 ± 0.88 | 78.48 ± 3.57 | 78.51 ± 3.11 | 78.75 ± 1.35 | 79.06 ± 4.62 | 81.73 ± 0.94 |
|            | 1.519 | 0.988 | 1.830 | 1.616 | 1.613 | 1.012 | 1.910 |
| CMC        | 54.74 ± 3.30 | 53.20 ± 2.79 | 54.59 ± 2.24 | 58.03 ± 0.87 | 57.74 ± 1.17 | 59.91 ± 0.13 | 59.91 ± 0.75 |
|            | 1.441 | 1.367 | 2.980 | 4.339 | 5.862 | 3.744 | 8.215 |
| Pima       | 70.42 ± 0.72 | 71.79 ± 1.24 | 72.21 ± 1.03 | 71.82 ± 0.14 | 72.70 ± 1.69 | 73.92 ± 0.66 | 73.53 ± 1.31 |
|            | 1.535 | 1.239 | 1.599 | 3.700 | 2.227 | 1.803 | 5.118 |
| Sonar      | 62.93 ± 2.54 | 62.71 ± 2.34 | 63.06 ± 1.19 | 63.14 ± 3.01 | 63.27 ± 1.42 | 63.85 ± 1.13 | 64.50 ± 4.95 |
|            | 1.184 | 0.917 | 0.726 | 1.715 | 1.476 | 1.244 | 1.619 |
| Codrna     | 80.67 ± 2.67 | 81.44 ± 3.61 | 81.61 ± 1.02 | 82.35 ± 1.11 | 84.29 ± 0.90 | 86.77 ± 1.30 | 87.54 ± 0.34 |
|            | 54.302 | 46.169 | 50.495 | 62.376 | 55.482 | 61.517 | 58.455 |

Table 3. Experimental results on UCI datasets with 10% noise.
we can directly observe that the classification accuracy of Linex-TSVM is higher than that of other models.

To sum up, the capped linear loss function twin support vector machine proposed in this paper is superior to the other six algorithms in terms of classification accuracy and robustness, indicating that Linex-TSVM is a robust learning algorithm for large-scale data classification with noise.

Figure 4. Accuracies of seven algorithms via different noises factors.
4.4. Analysis for the Convergence

In this section, we show the convergence curve of the proposed algorithm on four datasets to directly verify that the convergence speed of the proposed algorithm can achieve the desired speed. The result is Figure 6, where the horizontal axis represents the number of iterations and the vertical axis represents the value of the objective function. We set: when the difference between the target values of two consecutive iterations is less than 0.001 and the number of iterations is less than 50, the iterative process stops.

The result of Figure 6 shows that the value of the objective function of Linex-TSVM decreases monotonously with the increase in the number of iterations, and the algorithm can converge quickly in about 5 iterations, that is, it converges within a limited number of iterations, and we obtain satisfactory results, which is consistent with the previous theoretical analysis.
4.5. Statistical Analysis

In this section, the statistical analysis method-Friedman test is used to compare the differences among the six algorithms involved. In this paper, the Friedman test is a statistical test of the homogeneity of multiple (related) samples, which makes full use of all the information in the original data and has many advantages. It is worth noting that the zero hypothesis means that all algorithms have the same performance. When the zero hypothesis is rejected, we can perform the post-processing test of the Nemeny test [26].

Next, the average ranking and accuracy of the six algorithms on seven data sets are shown in Table 5.

Next, we take eight UCI datasets with 10% Gaussian noise as examples to compare the six algorithms. The formula for Friedman statistical variables is as follows:

$$
\chi^2_F = \frac{12N}{k(k+1)} \left( \sum_i R_i^2 - \frac{k(k+1)^2}{4} \right) = 30.18.
$$

(32)

where \( k \) is the number of algorithms and \( N \) is the number of UCI datasets. In our paper, \( k = 7, N = 8 \). \( R_i \) represents the average ranking of the \( i \) algorithm on the seven UCI datasets. In addition, according to the \( \chi^2 \) distribution with \( k - 1 \) degrees of freedom, we can obtain:

$$
F_F = \frac{(N - 1)\chi^2_F}{N(k - 1) - \chi^2_F} = 11.86.
$$

(33)

where \( F_F((k-1),(k-1)(N-1)) \) obeys the F-distribution, and its degree of freedom is \( (k - 1) \) and \( (k - 1)(N - 1) \). In this paper, we choose \( \alpha = 0.05 \) and we can get \( F_{a}(6,42) = 2.34 \). Obviously, \( F_F > F_a \), we reject the zero hypothesis.
Table 5. Average accuracy and ranks of six algorithms on UCI datasets with 0%, 10%, 25% Gaussian noise.

| Algorithm     | 0%  | 10% | 25% |
|---------------|-----|-----|-----|
| Avg.ACC       |     |     |     |
| SVM           | 74.31 | 73.08 | 71.64 |
| LSSVM         | 74.38 | 73.37 | 71.81 |
| C-SVM         | 76.14 | 74.54 | 72.97 |
| NPSVM         | 76.34 | 75.38 | 73.38 |
| Linex-SVM     | 76.76 | 75.48 | 74.44 |
| TBSVM         | 77.90 | 76.69 | 75.17 |
| Linex-TSVM    | 78.67 | 77.41 | 76.14 |
| Avg.rank      |     |     |     |
| SVM           | 6.63 | 6.38 | 6.63 |
| LSSVM         | 6.25 | 6.25 | 6.38 |
| C-SVM         | 4.38 | 4.13 | 4.63 |
| NPSVM         | 4.25 | 3.88 | 4.00 |
| Linex-SVM     | 3.38 | 3.50 | 2.88 |
| TBSVM         | 2.00 | 1.94 | 2.38 |
| Linex-TSVM    | 1.13 | 1.31 | 1.13 |

Intuitively, from the Table 5, we observe that Linex-TSVM has better classification performance, which means that our algorithm is more effective.

Next, through the Nemenyi post-hoc test, we can further compare the errors of the six algorithms in this paper. If the average rank difference between each other is greater than the critical value, the results demonstrate that the performance of the two algorithms is different. By dividing the Studentized range statistic by $\sqrt{2}$, we can obtain $q_{\alpha} = 2.95$. Therefore, we calculate the critical difference (CD) by the following formulation:

$$CD = q_{\alpha} \times \sqrt{\frac{k(k+1)}{6N}} = 3.18.$$  \hspace{1cm} (34)

Based on Figure 7, the performance of Linex-TSVM is significantly better than SVM, LSSVM, C-SVM, NPSVM, Linex-SVM and TBSVM, but the different between Linex-SVM and TBSVM is not obvious, because the different is smaller than the calculated CD value. Through the above analysis, the Linex-TSVM proposed in this paper has better performance.

Figure 7. Visualization of post hoc tests for UCI datasets.

5. Conclusions

The Twin Support vector machine classification has become a research hotspot. Twin support vector machine models based on different loss functions have been proposed, such as TPMSVM, TWSVM, SG-TSVM and so on. It is urgent to propose a loss function with better performance under the framework of support vector machine. The summary of this paper is as follows:

Firstly, this paper proposes capped linear loss function and applies it to twin support vector machine, and proposes a new robust classification model, which is called truncated linear loss function twin support vector machine. Compared with the linear loss support vector machine model proposed by Ma et al. [8], it has better classification performance.
Secondly, we give an efficient iterative algorithm to solve Linex-TSVM. Unlike SVM, which needs to solve a large QP problem, this algorithm needs to solve a pair of small QP problems. Finally, we strictly analyze the computational complexity of the algorithm; it is verified that Linex-SVM satisfies the Bayesian rule. Experimental results on multiple data sets demonstrate that our algorithm Linex-TSVM is more feasible and robust in dealing with large-scale datasets with outliers than other models, and intuitively show the convergence of the algorithm. In particular, compared with SVM, LSSVM, C-SVM, NPSVM, Linex-SVM and TBSVM, the average accuracy of Linex-TSVM is higher than that in the absence of noise. The average accuracy of the model in this paper is higher than that of 4.36%, 4.29%, 2.53%, 2.33%, 1.91% and 0.77%, respectively. Linex-TSVM is more robust and stable for outliers.

The focus of future work is that we should still focus on finding better models to improve different data classification results, shorten the computing time while ensuring accuracy, and extend the model of this paper to other work, such as multi-classification problems. In future work, we can further consider applying different models to practical hot issues, such as face recognition, fingerprint recognition, UAV scheduling and so on. Of course, how to develop a better new algorithm for our Linex-TSVM is also very important.

Author Contributions: Y.W., methodology, software, validation, formal analysis, investigation, data curation, writing—original draft. G.Y., conceptualization, methodology, validation, investigation, project administration, writing—original draft. J.M., project administration, writing—original draft. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (No. 11861002, 61907012) and Natural Science Foundation of Ningxia Provincial of China (No. 2022A0950). This research was also funded by the Young Talent Cultivation Project of North Minzu University (No. 2021KYQD23) and the Fundamental Research Funds for the Central Universities (No. 2022XYZSX03).

Informed Consent Statement: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References
1. Vapnik, V.N. Statistical Learning Theory; Wiley: New York, NY, USA, 1998.
2. Brown, M.P.; Grundy, W.N.; Lin, D.; Cristianini, N.; Sugnet, C.W.; Furey, T.S.; Haussler, D. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. USA* **2000**, *97*, 262–267. [CrossRef] [PubMed]
3. Ma, S.; Cheng, B.; Shang, Z.; Liu, G. Scattering transform and LSPTSVM based fault diagnosis of rotating machinery. *Mech. Syst. Signal Process.* **2018**, *104*, 55–170. [CrossRef]
4. Goh, K.S.; Chang, E.Y.; Li, B. Using one-class and two-class SVMs for multiclass image annotation. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 1333–1346. [CrossRef]
5. Bi, J.; Zhang, T. Support vector classification with input data uncertainty. In Proceedings of the Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 13–18 December 2004.
6. Saunders, C.; Gammerman, A.; Vovk, V. Ridge regression learning algorithm in dual variables. In Proceedings of the 15th International Conference on Machine Learning, ICML’98, Madison, WI, USA, 24–27 July 1998.
7. Suykens, J.A.K.; Vewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* **1999**, *9*, 293–300. [CrossRef]
8. Ma, Y.; Zhang, Q.; Li, D.; Tian, Y. Linex support vector machine for large-scale classification. *IEEE Access* **2019**, *7*, 70319–70331. [CrossRef]
9. Khemchani, R.; Chra, S. Twin support vector machines for pattern classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 905–910.
10. Gao, S.; Ye, Q.; Ye, N. 1-Norm least squares twin support vector machines. *Neurocomputing* **2011**, *74*, 3590–3597. [CrossRef]
11. Ye, Q.; Zhao, H.; Li, Z.; Yang, X.; Gao, S.; Yin, T.; Ye, N. L1-Norm distance minimization-based fast robust twin support vector k-plane clustering. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 4494–4503. [CrossRef] [PubMed]
12. Yan, H.; Ye, Q.; Zhang, T.A.; Yu, D.J.; Yuan, X.; Xu, Y.; Fu, L. Least squares twin bounded support vector machines based on L1-norm distance metric for classification. *Pattern Recognit.* **2018**, *74*, 434–447. [CrossRef]
13. Wu, M.J.; Liu, J.X.; Gao, Y.L.; Kong, X.Z.; Feng, C.M. Feature selection and clustering via robust graph-laplacian PCA based on capped L1-norm. In Proceedings of the 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Kansas City, MO, USA, 13–16 November 2017; pp. 1741–1745.
14. Zhang, C.; Wang, Z. Linex-RSVM: ramp linex support vector machine. Procedia Comput. Sci. 2022, 199, 524–531. [CrossRef]

15. Kinyanjui, J.K.; Korir, B.C. Bayesian Estimation of Parameters of Weibull Distribution Using Linex Error Loss Function. Int. J. Stat. Probab. 2020, 9, 1–38. [CrossRef]

16. Zou, G. Admissible estimation for finite population under the LINEX loss function. J. Stat. Plan. Inference 1997, 61, 373–384. [CrossRef]

17. Hwang, L.C. Second order optimal approximation in a particular exponential family under asymmetric linex loss. Stat. Probab. Lett. 2018, 137, 283–291. [CrossRef]

18. Wang, C.; Ye, Q.; Luo, P.; Ye, N.; Fu, L. Robust capped L_1-norm twin support vector machine. Neural Netw. 2019, 114, 47–59. [CrossRef] [PubMed]

19. Zhang, L.; Luo, M.; Li, Z.; Nie, F.; Zhang, H.; Liu, J.; Zheng, Q. Large-scale robust semisupervised classification. IEEE Trans. Cybern. 2018, 49, 907–917. [CrossRef] [PubMed]

20. Nie, F.; Huang, Y.; Wang, X.; Huang, H. New primal SVM solver with linear computational cost for big data classifications. In Proceedings of the 31st International Conference on International Conference on Machine Learning, Beijing, China, 22–24 June 2014; Volume 32, pp. 505–513.

21. Nie, F.; Huo, Z.; Huang, H. Joint capped norms minimization for robust matrix recovery. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017.

22. Zhao, M.; Chow, T.W.; Zhang, H.; Li, Y. Rolling fault diagnosis via robust semi-supervised model with capped L_{2,1}-norm regularization. In Proceedings of the IEEE International Conference on Industrial Technology, Toronto, ON, Canada, 22–25 March 2017; pp. 1064–1069.

23. Nie, F.; Wang, X.; Huang, H. Multiclass capped L_p-norm SVM for robust classifications. In Proceedings of the 32th AAAI Conference on Artificial Intelligence, Palo Alto, CA, USA, 4–9 February 2017.

24. Ahmadi, J.; Doostparast, M.; Parsian, A. Estimation and prediction in a two-parameter exponential distribution based on k-record values under LINEX loss function. Commun.-Stat. Theory Methods 2005, 34, 795–805. [CrossRef]

25. Pandey, B.N.; Dwividi, N.; Pulastya, B. Comparison between Bayesian and maximum likelihood estimation of the scale parameter in Weibull distribution with known shape under linex loss function. J. Sci. Res. 2011, 55, 163–172.

26. Demar, J. Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 2006, 7, 1–3.